Imaging Studies of Peripheral Nerve Regeneration Induced by Porous Collagen Biomaterials

by

Dimitrios S. Tzeranis

S.M., Massachusetts Institute of Technology (2005)
Diploma, National Technical University of Athens (2003)

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

September 2013

Signature of Author .....
Department of Mechanical Engineering
July 12, 2013

Certified by .....
Ioannis V. Yannas
Professor of Mechanical Engineering and Biological Engineering
Thesis Supervisor

Accepted by .....
David E. Hardt
Ralph E. and Eloise F. Cross Professor of Mechanical Engineering
Chairman, Department Committee on Graduate Students
Department of Mechanical Engineering

Imaging Studies of Peripheral Nerve Regeneration Induced by Porous Collagen Biomaterials

by

Dimitrios S. Tzeranis

Submitted to the Department of Mechanical Engineering
on July 12, 2013 in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Mechanical Engineering

Abstract

There is urgent need to develop treatments for inducing regeneration in injured organs. Porous collagen-based scaffolds have been utilized clinically to induce regeneration in skin and peripheral nerves, however still there is no complete explanation about the underlying mechanism. This thesis utilizes advanced microscopy to study the expression of contractile cell phenotypes during wound healing, a phenotype believed to affect significantly the final outcome.

The first part develops an efficient pipeline for processing challenging spectral fluorescence microscopy images. Images are segmented into regions of objects by refining the outcome of a pixel-wide model selection classifier by an efficient Markov Random Field model. The methods of this part are utilized by the following parts.

The second part extends the image informatics methodology in studying signal transduction networks in cells interacting with 3D matrices. The methodology is applied in a pilot study of TGFβ signal transduction by the SMAD pathway in fibroblasts seeded in porous collagen scaffolds. Preliminary analysis suggests that the differential effect of TGFβ1 and TGFβ3 to cells could be attributed to the "non-canonical" SMAD1 and SMAD5.

The third part is an *ex vivo* imaging study of peripheral nerve regeneration, which focuses on the formation of a capsule of contractile cells around transected rat sciatic nerves grafted with collagen scaffolds, 1 or 2 weeks post-injury. It follows a recent study that highlights an inverse relationship between the quality of the newly formed nerve tissue and the size of the contractile cell capsule 9 weeks post-injury. Results suggest that "active" biomaterials result in significantly thinner capsule already 1 week post-injury.

The fourth part describes a novel method for quantifying the surface chemistry of 3D matrices. The method is an *in situ* binding assay that utilizes fluorescently labeled recombinant proteins that emulate the receptor of interst, and is applied to quantify the density of ligands for integrins α1β1, α2β1 on the surface of porous collagen scaffolds. Results provide estimates for the density of ligands on "active" and "inactive" scaffolds and demonstrate that chemical cross-linking can affect the surface chemistry of biomaterials, therefore can affect the way cells sense and respond to the material.

Thesis Co-Supervisor: Ioannis V. Yannas
Title: Professor of Polymer Science and Engineering
Thesis Co-Supervisor: Peter TC So
Title: Professor of Mechanical and Biological Engineering

Ἀγαπήσω σε Κύριε ἡ ἰσχύς μου, Κύριος στερέωμά μου, καὶ καταφυγή μου, καὶ ῥύστης μου.

Acknowledgements

This thesis would not exist without the endless support of my two academic advisors, Prof. Ioannis V. Yannas and Prof. Peter TC So. I am grateful to both of you for being a constant source of inspiration throughout the difficult and demanding years of my PhD studies. I learned so much at MIT because you gave me the chance to work in your labs. I appreciate the fact that you never prevented me from following my curiosity to explore new scientific pathways. You taught me to be a better person and you have been valuable advisors, mentors, and friends.

I am thankful to Prof. Richard Hynes and Prof. Mark Bathe for accepting my invitation to become member of my PhD committee. You provided me valuable guidance and support, and have always been available for discussions.

I would also like to thank other faculty that provided me valuable input and advice. Prof. Barry Masters for his critical feedback and commitment on the ethical conduct of research. Prof. Myron Spector for his kind support, and professional advice.

I would like to thank the members of the Yannas Lab, the So Lab, and my collaborators for helping me conduct my experiments, troubleshoot my instruments, and for making my life in the lab enjoyable. Particularly, I would like to thank Dr. Eric Soller and Melissa Buydash for their extremely valuable help in conducting many kinds of experiments, and the many fun coffee breaks in lobby 7. Chengpin Shen and Jin Guo for setting up the image informatics methodology. Dr. Daekeun Kim, Jaewon Cha, Dr. Chris Rowlands, and Dr. Karsten Bahlman for troubleshooting my microscope. Kathy, Hyuk Sang, Poorya, Yang-Hyo, Sebastian, Yun-Ho, and Heejin for their collaboration and friendship. Dr Nora Sullivan for the fun time during working with the chaotic phenazine-producing bacteria. Giorgios Papachristoudis for providing me an implementation of the belief propagation algorithm. Several undergraduate students (Amanda, Lily, Rebecca, Viveka, Sean, David, Cindy) that I supervised through the UROP program for providing assistance to my work and making the lab a fun place to be.

This work could not be possible without several PIs letting me use the facilities of their labs, and introducing me to their students and post-doctoral fellows. My good friends Dr. Vassilios Dendroulakis, Dr. Bahar Edrissi, Dr. Mike DeMott from the Lab of Prof. Peter C. Dedon (dept. of biological engineering, MIT). Dr. Itzel Ramos-Solis, Prof. Yun Wang, Dr. Nora Sullivan, Prof. Lars Dietrich from the lab of Prof. Dianne K. Newman (dept. of Biology, MIT). My friends Dr. Kaan Certel, Jessica Louie, Dr. Tom Seegar from the lab of Prof. Richard O. Hynes (dept of Biology, MIT). Dr. John Maloney from the Van Vliet Lab (dept. of materials science and engineering, MIT). The one and only Xavier Duportet from the Weiss Lab (dept. of biological engineering, MIT). Dr Albert Tai (Tufts University core facility genomics core) for his valuable help and advice in conducting the BIACORE experiments. Mrs Deborah Pheasant (Biophysical Instrumentation Facility, MIT) for her valuable help in conducting the circular dichroism experiments.

Apart from all the people mentioned above, I would like to thank all the wonderful people that I met during my presence in Boston, who made my life fun and interesting, and provided valuable work-life balance: Ioannis, Christoforos, Argyris, Apostolos, Andreas, Antonis, Tasos, Paul, Paul, Peggy, Pepe, Rogelio, David, Manolis, Anna, Kostas, Thodoris, Spiros, Antonis, Vassilis, Josh, Jonathan, Mehmet, Leonidas, Magda, Stella, Ismini, Sophia, Olga, Magdalena, Yola, Ioannis, Giorgos, Mixalis, Ioannis, Nikos, Giorgos, Maria, Lila, Evangelos, Dimitris^n, Sungmin, Eleni, Youngjoo, Stefania, Giorgos^n, Kimonas, Nina, Konstantinos, Panagiotis, Sheila, Charalabos, Yue, Dimitra, Angelos, and Viktor.

My MIT experience would not be possible without the inspiration and the support of several teachers during my high-school years (particularly Mr Pristouris), and several Professors during my undergraduate years at the National Technical University of Athens: Prof. E. G. Papadopoulos (my advisor during my undergraduate thesis), Prof. K. J. Kyriakopoulos, Prof. G-C. Vosniakos, Prof. S. E. Simopoulos, and Prof. D. I. Pantelis.

I am grateful to my parents and my brother for their endless love, patience and support all these years, and for letting me free to decide my own future. I am thankful to my extended family in New Jersey for never letting me feel alone during my stay in the U.S.A., and for their love and support.

Finally, I am grateful to Maria for her love, for reminding me the important things in life, and for succeeding in withstanding the sacrifices that the completion of this thesis demanded from her.

Table of Contents

Abbreviation List

3D	three-dimensional
5-CTMR	5-Carboxytetramethylrhodamine
αSMA	alpha smooth muscle actin
βme	2-mercaptoethanol (CAS: 60-24-2)
ALS	alternative least squares
Amp	ampicillin
ATP	adenosine triphosphate (CAS: 56-65-5)
AU	arbitrary units
BAL	2,3-dimercaptopropanol
BD	biarsenical dyes
BFP	blue fluorescent protein
BP	Belief propagation
BSS	blind signal separation
CAF	cancer-associated fibroblasts
CAS	chemical abstracts service number
CBI	collagen-binding integrin
CCD	charge coupled device
CD	circular dichroism
CDF	cumulative distribution function
CFP	cyan fluorescent protein
CG	collagen-GAG
CMTMR	(5-(and-6)-(((4-Chloromethyl)Benzoyl)Amino)Tetramethylrhodamine)
CNS	central nervous system
CPU	central processing unit
CRB	Cramer-Rao bound
DAPI	4',6-diamidino-2-phenylindole
DHT	dehydro-thermal treatment
DMEM	Dubelcco's Modified Eagle Medium
DNA	deoxyribonucleic acid
DTT	dithiothreitol (CAS: 3483-12-3)
ECM	extracellular matrix
EDC	1-Ethyl-3-(3-dimethylaminopropyl)carbodiimide (CAS: 1892-57-5)
EDT	1,2-ethanedithiol
EDTA	ethylenediaminetetraacetic acid (CAS: 60-00-4)
EM	expectation maximization
EM	electromagnetic
EMCCD	electron multiplying charged coupled device
EtBr	ethidium bromide
FAK	focal adhesion kinase
FB	fibroblast
FCS	fluorescence correlation spectroscopy
FITC	fluorescein isothiocyanate

FlAsH	4,5-bis(1,3,2-dithiarsolan-2-yl) fluorescein
FLIM	fluorescence lifetime imaging
FoV	field of view
FP	fluorescent protein
FRET	Förster (Fluorescence) resonance energy transfer
GAG	glycosaminoglycans
GC	Graph cuts
GCN	Gaussian – common noise
GFP	green fluorescent protein
GPCR	G-protein coupled receptors
GPU	graphics processing unit
GST	glutathione S-transferase (CAS: 50812-37-8)
GUI	graphical user interface
GVN	Gaussian – variable noise
HDF	human dermal fibroblasts
HBS-P	hepes-buffered saline plus surfactant P20
HSV	hue-saturation-value
ICA	independent component analysis
ICM	Iterated conditional modes
ID	inner diameter
IF	immunofluorescence
IPTG	isopropyl β-D-1-thiogalactopyranoside (CAS: 367-93-1)
IR	infrared
KKT	Karush–Kuhn–Tucker
LB	luria borth
LED	light emitting diode
LS	least-squares
MA-PMT	multi anode photomultiplier tube
MAP	maximum a posteriori
MAPK	mitogen-activated protein kinase
MES	2-mercaptoethanesulfonate
MFB	myofibroblast
MIT	Massachusetts Institute of Technology
ML	maximum likelihood
MLL	multi-level logistic
MMM	multi-focal multi-photon microscopy
MMP	matrix metalloproteinase
MOI	multiplicity of infection
MPM	multiphoton microscopy
MRF	markov random field
MW	molecular weight
MWCO	molecular weight cut-off
NA	numerical aperture
NADH	nicotinamide adenine dinucleotide (CAS: 58-68-4)

NF200	neurofilament heavy chain (200kDa)
NHS	N-Hydroxysuccinimide (CAS: 6066-82-6)
NIH	national institute of health
NIR	near infrared light
NNLS	non-negative least squares
NRMSD	normalized root-mean-square deviation
OCT	optimal cutting temperature
PAGE	polyacrylamide gel electrophoresis
PBS	phosphate buffered saline
PBS^-	phosphate buffered saline, no $MgCl_2$ and $CaCl_2$
PCA	principal component analysis
PCA	phenazine-1-carboxylic acid
PCG	preconditioned conjugate gradient
PCN	phenazine-1-carboxamide
PDB	protein data bank
PDF	probability density function
PI-3	phosphatidylinositol 3
PKB	protein kinase B
PKC	protein kinase C
PMF	probability mass function
PMT	photomultiplier tube
PN	peripheral nerve
PNI	peripheral nerve injury
PNR	peripheral nerve regeneration
PNS	peripheral nerve system
POPOP	(1,4-Bis(5-phenyloxazol-2-yl)benzene
PSF	point spread function
PVD	pyoverdin
PYO	pyocyanine
RBS	ribosome binding site
ReAsH	4,5-bis(1,3,2-dithiarsolan-2-yl) resorufin
RGB	red-green-blue
RI	refractive index
RMSD	root-mean-square deviation
RNA	ribonucleic acid
RNAi	RNA interference
ROC	receiver operating characteristic
RS	remote sensing
RTK	receptor tyrosine kinase
RU	resonance units
SA	Simulated annealing
SC	Schwann cells
SDS	sodium dodecyl sulfate (CAS: 151-21-3)
SEM	scanning electron microscopy

SH	second harmonic	
shRNA	small hairpin RNA	
SNR	signal-to-noise ratio	
SPBA	solid phase binding assay	
SPR	surface plasmon resonance	
SS	steady state	
SVD	singular value decomposition	
TAE	tris acetate edta	
TC	tetracysteine	
TC	treatment condition	2
TCEP	tris(2-carboxyethyl)phosphine (CAS: 5961-85-3)	
TCSPC	time-correlated single-photon counting	
TE	tris-edta	
TEM	transmission electron microscopy	
TGFβ	transforming growth factor beta	
TIR	total internal reflection	
TPE	two-photon excitation	
TPEF	two photon-excitated fluorescence	
Tris	Tris(hydroxymethyl)aminomethane (CAS: 77-86-1)	
TRITC	rhodamine isothiocyanate	
TRS	thrombin recognition site	
TRWP	Tree-Reweighted Message passing	
TTL	transistor–transistor logic	
VIS	visible	
UV	ultraviolet	
WB	western blot	
WD	working distance	
WLS	weighted least squares	
YFP	yellow fluorescent protein	

Chapter 1: Introduction

This Chapter provides a brief introduction to the motivation, challenges, and key contributions of this thesis. Each Chapter is organized so that it can be read independently of the rest.

1.1 The Clinical Need for Induced Regeneration

Regenerative medicine is a branch of science and engineering that leads to development of treatments to restore the structure and function in damaged organs or tissues. Organ damage, due to either acute (e.g. accidents, blasts) or chronic injuries due to disease, is a common problem faced by millions of humans. In mammals (including humans) severe organ injury initiates a complex cascade of biological events (wound healing) that does not result in the restoration of the initial structure and function of the organ. Instead, wound healing in adult mammals leads to an organ that is usually severely compromised in terms of function and structure. This can result in significant human morbidity and can compromise quality of life.

Several animals are known to be able to regenerate spontaneously severely injured organs. For example, certain amphibians can regenerate large parts of their limbs [Brockes and Kumar 2008; Kragl et al. 2009] through their ability to de-differentiate cells in the proximity of the injury, eventually leading to re-differentiation to the appropriate cell types required for forming a new limb [Gurtner et al. 2008; Poss 2010]. Recently a strain of mouse was reported able to regenerate spontaneously large skin wounds, possibly the result of an evolutionary modification for escaping predators [Seifert et al. 2012]. It is known that mammals can regenerate injured organs during the early phase of gestation [Gurtner et al. 2008]. These observations suggest that adult mammals may conceivably possess the appropriate regenerative capacity in latent form, which under suitable conditions could lead to organ regeneration.

Organ regeneration has been a dream of humans since ancient times (Prometheus has claimed the very first patent in liver regeneration). However, treatments that provide clinically significant regeneration in adults have been developed only over the past thirty years [Yannas et al. 1982]. Such treatments include the use of porous collagen-based biomaterial grafts to induce regeneration in injured skin [Yannas et al. 1989] and peripheral nerves [Chamberlain et al.1998b]. Despite the success of these materials, there is a pressing need to develop new treatments that could induce regeneration in other organs. To address this need, a great deal of research effort is underway towards various directions: biomaterial development, decellularized xenografts, stem cells, and chemical strategies. Nevertheless, the field of regenerative medicine can still benefit very much by improvement in understanding of just how existing successful treatments manage to change the wound healing outcome in ways that favor regeneration.

1.2 Challenges in Biomaterial Design

Over the past twenty years a lot of research has been devoted to the development of new materials that could induce regeneration in a wide range of injured organs [Ratner and Bryant 2004]. Several companies have developed collagen-based materials similar to the ones described in [Yannas et al. 1982]. A large number of research efforts have led to development

of new bio-mimetic materials, which have similar structure (at the micron level) and/or chemical composition as the extracellular matrix (ECM) of the organ of interest [Ratner and Bryant 2004]. Such approaches include the application of novel 3D printing methods [Derby 2012]. Other research efforts focus on designing artificial materials that can guide the differentiation of stem cells into organ-specific cell types [Lutolf and Hubbell 2005; Sands and Mooney 2007]. Such approaches consider regeneration as a synthetic event that emulates the formation of the organ during development. However, the process of regeneration is probably more complicated. The wound healing process is a complex cascade of many processes that affect the synthesis of new organ tissue. Therefore, the rational design of biomaterials would benefit from consideration of all aspects of wound healing, and not just the synthesis of new organ tissue. One example of a wound healing process that has been largely neglected in the literature is closure of a wound by contraction [Yannas 2001]. Wound contraction refers to the organized expression of contractile phenotypes in the wound leading to differentiated cells that induce wound closure primarily by pulling together the wound edges toward the center of the wound.

Another factor that prevents principle-based design of biomaterials, is the lack of knowledge of the effects induced in cells by their interaction with insoluble 3D microenvironments, such as those presented by biomaterials. Even though the elementary molecular components that mediate cell-matrix adhesion are known, there is still a lot to learn about how adhesion of different cell types to a variety of 3D matrices by specific receptors regulates cell phenotypes related to wound healing. Investigators of the required cell-matrix interaction studies that could provide this information face several challenges. One of them is the difficulty in quantifying the microenvironment that cells sense inside the matrix. At the moment there are very few methods for quantifying *in situ* several key properties of a 3D matrix that are detected by cells, including its surface chemistry and its strain state. Another factor that makes cell-matrix studies hard is the difficulty in quantifying the cell response inside highly porous 3D matrices used as biomaterials (scaffolds) *in situ*. Finally, due to the complex nature of the wound healing process, these studies need to be done *in vivo* using animal models. Quantifying cell-biomaterial interactions and their resulting effects on the synthesis of new organ tissue *in vivo* is labor intensive, and expensive.

1.3 Using 3D Optics to Study Cell-Matrix Interactions

3D imaging methods are imaging modalities that have optical sectioning capability and therefore can provide sharp images of the focal plane located inside a 3D sample with minimal interference from out-of focus signals. 3D imaging methods, such as confocal microscopy [Pawley 2006] and multi-photon microscopy [Denk et al. 1990], can provide a 3D image of a thick specimen without the need to cut the sample into thin sections.

3D optics and in particular multi-photon microscopy is a very promising methodology for studying 3D matrices, for several reasons. They enable high-resolution imaging of cells and biomaterials *in situ* without the need to stain or cut the cell-matrix system. Combined with the toolbox of fluorescent markers and modern gene manipulation methods, they can provide molecular level information for a large variety of processes. Multi-photon microscopy can be implemented either with *in vitro* experimental systems, or with *in vivo* animal models. Several components of the ECM (frequently used in biomaterial fabrication) have intrinsic optical

spectroscopic signatures. Finally, advanced imaging methods enable multiplexing several optical signals in order to acquire information-rich data with small effort.

1.4 Organization of this Thesis

In this thesis we have exploited the capabilities of multi-photon imaging in order to conduct three studies related to peripheral nerve regeneration. The three studies are designed to acquire information that describes quantitatively cell-matrix interactions and can be put to use in principle-based biomaterial design. Of particular interest is the study of phenomena related to wound contraction.

This thesis is organized in the following manner:

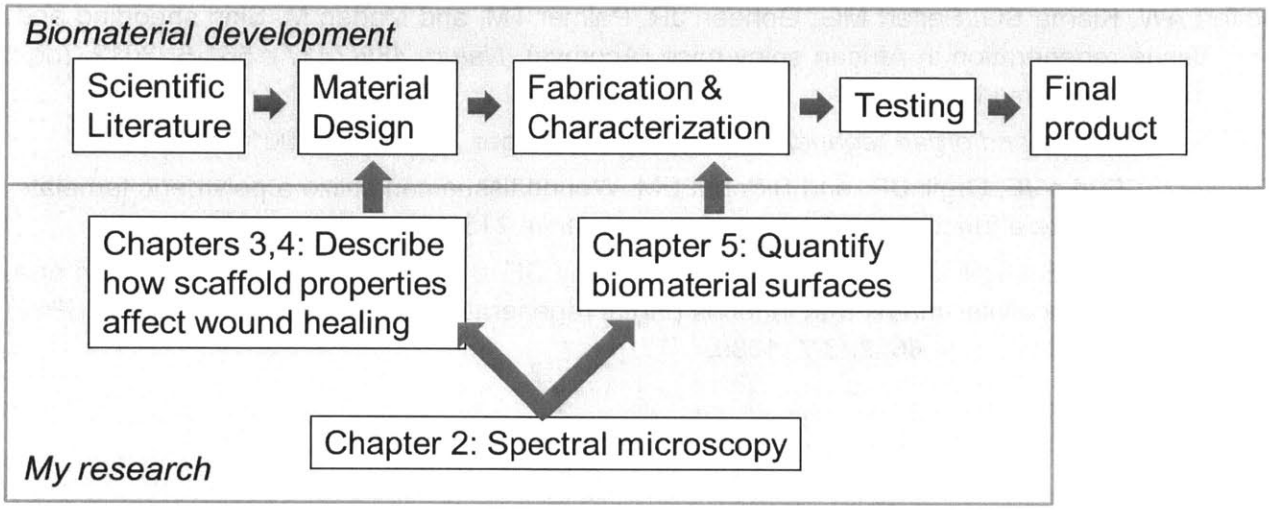
- Chapter 2 addresses the application of spectral multi-photon microscopy in the acquisition of large information-rich data sets. It describes an image processing pipeline that processes raw spectral microscopy data and outlines physical objects that can be further described quantitatively in an application-dependent way. It provides analytical information that can be used to determine the accuracy and the precision of the calculations as well as guidelines for the design of accurate high-content imaging. This chapter provides a toolbox that is used by the following three chapters, all of which utilize spectral multi-photon microscopy.

- Chapter 3 describes an imaging-based approach for studying signal transduction pathways in cells that interact with 3D matrices. The methodology developed is applied to a pilot study of TGFβ signaling via the SMAD pathway in primary cells seeded inside collagen scaffolds, an environment similar to that applied clinically for skin regeneration. The TGFβ pathway was chosen because it is known to regulate wound contraction, a critical cell phenotype for wound healing. The method can be utilized to study how cell-matrix adhesion controls cell phenotypes by modulating particular signal transduction pathways.

- Chapter 4 describes an *ex vivo* imaging study of induced regeneration in peripheral nerves. This study follows recent findings of a strong inverse relationship between the quality of nerve regeneration and wound contraction 9 weeks post-injury in experimental animals. It utilizes the imaging tools described in Chapter 2 to provide high-resolution high-content images of the contractile cell capsule that forms around the newly-formed nerve tissue during the early phases of wound healing. The study provides results for wounds grafted with two collagen conduits known to possess significantly different ability to induce peripheral nerve regeneration (one is "active", the other is "inactive"), in order to collect additional evidence supporting the hypothesis that nerve regeneration conduits induce regeneration by blocking wound contraction.

- Chapter 5 describes a novel methodology for quantifying the surface chemistry of 3D matrices *in situ.* The method utilizes soluble fluorescent markers that emulate the binding behavior of the adhesion receptor of interest. This Chapter provides extensive biochemical characterization of the markers. The density of adhesion ligands is

quantified via an *in situ* binding assay based on the multi-photon imaging tools described in Chapter 2 and appropriate binding analysis. The methodology developed is applied to the quantification of ligand density for integrins α1β1 and α2β1 for the two collagen scaffolds used in the animal study described in Chapter 4. This methodology can be used to characterize different kinds of biomaterials by quantifying the insoluble microenvironment felt by cells inside them.



1.5 Literature Cited

Abou Neel EA, Bozec L, Knowles JC, Syed O, Mudera V, Day R, and Hyun JK. Collagen--emerging collagen based therapies hit the patient. *Adv Drug Deliv Rev.* **65**(4): 429-56, 2013. (doi: 10.1016/j.addr.2012.08.010).

Brockes JP, and Kumar A. Comparative aspects of animal regeneration. *Annu Rev Cell Dev Biol.* **24**: 525-49, 2008. (doi: 10.1146/annurev.cellbio.24.110707.175336).

Chamberlain LJ, Yannas IV, Hsu HP, Strichartz G, and Spector M. Collagen-GAG substrate enhances the quality of nerve regeneration through collagen tubes up to level of autograft. *Exp Neurol.* **154**(2): 315-29, 1998b.

Denk W, Strickler JH, and Webb WW. Two-photon laser scanning fluorescence microscopy. *Science* **248**(4951): 73-6, 1990.

Derby B. Printing and prototyping of tissues and scaffolds. *Science* **338**(6109): 921-6, 2012. (doi: 10.1126/science.1226340).

Gurtner GC, Werner S, Barrandon Y, Longaker MT. Wound repair and regeneration. *Nature* **453**(7193): 314-21, 2008. (doi: 10.1038/nature07039).

Kragl M, Knapp D, Nacu E, Khattak S, Maden M, Epperlein HH, and Tanaka EM. Cells keep a memory of their tissue origin during axolotl limb regeneration. *Nature* **460**(7251): 60-5, 2009. (doi: 10.1038/nature08152).

Lutolf MP, and Hubbell JA. Synthetic biomaterials as instructive extracellular microenvironments for morphogenesis in tissue engineering. *Nat Biotechnol.* **23**(1):47-55, 2005.

Pawley J. *Handbook of Biological Confocal Microscopy*, Third Edition, Springer, 2006.

Poss KD. Advances in understanding tissue regenerative capacity and mechanisms in animals. *Nat Rev Genet.* **11**(10): 710-22, 2010. (doi: 10.1038/nrg2879).

Ratner BD, and Bryant SJ. Biomaterials: where we have been and where we are going. *Annu Rev Biomed Eng.* **6**: 41-75, 2004.

Sands RW, and Mooney DJ. Polymers to direct cell fate by controlling the microenvironment. *Curr Opin Biotechnol.* **18**(5): 448-53, 2007.

Seifert AW, Kiama SG, Seifert MG, Goheen JR, Palmer TM, and Maden M. Skin shedding and tissue regeneration in African spiny mice (Acomys). *Nature* **489**(7417): 561-5, 2012. (doi: 10.1038/nature11499).

Yannas IV. *Tissue and organ regeneration in adults*, Springer, New York, 2001.

Yannas IV, Burke JF, Orgill DP, and Skrabut EM. Wound tissue can utilize a polymeric template to synthesize a functional extension of skin. *Science* **215**(4529): 174-6, 1982.

Yannas IV, Lee E, Orgill DP, Skrabut EM, and Murphy GF. Synthesis and Characterization of a model extracellular-matrix that induces partial regeneration of adult mammalian skin. *Proc Natl Acad Sci U.S.A.* **86**: 933-7, 1989.

Chapter 2: Probabilistic Data Processing for Spectral Fluorescence Microscopy

2.1 Chapter Overview

This chapter describes a methodology for processing data acquired by single photon-counting spectral multi-photon microscopy. The methodology consists of a general purpose image processing pipeline (a series of image processing tasks) that converts raw spectral imaging data into information that can be further utilized for application-dependent quantitative analysis. Specifically, the pipeline provides two outputs: i) a segmentation of the 3D image into regions of particular physical meaning ("classes") described by the presence of particular emission sources, ii) an estimation of the emission rate of the emission sources present in each pixel. The methodology described in this chapter has been developed for processing low-signal images of complex samples (samples that contain multiple kinds of "classes" and multiple emission sources), and can be applied to process images of a large variety of biological samples. The remaining chapters of this thesis describe imaging-based studies related to cell-matrix interactions, all of which rely in the methodology described in this chapter.

2.1.1 Motivation

Fluorescence microscopy is one of the most widely applied tools in biological research and medicine, and is applied in increasingly more complex biological systems. This trend has been triggered by the development of novel imaging modalities (such as multi-photon microscopy [Denk et al. 1990] and super-resolution imaging [Hell and Wichman 1994; Betzig et al. 2006; Rust et al. 2006; Bates et al. 2007; Hell 2007]), by the development of fluorescent protein technology [Zhang et al. 2002; Giepmans et al. 2006; Lippincott-Schwartz and Patterson 2009; Lippincott-Schwartz 2011; Crivat and Taraska 2012], and by the development of molecular biology methods that enable fluorescent tagging in cells or organ-specific expression of fluorescent markers animals (for example the "rainbow moue" [Livet et al. 2007]). These technological advances led to novel imaging-based methodologies such as large-scale *in vitro* imaging-based screening [Carpenter and Sabatini 2004; Pepperkok and Ellenberg 2006; Carpenter 2007; Kitami et al 2012], image informatics [Orvedahl et al. 2011], and *in vivo* imaging of disease models in small animals [Kerschensteiner et al. 2005]. Fluorescence imaging based methods can provide unique information of high temporal and spatial resolution that would be otherwise hard or impossible to acquire using the traditional tools of biological sciences (genetics, biochemistry, and molecular biology).

The most common fluorescence-based imaging modality is wide-field microscopy, a straightforward technique that acquires 2D images and requires cheap instruments found in almost all biological research labs. Usually samples contain more than one fluorescence emission sources, and each fluorophore is imaged sequentially using appropriate filter sets (emission/excitation filters and dichroic filter). A similar sequential approach for imaging multiple sources is utilized in confocal fluorescence microscopy, the most widely-used 3D imaging modality [Pawley 2006]. However the finite size of fluorophore emission spectra limits the

number of fluorophores that can be imaged using this standard approach to 3 or 4, and the sequential acquisition of an image for each source reduces throughput.

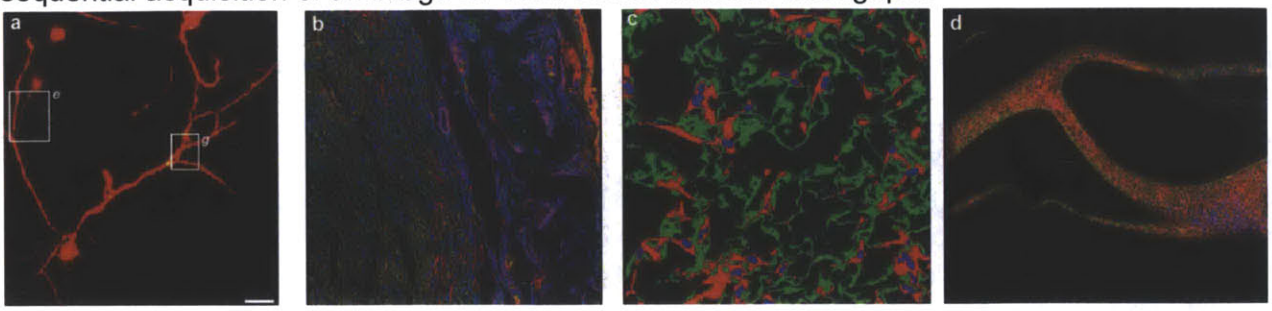


Figure 2.1.1: Applications of multi-photon imaging. **a**: imaging inhibitory synapses and dendritic spines in L2/3 pyramidal neurons of mice brain *in vivo* [Chen et al. 2012]. **b:** spectral multi-photon imaging of transected rat sciatic nerves treated with collagen biomaterials *ex vivo* (Chapter 4). **c:** image informatics study of TGFβ signaling in fibroblast interacting with porous collagen scaffolds *in vitro* (Chapter 3). **d:** *in situ* quantification of surface chemistry on porous collagen biomaterials by multi-photon microscopy (Chapter 5).

The application of imaging-based methods in increasingly complex problems leads to the need for increasingly larger throughput (information acquired per time). Currently there are two approaches to increase throughput in imaging experiments:

- Increase the imaging acquisition speed. Examples include the development of multi-focal multi-photon microscopy (MMM) [Kim et al. 2007] and wide-field multi-photon microscopy [So and Kim 2009; Kim and So 2010].
- Increase the information content of the acquired images by acquiring data that can provide more information per experimental time. Examples include spectral microscopy, fluorescence lifetime imaging (FLIM), and spectral-lifetime imaging [Bird et al. 2004].

This thesis focuses on the second approach for increasing the throughput of biological fluorescence microscopy. Specifically, this chapter describes a novel image processing pipeline for extracting high-content information from low-signal spectral multi-photon microscopy data. The remaining chapters utilize this methodology to study several phenomena related to peripheral nerve regeneration. The developed methodology is not limited to multi-photon microscopy, and can be applied also in spectral confocal fluorescence microscopy.

2.1.2 Objectives

The objective is to develop an image processing pipeline that converts raw spectral microscopy data into reliable information that can be used to study biological phenomena quantitatively, such as the ones described in Chapters 3-5. The pipeline is developed with the following key specifications in mind:

- Process large imaging datasets (10^6-10^7 pixels) with feasible computational cost.
- Process complex images that contain multiple kinds of objects, each containing multiple distinct emission sources. Resolve both the spatial extent of the objects and their composition (contribution of emission sources).
- Capable to process low signal images (less than 50 photon counts per pixel) to enable studying difficult samples that contain multiple weak endogenous fluorophores.
- Applicable to various kinds of spectral microscopes, and not just to the particular spectral multi-photon instrument (Section 2.3) used in this thesis.

2.1.3 Image Processing Pipeline Overview

Overview of Image Processing Tasks

Chapter 2 describes a generic (application independent) image processing pipeline that can be utilized as the first computation steps in a wide variety of imaging-based studies (for example the three imaging applications described in Chapters 3-5) based on spectral 3D microscopy. The pipeline is implemented in MATLAB (Appendix M) and consists of five major tasks:

- Image registration. Due to optical reasons the field of view (FoV) of microscopy images (called frames) is limited. Imaging samples whose dimensions are larger than reasonable FoV is implemented by sequentially first translating the sample in the X-Y plane (image plane) using a stage and then acquiring frames. Each frame of a 3D microscope (e.g. confocal, multi-photon) images a thin slice of the sample. 3D imaging is implemented by translating the objective using a piezoelectric actuator and acquiring frames at different XY planes inside the sample. Image registration is the task of calculating the optimal "stitching" of the frames I at the same plane. The outcome of image registration is then utilized to assemble the frames. Since the spectral signatures $\boldsymbol{s}_i$ may be position dependent (row-column), it is advised to stitch the frames after local segmentation.
- Spectral unmixing: This is the key computation in processing spectral microscopy data, For each pixel of an image, spectral unmixing processes the M-channel signal $\boldsymbol{y}$ in order to estimate the emission rates of the N sources assumed to be present in the sample. The accuracy and precision of spectral unmixing depends on several factors, including the number, identity, and emission rates of the sources present in the sample modeling errors, and the degree of accuracy in knowledge of source spectral signatures $\boldsymbol{s}_i$ (Section 2.4).
- Spectral calibration. Sometimes spectral signatures $\boldsymbol{s}_i$ can be measured experimentally. Otherwise, spectral signatures need to be estimated based on the spectroscopic properties of the source and the optical configuration of the instrument. The purpose of the spectral calibration step is to estimate parameters of the image detection system that affect estimation accuracy of the source spectral signatures $\boldsymbol{s}_i$ (Section 2.3.7).
- Local image segmentation: This step classifies each pixel into one of possible "classes". A class is a region that corresponds to a specific physic entity. Each class contains a distinct set of emission sources. For each pixel, pixel-wide segmentation consists of three steps: spectral unmixing, class evidence calculation, and Bayesian classification (Section 2.5.2).
- Global image segmentation: This step refines the outcome of the pixel-wide segmentation in order to improve segmentation accuracy. The need for global segmentation is due to the fact that local segmentation is not accurate at low photon counts, does not incorporate information about the optical physics of the system, and does not incorporate a-priori knowledge on the "meaningful" spatial context in the image. Global segmentation partitions each image into regions, defined as continuous clusters of pixels that belong to a class (Section 2.5.3).

This generic image processing pipeline provides a segmentation of the assembled frames into regions of particular classes, and estimates of the sources present at each pixel based on the assigned class of this pixel. Imaging-based studies may require subsequent application-specific image processing tasks that convert regions into objects (e.g. cells, fibers), quantify objects

using a series of features, and finally use statistical analysis to process the features and provide answers to biological questions:

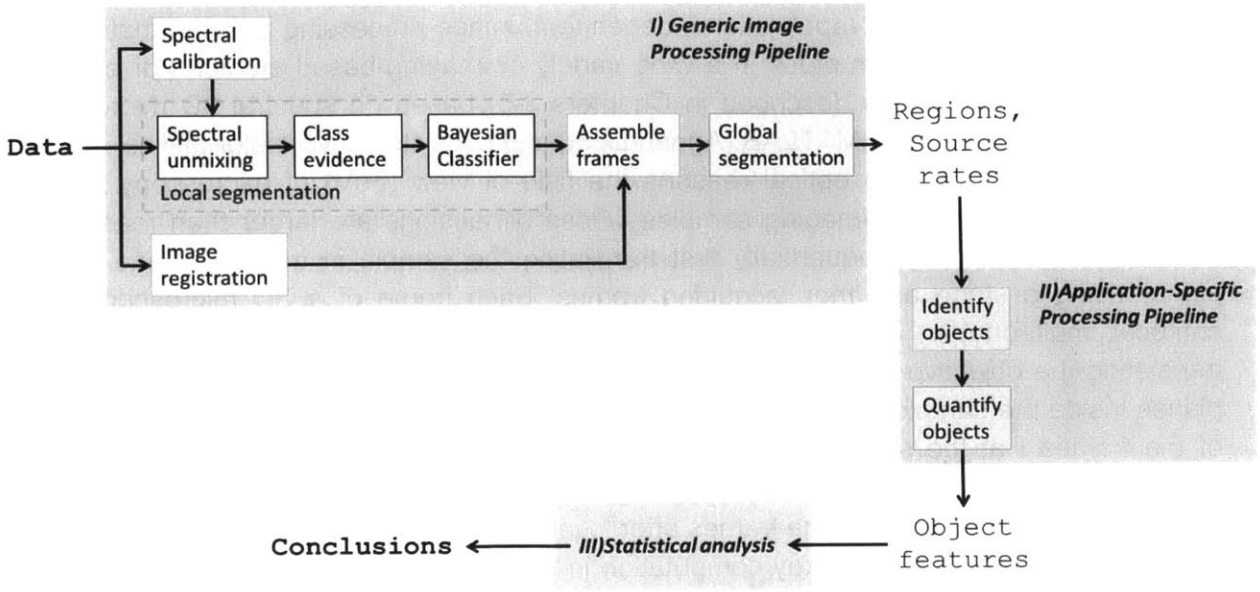


Figure 2.1.2: Schematic of the key steps of an imaging-based study, which converts spectral fluorescence microscopy data into evidence/conclusions. i) a generic image processing pipeline that converts raw data into regions and source emission rates, ii) an application specific pipeline that converts the outcome of the first part into objects and quantifies them using a series of features, and iii) appropriate statistical analysis that integrates imaging results with biological knowledge in order to provide evidence and conclusions. Chapter 2 describes a generic image processing pipeline (first part).

Required Input from the User

The developed image processing pipeline requires the following input from the user:

1. Data acquisition details (e.g. number of frames, stage raster scanning pattern)
2. Number and identity of classes.
3. Number and identity of emission sources present at each class.
4. An approximate knowledge on the upper and lower limits on the emission rate of each source of each class. If these limits are not known in advance they can be approximated based on the outcome of a spectral unmixing step that includes all sources.

2.1.4 Chapter Description

Section 2.2 provides background on multi-photon microscopy, spectral microscopy, and the state-of-the art on the computation tasks related processing spectral fluorescence microscopy data. Section 2.3 describes in detail the instrument utilized in this thesis, and its calibration. Section 2.4 describes in detail the spectral unmixing task, the backbone of pipeline. It discusses the underlying mixing models, numerical implementation of ML estimation, theoretic estimations on the estimation accuracy and precision in the presence and absence of modeling error, and provides simulation results. Section 2.5 describes the local and global image segmentation tasks. Section 2.6 provides examples of the application of the developed image processing pipeline in the processing of several biological samples imaged by spectral multi-photon microscopy.

2.2 Background: Processing Spectral Multi-photon Microscopy Data

2.2.1 Multi-photon Microscopy

In fluorescence imaging contrast is generated by fluorophores, molecules that emit light upon light excitation. Due to energy losses, the energy of each emitted photon is slightly less compared to the energy of the incident photon that excited the fluorophore, causing the wavelength of emission light to be longer than the wavelength of excitation light (Stokes shift) [Lakowicz 2006]. Fluorescent microscopes utilize this shift to generate low background images by blocking the emission light without affecting the emission. Multi-photon microscopes refers to a special class of instruments where fluorophores are excited by the near simultaneous absorption of two low-energy photons, instead of one high energy photon [Goeppert-Mayer 1931; Denk et al. 1990].

In multi-photon microscopy (MPM), the emission light signal is generated by particular molecules in the sample (denoted as "sources") that are excited by incident light through two mechanisms that depend non-linearly on the intensity of the excitation light [Zipfel et al. 2003]: i) multi-photon excitation fluorescence, where fluorophores are excited by the near-simultaneous absorption of two or more photons [Goeppert-Mayer 1931; Denk et al. 1990], and ii) higher harmonic generation, where illumination light is coherently scattered into light of integer-multitude frequency [Gannaway and Sheppard 1978; Campagnola and Loew 2003; Chen, Nadiarynkh et al. 2012]. The emissions detected in this thesis are originated by the two most common nonlinear optical mechanisms: two-photon excitation fluorescence (TPEF), and second harmonic (SH) emission. Although usually the same instrument can excite and detect both TPEF and SH, [Zipfel et al. 2003], for simplicity only the term MPM is used.

The non-linear dependence of the signal on excitation light intensity, and the very small cross-sections of nonlinear optical phenomena (TPEF, SH) create the need for extremely large excitation light intensities for efficient source excitation. At the same time, the incident light should not damage the samples. Both contradicting requirements can be satisfied when a femtosecond pulsed laser is focused inside a sample by a high NA objective. A critical feature of MPM is that the vast majority of emission originates within a small region surrounding the excitation laser focus[1] (Figure 2.2.1). This provides the optical sectioning capability that is necessary for 3D imaging, without the need for a spatial filter (pinhole) that is used in confocal imaging [Denk et al. 1990].

The two major advantages of multi-photon imaging compared to comparable 3D imaging methods (confocal imaging) are: i) the ability to image deeper inside biological samples due to the IR excitation wavelength used (less prone to scattering) and due to the little amount of out-of-focus light generated by the incident excitation light [Denk et al. 1990; Helmchen and Denk 2005], ii) the reduced photo-toxicity caused to the biological samples, again due to the use of IR excitation light. Therefore, most applications of MPM lie in the area of *in vivo* imaging in animal models [So et al. 2000]. However, this thesis takes advantage of another feature of two-photon excitation that is very useful for spectral microscopy: multi-photon excitation has broader cross-

[1] An interesting exception of the multiphoton signal localization takes place in very deep-tissue imaging in scattering samples [Helmchen and Denk 2005], a case that is not considered in this study but is discussed in Sections 2.4.5 and 2.4.7.

section peaks compared to single-photon excitation, making possible to simultaneously excite more fluorophores of different emission color (Figure 2.2.2).

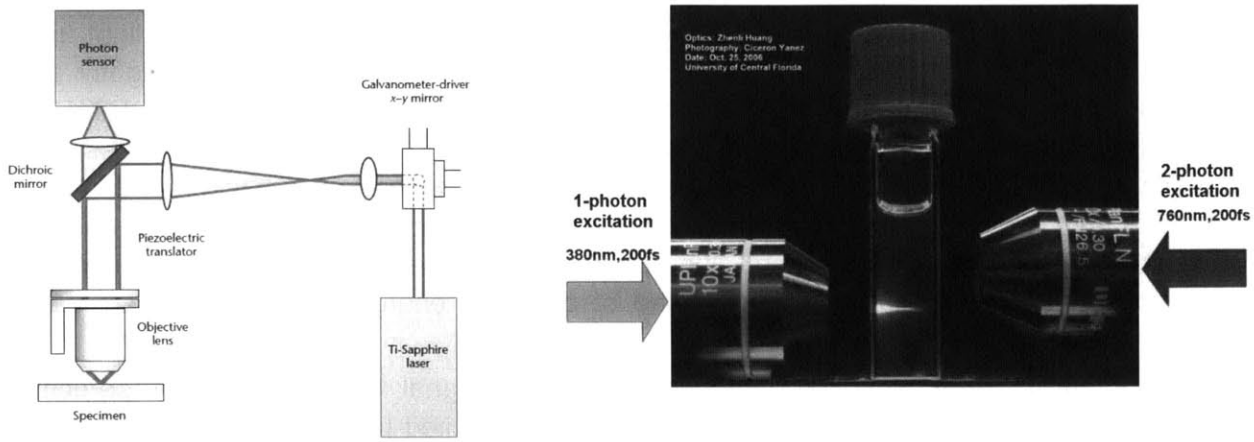


Figure 2.2.1: Principles of two-photon excitation microscopy. **Left**: Basic schematic of a two-photon microscope. A beam of IR femtosecond laser (red lines) is focused inside a sample by a high NA objective lens. Emission (green lines) is collected by the same objective, separated in a dichroic mirror and detected in a sensor. 3D imaging is acquired by controlling the position of excitation light focus using x-y galvanometric mirrors and a piezoelectric objective actuator [So 2002]. **Right**: Two-photon excitation is localized around the focus (top) resulting in intrinsic optical sectioning. In contrast, in single-photon excitation there is significant out-of focus light, that needs to be removed by optical filtering [Belfied research group, University of central Florida, http://chemistry.cos.ucf.edu/belfield]

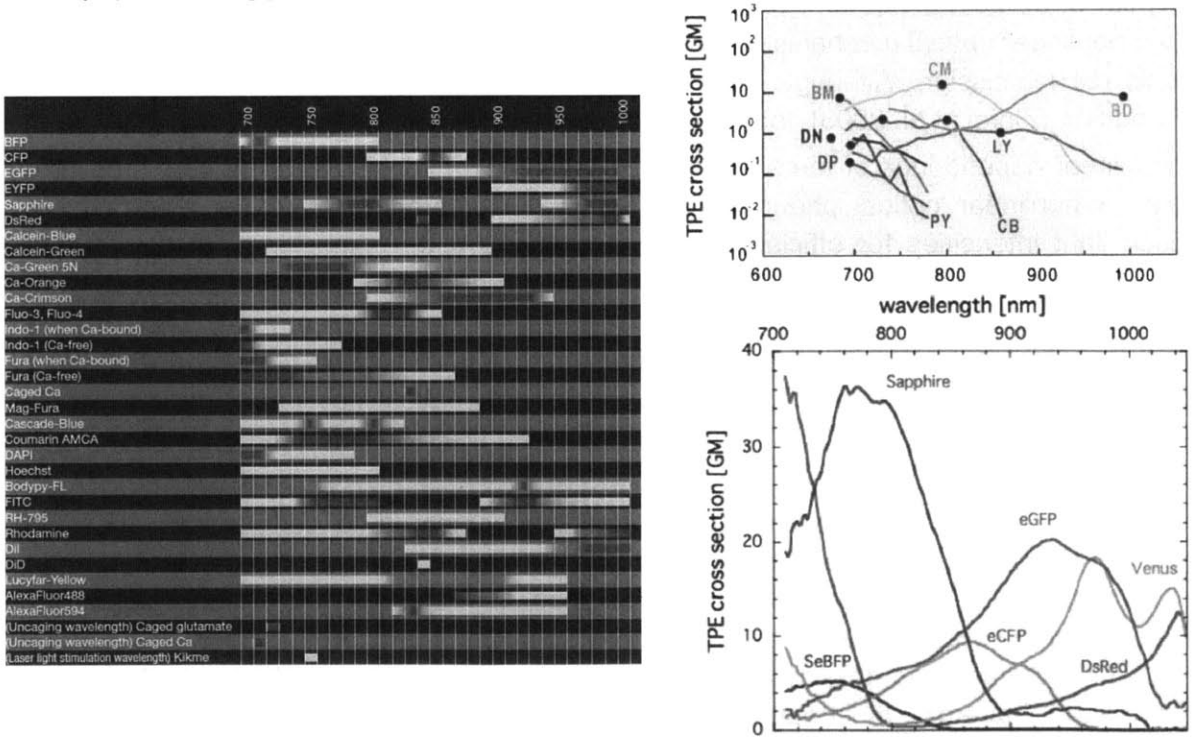


Figure 2.2.2: A major advantage of multi-photon microscopy of particular interest to spectral imaging, is the broader peaks of two-photon excitation. **Left**: Optimal two-photon excitation wavelengths for several fluorescent dyes [Olympus Corporation, http://www.olympusmicro.com/]. **Top Right**: Two-photon excitation spectra (in GM) of several fluorescent dyes as a function of excitation wavelength. BM: p-bis(omethylstyryl)benzene, CB: Cascade Blue, LY: Lucifer yellow; BD: Bodipy, DP: DAPI not bound to DNA, DN: (dansyl), PY: 1,2-bis-(1-pyrene decanoyl)-sn-glycero-3-phosphocholine, CM: coumarin 307. [Xu et al. 1996]. **Bottom Right**: Two-photon excitation spectra (in GM) of several fluorescent protein estimated by Fourier-transform spectroscopy [Hashimoto et al. 2010].

2.2.2 Spectral Microscopy Instrumentation

The standard practice in multi-label fluorescence imaging is to acquire several exposures for each image sequentially. At each exposure, an appropriate choice of excitation light wavelength and emission filters ensures that only one fluorophore contributes to the detected signal. Such sequential approach limits acquisition speed and can cause photo-bleaching. Furthermore, due to the large width of fluorophore emission spectra peaks, the number of fluorophores that can be imaged with minimum cross-talk is limited [Zimmerman 2005]. Spectral imaging bypasses many of these problems by acquiring a single exposure for each image, using a multiple-channel detector. Every channel of the detector detects signal within a particular region of the EM spectrum. Subsequent processing of the acquired data aimS to resolve the emission of the sources that contribute to this signal based on the different emission spectra of each source. Spectral imaging can work even in the case of multiple sources of overlapping emission spectra, albeit with decreasing accuracy as this overlap increases.

Spectral fluorescence microscopy can be implemented through several different configurations [Hiraoka et al. 2002; Berg 2004; Garini et al. 2006; Rietdorf and Stelzer 2006; Burton et al. 2009]. The key difference of each configuration is the way the emission optics resolve the emission light by its wavelength. *Wavelength-scanning configurations* use a fixed or tunable optical filter (circular-variable, liquid crystal, acousto-optical) to allow a particular wavelength region reach the optical detector [Wachman et al. 1997, Lansdford et al. 2001]. In this case, the same detector acquires all channels sequentially. *Spatial-scanning configurations* resolve spectral components using a prism or a diffraction grating into a multi-channel optical detector, which acquires all channels in parallel [Haraguchi et al 2002; Berg 2002]. *Temporal-scanning configurations* extract the emission spectrum of the detected signal by post-imaging Fourier spectroscopy [Chamberlain 1978; Malik et al. 1996; Tsurui et al. 2000; Hashimoto et al. 2010].

Published configurations of spectral fluorescence microscopes include single-photon excitation microscopes (in confocal detection mode [Haraguchi et al. 2002; Sinclair et al. 2006] or wide-field detection mode [Tsurui et al. 2000]) and multi-photon microscopes (MPM) [Landsford et al. 2001; Buehler et al. 2005; Im et al. 2010]. Compared to single-photon excitation wide-field detection systems, confocal and MPM offer the ability of 3D spatial resolution. Specifically for spectral imaging applications, MPM offer several advantages compared to single-photon confocal microscopes: i) two-photon excitation has broader cross-section peaks compared to single-photon, making possible to simultaneously excite more fluorophores of different emission color, therefore increasing instrument throughput (for example CFP, GFP and YFP can be simultaneously excited by 880nm light, although not equally efficiently), Figure 2.2.2 [Xu and Webb 1996; Xu et al. 1996; Bestvater et al. 2002], ii) two-photon excitation microscopy utilizes a dichroic filter of a single cut-off frequency. Confocal microscopes that utilize multiple laser lines to excite multiple fluorophores need dichroic mirrors of complex transmission curve that can complicate the accurate estimation of the spectral signatures of the sources [Berg 2004].

There are several commercially available spectral microscopes with 3D resolution. All of them are available in both single-photon confocal and MP excitation versions. The Radiance Rainbow system (Biorad, Hercules, CA) utilizes dichroic and acousto-optical filters similar to the instrument described in [Davis and Shen 2007]. The TCS SP system (Leica Microsystems, Wetzlar, Germany) utilizes a dispersion grating and reflecting mirrors. The META system (Carl Zeiss, Oberkochen, Germany) utilizes a spectrograph and a multi-anode PMT [Haraguchi et al.

2002; Berg 2004], similar to the system used in this thesis (Section 2.3). Sandia National Labs commercialize a spectral microscope of high spectral resolution and dynamic range that utilizes EMCCD progressive scanning [Sinclair et al. 2006]. Although the original design was developed for confocal imaging, the same configuration has been applied in multi-photon imaging [Im et al. 2010]. Fourier spectroscopy can also be applied in widefield fluorescence microscopy microscopes using commercially available interferometric imaging spectrometers and CCDs [Tsurui et al. 2000].

Spectral microscopes use either CCDs or PMTs to detect light emission. CCD provide analogue output readings. PMT output can be either in analog format or in photon counting mode [Hakamata et. al. 2006], the later providing the best possible SNR and therefore is the most appropriate configuration for imaging very weak signals.

2.2.3 Spectral Unmixing of Fluorescent Spectral Microscopy Images

Models and Algorithms

Spectral unmixing is a key computation step for spectral microscopy images. Spectral unmixing provides the emission contribution of the individual sources present in the image. Almost all publications that analyze fluorescent spectral microscopy data, describe the detected emission $\boldsymbol{y}$ as a linear combination of the spectral signatures of the sources, described by the spectral matrix $\boldsymbol{S}$ [Tsurui et al. 2000; Landsford et al. 2001]:

$$\boldsymbol{S} \cdot \boldsymbol{\lambda} = \boldsymbol{y} \qquad [2.2\text{-}1]$$

This system of equations is related (but not identical) to the ML solution of the Gaussian common noise mixing model (Section 2.4.4), which assumes equal noise in all channels [Tsurui et al. 2000]. Since usually the number or detector channels M is larger than the number of emission sources N, Eq. 2.2-1 is usually solved as a least-squares (LS) problem, using the left pseudo-inverse of the spectral matrix $\boldsymbol{S}$ calculated by SVD [Tsurui et al. 2000]. Many references refer to this solution of Eq. 2.2-1 as "linear unmixing. This left pseudo-inverse solution does not impose non-negativity constraints, however it seems that in most applications this is not a problem, probably due to high signal level and good knowledge of the sources' spectral signature. Few studies utilize the non-negative least-squares (NNLS) method, which incorporates non-negativity constraints, to solve the linear unmixing problem [Lawson and Hanson 1974; Radosevich et al. 2008].

Few studies have utilized the Poisson observation model that incorporates the nature of photon arrival in optical detections (Section 2.4.3), and therefore is expected to be more accurate for low signal level images. The main reason is that the ML estimate from the Poisson mixing model cannot be derived analytically, and can only be solved numerically using an iterative numerical algorithm. [Davis and Shen 2007] claim to solve the nonlinear mixing models using a proprietary non-iterative algorithm that is more accurate in low signal levels (less than 100 photons per pixel) compared to Eq. 2.2-1, but provide no information about the algorithm. The same paper comments that the full additivity constraint may induce errors in the low photon-count regime but this is not the case because the EM solution to the Poisson mixing model satisfies the full additivity constraint automatically [Shepp and Vardi 1982]. The ML estimate of related Poisson mixing models in PET and SPECT imaging has been calculated using variations of the established expectation-maximization algorithm [Dempster et al. 1977; Bilmes

1998]. [Shepp and Vardi 1982] proposed an iterative EM-like algorithm that automatically satisfies the positivity constraints, and provide a proof for the convexity of the likelihood function. [Tsui et al. 1991] compared the performance of the EM-like algorithm with the performance of the WLS linear model solved using a conjugate gradient technique. [Kaufman 1993] demonstrated that this EM-like algorithm is a scaled steepest ascent algorithm and equivalent to a continuously re-weighted WLS, and proposed a preconditioned conjugate-gradient algorithm to solve the WLS problem that satisfies non-negativity. Although the EM-like algorithm of [Shepp and Vardi 1982] can be applied in the nonlinear fluorescent mixing model, fluorescent imaging data are different than PET or SPECT data, which affect the choice of the appropriate numerical implementation (Section 2.4.3).

A large part of the literature on spectral unmixing comes from the field of remote sensing (RS), where multi-channel satellite images (spanning the visible and infrared EM regions) are used to study the surface of earth [Varshney and Arora 2004; Richards and Jia 2006]. [Keshava, 2003] provides an extensive overview of spectral unmixing algorithms for RS. Spectral unmixing for RS consists of three steps: i) dimension reduction, ii) end-member estimation, iii) inversion. Dimension reduction methods (principal component analysis (PCA), maximum noise fraction) are critical for RS applications because of the large number of channels (bands) M that makes computation expensive. End-member determination methods (minimum volume transforms, fuzzy κ-means clustering, non-linear least-squares, independent component analysis (ICA)) identify the spectra of the sources that contribute in the image. Inversion methods (least squares (LS) with or without additivity constraints, non-negative least squares (NNLS), nonlinear least-squares, and maximum a posteriori (MAP) estimators) identify the contribution of each component based on the detected signal and the source emission spectra.

Although some of the methods developed for RS applications can be used to process spectral fluorescence microscopy data, there are several important differences that affect the implementation of spectral unmixing. Specifically, in spectral fluorescence microscopy

i) the number of channels M is larger than the number of sources N, but not large enough to make dimension reduction necessary.
ii) there is much less uncertainty about the number and emission spectra of the "sources" that contribute to the signal. therefore the end-member determination step is usually also omitted
iii) the signal level per channel can be quite low, sometimes approaching the photon-counting regime. This affects the precision and accuracy of spectral estimation and affects the choice of the appropriate algorithm for ML estimation.

Performance of Spectral Unmixing Algorithms

Several authors have attempted to describe how various factors affect the validity of spectral unmixing calculations. Neher and Neher 2004] proposed to describe the performance of spectral unmixing using a Figure of Merit (FoM), defined as the square of the ratio of SNR in the presence of spectral mixing divided by the SNR in the absence of spectral mixing. Several papers describe qualitatively that the accuracy of spectral unmixing calculations depends on the degree of similarity between the emission spectra of sources [Lansdford et al. 2001], and accurate knowledge of the spectral signatures of the sources present in the sample (Section 2.4.5). The accuracy of spectral unmixing estimation is correlated to the determinant of the matrix of the Fisher Info matrix of the system, which depends on the spectral signature of the

sources [Neher and Neher 2004]. Although the emission spectra of many chemical elements are known, spectral unmixing requires the spectral signature of sources, which are either measured experimentally or estimated them based on the properties of detection optics and detectors (Section 2.3.7) [Tsurui et al. 2000; Berg 2004].

The number of stains used to treat a sample is not necessary equal to the number of sources that can simultaneously contribute to the signal detected in a pixel. In many cases fluorophores are localized in different compartments of the sample. In this case including all sources present in the sample in the spectral unmixing calculations of each pixel induces significant estimation errors that have not been addressed in the literature. [Zimmermann 2005] provides simulations that evaluate the effect of uncertainty in the number of sources present in the sample. It suggests that including additional sources does not induce significant estimation error because spectral unmixing will assign to the extra sources a negligible contribution. Based on the discussion of Sections 2.4.5-7 this seems reasonable only when the signal level is large. On the other hand, omitting sources from the spectral unmixing calculation can cause serious estimation errors. [Keshava 2000] suggests that an estimation of the number of components can be derived based on the number of eigenvalues of the data matrix that are significantly different than zero.

Several publications suggest optimum detector characteristics (number of channels, channel spectral width) for particular spectral imaging application. Utilizing detectors of very large channel number and small spectral detection width per channel is not advantageous, since the photon count per channel decreases [Neher and Neher 2004; Zimmerman 2005]. The optimal spectral detection windows for optimal spectral unmixing estimation accuracy depends on the magnitude of background noise, the magnitude and ratio of fluorescent emissions, and the shape of emission spectra [Neher and Neher 2004].

Most studies that have utilized spectral microscopy studied bright samples, as evident by the published sharp images that lack the characteristic noise pattern of Poisson noise. Even papers that suggest that the accuracy of the spectral unmixing calculation depends on the fluorescent signal level [Lansdford et al. 2001] do not address the case of low signal level, where Poisson noise makes spectral unmixing and classification less precise and accurate. [Davis and Shen 2007] apply spectral unmixing in relatively dim samples to resolve the emission of three dyes whose emission peaks lie within 20nm (Texas red, alexa fluor 610, alexa fluor 633). Results suggest estimation error is significant when the total photon count is N=50 photons/pixel but much smaller for N=300 photons/pixel. However, the results of such studies cannot be generalized to other systems. Instead, an analytic approach is necessary to derive performance estimations for any set of sources of interest.

Challenging Applications of Spectral Imaging

Several applications of spectral microscopy are reviewed in [Zimmerman et al. 2005; Garini et al. 2006]. Spectral microscopy has been applied to quantify FRET in live cells [Hiraoka et al. 2002, Haraguchi et al. 2002, Zimmermann 2005]. Published challenging applications include seven-color immunofluorescent imaging of tissue sections [Tsurui et al. 2000], separation of quantum dot signal from tissue autofluorescence in *in vivo* mouse imaging [Gao et al. 2004], eight-color multiple-filter leucocyte cytometry [Mittag et al. 2005], spectral karyotyping [Garini et al. 1996], spectral unmixing of four components (CFP, GFP, YFP, DiI) or the two very similar

dyes GFP & fluorescein [Landsford et al. 2001]. Multiple exposures were often required for imaging those samples than contain many sources. On the other hand, reducing the number of exposures is crucial for minimizing photo-bleaching and photo-damage [Zimmermann 2005]. One way to reduce exposes is by using elaborate dichroic mirrors: [Haraguchi et al. 2002] use one exposure to excite four dyes whose peaks lie in the [515, 645 nm] region. Multi-photon excitation can also excite more stains simultaneously due to the broader two-photon cross section peaks of fluorophores. [Sinclair et al. 2006] imaged simultaneously five dyes whose spectra lie mostly in the 500-800 nm range excited by a single 488nm laser line and detected by a novel EMCCD-based confocal microscope. However, three of the dyes were excited indirectly through FRET. [Vermaas et al. 2008] imaged cyanobacteria cells via spectral confocal microscopy and resolved the presence and localization of six fluorescence components within the [500, 800] nm range. All fluorophores were excited by a single 488nm laser. The emission spectra of four fluorophores lie within 50nm of the EM spectrum.

Blind Signal Separation

In contrast to spectral unmixing methods that estimate the emission contributions of all sources assuming that the spectral signatures of all sources are known, blind signal separation (BSS) methods attempt to estimate simultaneously the detected emission spectra of the sources and their contributions. BSS methods include independent component analysis (ICA) [Hyvärinen et al. 2001], non-negative matrix factorization (NMF) [Lee and Seung 1999, Berrya et al. 2007], alternative least squares (ALS) [Keenan et al. 2002; van Benthem et al. 2002; Berrya et al. 2007], and expectation maximization (EM) [Fish et al. 1995]. BSS methods require multiple measurements (or global analysis of images), and rely on assumptions regarding the nature of the spectral matrix $\boldsymbol{S}$ and the emission rate vector $\boldsymbol{\lambda}$. BSS algorithms are more complex and sensitive to noise compared to spectral unmixing, however they can be useful in analyzing samples that contain unknown sources, or samples where the emission spectrum of sources can vary (e.g. due to interactions with other molecules, differences in the pH of the surrounding environment or due to scattering in deep tissues that could affect the detected emission spectra).

2.2.4 Local Image Segmentation

Image segmentation is the computation task that partitions a digitized image into discrete regions. When regions have a particular physical meaning, they can be classified into one of a set of possible region types (e.g. cells, fibers) referred to as "classes".

Image segmentation is a critical task for automated high-throughput imaging-based studies of cells and tissues that rely on quantifying accurately the shape of features (e.g. cytoplasm, axons), such as image informatics (Chapter 3), and connectomics [Glory and Murphy 2007; Jain et al. 2010; Hangwood et el 2012]. The outcome of these methodologies depends critically on the ability to accurately identify cells, estimate their boundaries, and then quantify various spatial metrics of their morphology and emission (see Chapter 3). Image segmentation can be challenging due to several reasons, including intensity variations, sample variations, presence of noise, and complex morphology.

Image segmentation methods can be classified as "local" (pixel-wide) and "global" (region-wide). Local segmentation methods assign each pixel to a class based only on information (intensity, gradient, detected spectrum) from this pixel. Global segmentation methods process regions of

pixel and fuse local pixel information with contextual spatial information, which suggests that neighboring pixels are most likely to be of the same type (class). Section 2.2.4 provides background on local segmentation methods. Section 2.2.5 provides background on Markov Random Fields, a class of mathematical models that are utilized in this thesis for global image segmentation.

Algorithms for Segmentation of Single-Channel Images

A large number of algorithms have been developed for segmenting single-channel (intensity) images into distinct regions. These algorithms segment images based on particular features (intensity, intensity gradient, and object shape) and can be classified into two families:

The first family of algorithms segment images into regions based on the assumption that different regions have distinct and relatively homogenous intensity signals.

- Intensity thresholding, the simplest approach, classifies the pixels of an image based on the pixel intensity and (one or many) threshold values. There are several approaches to choose the threshold value, including Otsu's method (minimize in-class variance, maximize inter-class variance) [Otsu 1979], and the related watershed transformation of the histogram [Soille 2003]. Thresholding can be implemented globally (a single threshold for the whole image) or locally (a distinct threshold for every image region). When thresholding works it is fast and robust. However, thresholding could fail due to artifacts in the image or region signal variations, and does not consider any information regarding the morphology of the regions.
- Region growing algorithms grow homogenous regions based on similarity metrics between neighboring pixels until all pixels are assigned to a region. A very popular algorithm is the watershed algorithm. Watersheding is used to segment a region into contacting regions based on particular geodesic distances defined in the region [Soille 2003]. Watersheding can be applied either to an intensity image, or to the gradient of an image, or to the distance transform of a binary mask. It is critical to make a correct initial estimate of region seeds in order to avoid over-segmentation or under-segmentation.

The second family of algorithms segment images into regions by detecting region boundaries. These algorithms assume that large intensity variations take place at region edges.

- Edge detection algorithms calculate the gradient of the image intensity, and then identify boundaries based on the magnitude of the gradient. Examples include Canny's algorithm [Canny 1986]. Edge detection algorithms methods are sensitive to noise, and sometimes provide edges that are not connected.
- Active contour methods (also called "snakes") are methods that identify the outline of objects in noisy images. A snake is an energy-minimizing spline guided by external constraint forces and image-based potentials that pull it towards particular features such as edges [Kass et al. 1988]. Over the past fifteen years several implementations (gradient vector flow active contours [Xu and Prince 1997], diffusion snakes [Cremers et al. 2002], and geometric active contours [Caselles et al. 1997]) have been applied in segmenting several kinds of grayscale images.

A distinct set of algorithms utilized for image segmentation are clustering algorithms (e.g. k-means clustering [Ng et al. 2006]), which group pixels into sets of similar intensity properties.

Clustering algorithms belong to the field of unsupervised learning, and are preferred in applications of limited a-priori information on the properties of the classes.

Algorithms for Segmentation of Spectral Fluorescence Images

Different kinds of objects present in samples usually contain distinct sets of sources of distinct spectral signatures. Therefore different image classes (each corresponding to a corresponding to different kind of object) usually have distinct spectroscopic properties. This additional information provided by the detected spectrum $\boldsymbol{y}$ at each pixel can be used to enhance segmentation performance.

Most local segmentation approaches utilize supervised classification algorithms, very similar to the ones developed for processing multi-spectral Remote Sensing (RS) hyper[2]-spectral data such as Bayesian classification, minimum-distance classification, and Mahalanobis classification [Jackson and Landgrebe 2002; Varshney and Arora 2004; Richards and Jia 2006]. For example [Landsford et al. 2001] utilized the Mahalanobis classifier of the commercial software ENVI developed for RS applications. Brute force application of methods developed for RS to process fluorescent images may not be optimal. RS images of earth derived by satellites contain a wide range of vegetation, soil types, and human-made features, whose spectroscopic properties are hard to model and may depend on ambient light. Therefore, in RS applications the spectral properties of each class need to be estimated based on a training sample [Richards and Jia 2006; Bai et al. 2013]. On the other hand, in fluorescence imaging usually there is more concrete information about the sources present in the samples and their spectroscopic properties, enabling a more parametric application of Bayesian classification (Section 2.5.2).

Several related methods classify each pixel based on the "similarity" of its signal $\boldsymbol{y}$ with "reference" spectra, each corresponding to a particular class. Spectral similarity is evaluated using the Eucledian distance, the spectral angle measure (SAM) [Landsford et al. 2001, Garini et al. 2006], or an entropy-based criterion (SPD) [Chang 2000].

Validation of Image Segmentation Algorithms

The performance of image segmentation algorithms is evaluated either by simulations, or by comparing the outcome of the segmentation algorithm with a "ground truth" segmentation, obtained usually by manual segmentation [Hangwood et el 2012]. There are two approaches for quantifying the performance of image segmentation [Jain et al. 2010]:

- quantify the ability of a segmentation algorithm to assign each pixel the correct class corresponding to the kind of object imaged in that pixel. This approach treats image segmentation as a labeling problem, and quantifies the performance using several metrics such as the percentage of pixels assigned to the correct class [Hangwood et el 2012], the Rand index [Rand 1971; Coelho et al. 2009], the Jaccard similarity index [Dima et al. 2011], the bivariate similarity index [Dima et al. 2011], and the error matrix [Varshney and Arora 2004]. This approach seems appropriate for evaluating the performance of local image segmentation algorithms.
- quantify the ability of a segmentation algorithm to partition the image correctly into regions that make physical sense without topological errors. This approach incorporates spatial

[2] In remote sensing the term "hyper-spectral" is used instead of the term "spectral" to denote that the imaging bands consist of both visible and infrared wavelengths

context and gives emphasis on i) identifying correctly the bulk presence of objects, ii) identifying correctly the borders between regions, iii) preventing invalid neighboring object encounters, and iv) avoiding topologic mistakes. Sample metrics include the "warping error" and the "rand" error described in [Jain et al. 2010]. This approach seems appropriate for evaluating the performance of global image segmentation algorithms that incorporate spatial context information, see also [Kerfoot and Bresler 1999].

Published studies suggest that there is no algorithm that outperforms other in all applications. Segmentation accuracy depends on several factors including signal level, texture, noise, and shape of objects [Hangwood et el 2012]. At the moment there is no study that evaluates the performance of image segmentation algorithms for spectral fluorescence images, particularly in the case of low signal level such as the one encountered in this thesis.

2.2.5 Global Image Segmentation by Markov Random Fields

A Markov Random Field (MRF) is a set of random variables $\mathcal{F} = [F_1, F_2, \dots, F_m]$ defined on an undirected graph $\mathcal{G}$ (a set of sites $\mathcal{S}$ with respect to a neighborhood system $\mathcal{N}$) that satisfy the positivity and Markov property [Li 2009; Koller and Friedman 2009]. Each F_i corresponds to a particular site (node) of the undirected graph. The value f_i of each random variable F_i can take values from a label set $\mathcal{L}$. A particular configuration $F = \{f\}$ of the MRF corresponds to the joint event that variables $[F_1, F_2, \dots, F_m]$ take values $\{f\} = [f_1, f_2, \dots, f_m]$.

The purpose of MRF is to model spatial context and spatial interaction between the random variables F_i located at neighboring sites of the graph $\mathcal{G}$. Image segmentation is a straightforward application of MRF because usually the context of neighboring pixels is correlated. The site set $\mathcal{S}$ corresponds to image pixels, the discrete-value label set $\mathcal{L} = \{1,2, \dots, D\}$ corresponds to the D classes present in the sample, and the neighborhood system $\mathcal{N}$ is chosen based on modeling and computational factors. The joint probability $\mathbb{P}(\{f\}|\{\boldsymbol{y}_i\})$ of a configuration $\{f\}$ given an image $\{\boldsymbol{y}_i\}$ ($i = 1,2, \dots, m$ correspond to the m pixels of the image) is calculated via the Hammersley-Clifford theorem as [Hammersley and Clifford 1971; Li 2009]:

$$\mathbb{P}(\{f\}|\{\boldsymbol{y}_i\}) = \frac{1}{Z}\exp(-U(\{f\}|\{\boldsymbol{y}_i\}))$$

$$U(\{f\}|\{\boldsymbol{y}_i\}) = U(\{\boldsymbol{y}_i\}|\{f\}) + U(\{f\})$$

Where $U(\{f\}|\{\boldsymbol{y}_i\})$ is the posterior energy, which equals the sum of the prior energy $U(\{f\})$ and the likelihood energy $U(\{\boldsymbol{y}_i\}|\{f\})$. Prior energy $U(\{f\})$ models spatial context information. Likelihood energy $U(\{\boldsymbol{y}_i\}|\{f\})$ incorporates the information available from the measurements $\{\boldsymbol{y}_i\}$ (the image itself). The solution of a MRF model usually involves finding the maximum a posteriori (MAP) configuration $\{f\}_{MAP}$ that maximizes $\mathbb{P}(\{f\}|\{\boldsymbol{y}_i\})$ or equivalently minimizing the posterior energy $U(\{f\}|\{\boldsymbol{y}_i\})$. Different MRF models different in the choice of the graph $\mathcal{G}$, and the expression of the prior $U(\{f\})$ and likelihood $U(\{\boldsymbol{y}_i\}|\{f\})$ as a function of configuration $\{f\}$.

Modeling

Image segmentation is one kind of "pixel labeling" problems that can be modeled and solved using Markov Random Fields (MRF) [Strauss 1977]. Models usually describe MRF energy using a auto-binomial model [Li 2009], use 4-neighbor systems, and model data uncertainty using Gaussian uncorrelated noise [Elliott et al. 1984; Dubes et al. 1990]. When there are only two labels, the corresponding labeling problem (an Ising model) can be solved exactly using graph

cuts. However, when there are more than 2 labels, the problem is NP-hard (non-deterministic polynomial-time hard), making the vast majority of MRF-based energy function intractable [Szeliski et al. 2008].

The parameters of a MRF model can be estimated by maximum-likelihood via the coding method [Besag 1974; Dubes et al. 1990]. More advanced algorithms attempt to estimate simultaneously the parameters of the MRF model and solve the MRF-based imaging problem [Li 2009]. Such methods include the double MRF model of [Marroquin et al. 2003].

More complex modeling approaches for segmenting grayscale images include double hierarchical MRF [Cohen and Cooper 1987], where a Gaussian MRF is used to model texture within image regions, and an auto-binary MRF is used to describe a-priori information about region location and geometry.

Algorithms

Solving a MRF for image segmentation based on the MAP framework is equivalent to the nonlinear integer optimization problem of finding the configuration $\{f\}$ that maximizes the posterior $\mathbb{P}(\{f\}|\{\boldsymbol{y}_i\})$. When there are more than $D > 2$ labels, this is an NP-hard problem, a hard computation task even in the case of modest number of nodes due to the non-convexity of the cost function and the presence of multiple local minima. The initial algorithms utilized for solving MRF are Simulated Annealing (SA) [Geman and Geman 1984], maximizer of posterior marginal [Marroquin et al. 1987], and Iterated Conditional Modes (ICM) [Besag 1986]. SA utilizes a thermodynamics-based approach to "cool" progressively a temperature parameter T in the MRF energy field and avoid local minima. SA is proven to be able to find the global optimum of MRF energy fields, however it is very computationally intense and therefore not applicable to process large imaging data files quickly. ICM, one of the earliest and simplest methods, starts from an initial estimate of pixel labeling and updates each pixel label sequentially. Although ICM is not very computationally expensive, it suffers from ineffectiveness and its results are very sensitive on the initial pixel labeling.

During the past fifteen years several new methods have been developed to solve MRF problems more efficiently. These methods include Belief Propagation (BP), Graph Cuts (GC), and Tree-Reweighted Message Passing (TRWP). The performance of these methods compared to ICM is compared in [Szeliski et al. 2008]. At the moment, most state-of-the art applications of MRF utilize GC or BP. Graph cuts algorithms (such as "swap-move", "expansion move", "max-flow") are describes as "very large neighborhood search techniques" that iteratively compute the global minimum of elementary binary problems [Boykov et al. 2001]. Both GC and TRWP algorithms share several features with BP, the method utilized in this thesis. TRWP algorithms also use half the memory required by BP and provide lower bounds of energy that can be used to assess the quality of a MAP solution [Wainright et al. 2003, Kolmogorov 2006; Szeliski et al. 2008].

Belief propagation (BP) methods [Pearl 1988, Yedidia et al. 2000] include the sum-product and max-product algorithms (or equivalently sum-sum and max-sum when energy is expressed using the enrgy function $U(\{f\}|\{\boldsymbol{y}_i\})$). The sum-product algorithm calculates the marginal probability at each node, while the max-product algorithm calculates the MAP probability of the whole graph. BP methods are based on passing "believes" between nodes. Believes are vectors that contain information about the optimal configuration. In graphs that do not contain loops,

believes can be calculated either analytically (following the graph structure) or iteratively. In loopy graphs, it is not possible to calculate believes analytically, so an iterative scheme is the only way to estimate them [MacKay 2003]. BP was initially developed for graphs without loops [Pearl 1988], where the iterative calculation of believes converges to the optimal result within N iterations. In loopy graphs BP provides an approximate solution for the most likely graph labeling [Weiss and Freeman 2001, MacKay 2003; Li 2009]. Even though there is no analytic proof that in loopy graphs BP converges to the MAP solution of the MRF, BP methods empirically are considered to provide good approximate solutions, apart from rare cases where the algorithm does not converge and gets trapped into an infinite loop [Koller and Friedman 2009]. The iterative calculation of believes is the most expensive step of BP methods. Several themes of belief propagation have been developed in order to accelerate believes calculation. Instead of the synchronous calculation of believes in the whole graph utilized in normal BP, accelerated belief calculation schemes iterate along rows and then along columns [Tappen and Freeman 2003]. More tricks that accelerate BP methods (better initialization of believes via a pyramid-like scheme) can be found in [Felzenszwalb and Huttenlocher 2006, Szeliski et al. 2008].

Performance

The performance of an algorithm that solves a MRF is evaluated based on its computation cost and the ability to find the optimal or a near-optimal solution in a simulated problem of known global solution. Several papers that compare the performance of MRF algorithms in real images compare the final estimated potential U (the energy that corresponds to the solution), implying that lower energy corresponds to better solutions. [Dubes et al. 1990] suggests that this is not a reliable indicator of accuracy, because errors in the MRF modeling procedure can result in final estimated potential U that is lower than the energy U of the ground truth. A better indicator of image segmentation performance is the percentage of pixels that have been miss-classified.

Several papers evaluate the performance of MRF algorithms in the segmentation of binary [Elliott 1984], or grayscale images [Dubes et al. 1990; Kolmogorov and Rother 2006; Szeliski et al. 2008]. Most papers assume known MRF parameters and process pictures at most 800×600 pixels in size, which are much smaller compared to the datasets obtained by high-throughput imaging. [Dubes et al. 1990] compared the performance of earlier methods (SA, ICM) in the segmentation of 64×64 binary images corrupted with either gaussian noise, or correlated noise or texture. ICM was the method that provided the most robust performance in segmenting either noisy (SNR=1) simulated images (unimodal histogram), or real images (bimodal histogram). Several papers compared the performance of newer and more efficient algorithms (GC, BP and TRWP). GC, BP and TRWP converge much faster and much closer to the global minimum compared to ICM [Szeliski et al. 2008]. [Kolmogorov and Rother 2006] found out that graph cuts (GC) outperform BP and TRW in highly-connected graphs, however such difference was not observed in low-connectivity (4-connected) graphs [Szeliski et al. 2008] where methods performed similarly.

Applications of MRF in Segmentation of Spectral Images

Image segmentation is one of the major applications of MRF models [Li 2009]. A big fraction of the papers that discuss the development of an MRF algorithm utilize a benchmark that is based segmenting a set of reference pictures [Szeliski et al. 2008]. A large number of papers propose

MRF implementations for the segmentation of grayscale images based on different assumptions of image properties [Elliott et al. 1984; Dubes et al. 1990; Marroquin et al. 2003]. The number of papers that discuss the practical application of MRF in real-world imaging applications is far less. Among them, there are several papers and books that discuss the application of MRF in segmenting hyper-spectral images in remote sensing (RS) [Jackson and Landgrebe 2002; Varshney and Arora 2004; Richards and Jia 2006]. Kerfoot and Bresler 1999] provides an interesting analysis of segmentation error for a particular case of segmentation of RS images. Brute-force application of MRF to unmix and segment hyperspectral RS images has been applied but it is extremely computational intense [Eches et al. 2010]. [Bai et al. 2013] provide a practical application of MRF for hyper-spectral RS data, which follows the approach followed in this study: an MRF field solved by graph cuts is used to refine an initial segmentation by a local segmentation algorithm (support vector machines classification). This approach is proposed to be more robust and require training samples of smaller size compared to a brute-force application of MRF.

Applications of MRF in 3D fluorescence imaging

In contrast to the remote sensing field, at the moment, the practical application of MRF in the segmentation of 3D spectral fluorescence images is at its infancy. MRF have been applied for image reconstruction in single-channel 3D confocal and multiphoton fluorescence images [Mondal et al. 2007, 2008]. The image reconstruction problem does not attempt to label the image pixels, but instead attempts to correct image degradation induced by the finite size of optics PSF, and detector noise (sensor noise, Poisson noise). This study therefore provides one of the first practical applications of MRF for the segmentation of large volume high-throughput high-content 3D fluorescence data.

2.3 Instrumentation

2.3.1 Instrument description

All imaging data in this thesis have been acquired using a custom-made multi-photon microscope located in the So lab (Department of Biological Engineering, MIT). The instrument was originally designed and built by Dr. K.H. Kim and Dr. Christof Buehler around 2003 [Kim 2005; Buehler et al. 2005]. The major contribution of the original design was the development of a 16-channel single-photon counting card, which enabled high-sensitivity spectral imaging based on a MA-PMT. During 2010 several optical (lenses, filters, dichroic mirror, MA-PMT) and mechanical components (tube lens-scan lens housing) of the instrument were modified in order to improve its performance and ease of use. A schematic of the post-2010 instrument (used to acquire the data shown in this thesis) is shown in Figure 2.3.1. The microscope is excited by a Tsunami 3960 mode-locked titanium-sapphire laser (Spectra Physics, Santa Clara, CA) that provides femtosecond laser pulses of 2W power (at 775 nm) of tunable wavelength in the [720, 900] nm range. The laser beam power provided to the sample is precisely manually controlled by a half-wave plate (H1) and analyzer (A1) system. The beam is deflected by a pair of computer-controlled galvo mirrors (SM) and further expanded 5× (by the theta scan lens (SL) and tube lens (TL) in order to overfill the back focal plane of the objective lens (OL). The C-apochromat 40× 1.2NA, 0.28mm WD, water immersion objective lens (441757-9970-000, Carl Zeiss Microscopy, Thornwood, NY) focuses the laser beam inside the sample and collects a

fraction of the emission in epi-fluorescence configuration. Imaging a plane is implemented by raster-scanning the position of the laser focus using the scanning mirrors. 3D imaging is implemented by using a piezoelectric actuator (PI) to translocate the objective and image multiple planes inside the sample.

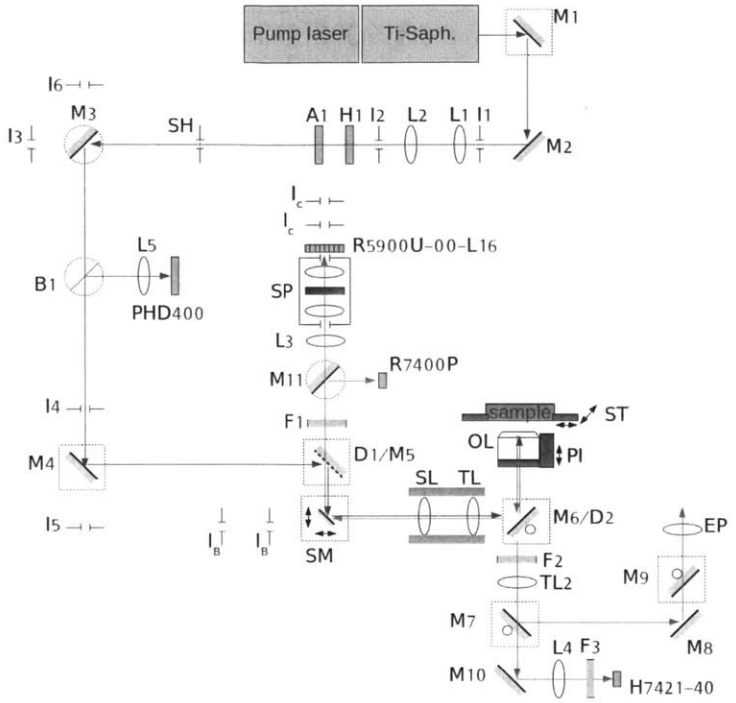


Figure 2.3.1: Schematic of the custom-made multi-photon microscope utilized for spectral multi-photon imaging.

The particular instrument contains three imaging sensors, where each one can be used for specific applications. All spectral imaging data shown in this thesis were acquired using a 16-channel multi-anode photomultiplier tube (MA-PMT) (R5900P-00-L16, Hamamatsu, Bridgewater, NJ). Spectral imaging data are acquired in a de-scanned configuration: the emission is separated from the excitation laser by a dichroic filter (T700DCSPXRUV, Chroma Technology Corporation, Bellows Falls, VT), filtered to remove residual infrared emission by a bandpass filter (ET680SP-2p8, Chroma Technology Corporation), resolved in a diffraction grating spectrograph (MS125, Oriel instruments, Stratford, CT), and detected in the MA-PMT operated in single-photon counting mode [Hakamata et. al. 2006]. The 16-channels of the MA-PMT are conditioned and detected in a custom-made 16-channel photon counting card [Buehler et al. 2005], and the resulting photon count signal per pixel is transmitted to the control PC.

The instrument is also equipped with a high-efficiency photon counting head (H7421, Hamamatsu) used for intensity imaging and fluorescence correlation spectroscopy (FCS) in scanned configuration, and a fast PMT (R7400P, Hamamatsu) used for intensity imaging and for fluorescence lifetime imaging (FLIM) in de-scanned configuration

2.3.2 Spatial Resolution and Pixel Size

The spatial characteristics of the MPM microscope described in Section 2.3.1 are described by its spatial resolution and its pixel size.

The spatial resolution of a MPM describes the ability to focus the excitation laser within a volume as small as possible (and therefore cause emission by molecules located within a

volume as small as possible). The spatial resolution depends on the excitation light wavelength and the excitation optics (particularly the numerical aperture of the objective lens, the degree of overfilling of the objective back focal aperture by the excitation beam, the width of the femtosecond pulses and any aberrations present in the optics of the excitation light path [Born and Wolf 1999; Gu 2000]). The optics of the emission path do not have a major impact on the instrument PSF, however they affect the amount of emitted light that is collected and transmitted from the sample to the MA-PMT.

The spatial resolution of a MPM is described by its excitation point spread function (PSF), which can be approximated using the equation that describes the square of intensity of a Gaussian beam [Saleh and Teich 2007].

$$I^2(r,z) = I_0^2 \cdot (1 + \left(\frac{z}{z_R}\right)^2)^{-2} \cdot \exp(-\frac{4 \cdot r^2}{w^2(z)}) \qquad [2.3.1]$$

where r and z are the radial and axial position from the light focus respectively, $z_R = \pi w_0^2/\lambda$ is the Rayleigh range, and $w(z) = w_0\sqrt{1 + \left(\frac{z}{z_R}\right)^2}$ is the waist radius. The single parameter of the Gaussian beam is the waist radius w_0. The value of w_0 for the Gaussian beam that approximates the PSF of a MPM is estimated using z-stack of images of 0.1 μm fluorescent beads immobilized inside a 2% agarose marix, acquired using low excitation intensity to avoid saturation of excited molecules [Juskaitis 2006].

The pixel size δ of each image acquired by the microscope depends on the excitation optics, is analogous to the objective magnification, is analogous to the range of motion of the scanning mirrors, and depends on the pixel number per image row/column. Given microscope configuration, the pixel size can be calculated by imaging a Ronchi ruler of known spacing and identifying the peak of the first harmonic in its 2D FFT spectrum [Oppenheim et al. 1996]. The field of view of each image equals δ times the number of pixels.

For the microscope shown in Section 2.3.2 the estimated beam waist radius is approximated as $w_0 = 0.405$ μm (equivalently $z_R = 0.667$ μm), and the pixel size is 0.432 μm (for a 256×256 image this corresponds to a 110.6×110.6 μm field of view).

2.3.3 Generation of the Wavelength-Resolved Spectral Signal

Figure 2.3.2a shows the propagation of signal in a spectral MPM operating in single-photon counting mode such as the instrument described in Section 2.3.1. A photon emitted in the sample is collected by the microscope objective lens, propagates through the microscope intermediate optics, its trajectory is spectrally-resolved in the spectrograph, and finally arrives in the PMT. In the cathode of the 16-channel MA-PMT the photon is converted into electrons. The resulting electrical signal is amplified in the PMT dynodes and reaches one of the 16 dynodes of the MA-PMT, generating an electric pulse of mV amplitude and O(nsec) duration [Hakamata et. al. 2006]. A pulse originating from the j-th PMT channel propagates through a dedicated coaxial 50Ω cable and reaches the j-th discriminator of the 16-channel card that converts pulses of significant amplitude into a TTL pulse that is counted in a counter [Buehler et al. 2005]. The resulting photon counts for the 16 channels of each pixel are transmitted to the control PC, which acquires data for all pixels and forms the image.

The key component of a spectral microscope compared to a normal microscope, is the optical system that resolves the detected emission based on its wavelength. In an ordinary MPM each image pixel provides a scalar measurement y, the detected emission intensity [Hakamata et. al. 2006]. In a spectral MPM, each image pixel provides a vector measurement $\boldsymbol{y}$, where each $\boldsymbol{y}$ element detects the emission within a particular region of the EM spectrum. In the instrument shown in Figure 2.3.1, emission is resolved by a reflectance diffraction grating (77414, Newport Optics, Irvine, CA) inside a spectrograph (MS125, Oriel instruments, Stratford, CT). The operating principle of the spectrograph is depicted in Figure 2.3.2b. An incident light beam of wavelength λ is reflected by the grating into multiple orders [Palmer and Loewen 2005; James 2007]. The reflection angle θ_m of the m-th order depends on the incident angle φ and the grating frequency G:

$$\sin(\theta_m) = mG\lambda - \sin(\varphi)$$

In a spectrograph, the mode of interest is the first mode (m=1). Since θ_1 depends on λ, photons of different wavelength arrive at different positions $x(\lambda)$ of the MA-PMT, and are detected by a different detector channel. The range of the EM spectrum $\Delta\Lambda$ that is simultaneously detected by the MA-PMT corresponds to wavelengths λ such that $x(\lambda)$ lie within the MA-PMT aperture. $\Delta\Lambda$ depends on the sensor area width W and the spectrograph optics focal length f. Similarly, the range of wavelength $\delta\lambda$ (channel detection range) that is detected by each channel depends on light dispersion by the grating and the width of the channel w.

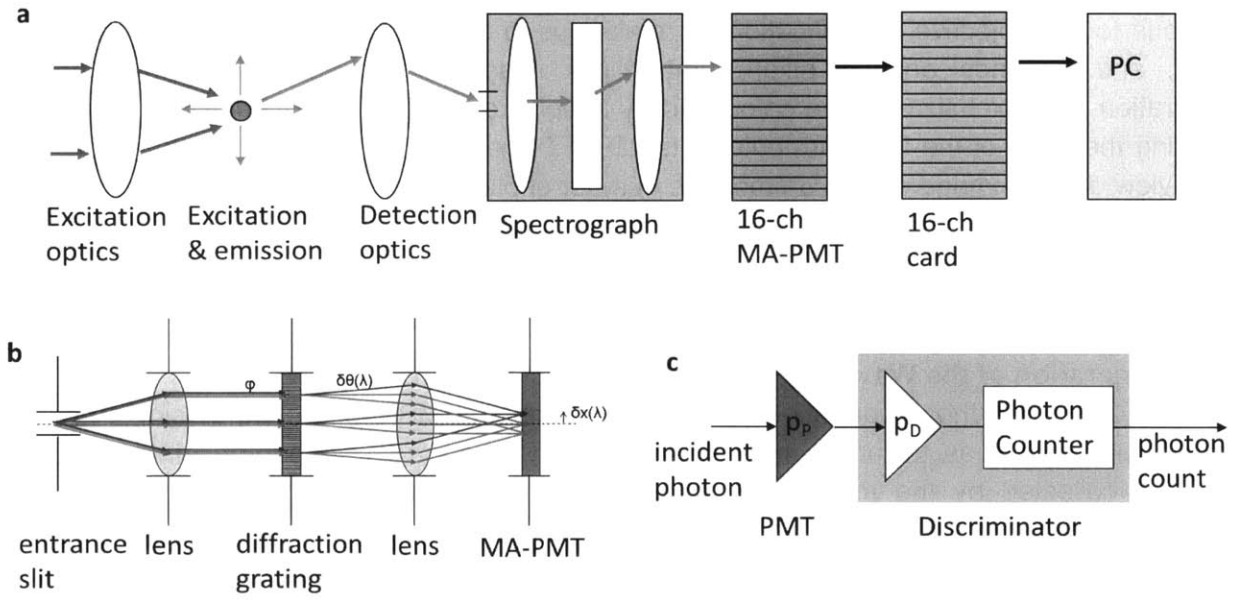


Figure 2.3.2: Signal propagation in a spectral multi-photon microscope. **a)** Propagation of signal through the optical and electronic parts. **b)** Schematic of the spectrograph-MAPMT system used to resolve and detect emission photons based on their wavelength, **c)** Schematic of the effective amplification in the photon detection and counting process.

The emission optics of the instrument presented in Figure 2.3.1 are designed to transmit light in the region [380, 620] nm. In Figure 2.3.2b the nominal design wavelength $\lambda_0 = 500$ nm is highlighted green, while the minimum and maximum design wavelengths ($\lambda_{min} = 380$nm, $\lambda_{max} = 620$ nm) are highlighted blue and red respectively. Spectrograph properties are $\lambda_B =$ 400 nm (brazing wavelength), $G = 600$ lines/mm, f =120 mm, and maximum efficiency η_{max} =

85%. Combined with the W=16mm width of the MA-PMT (R5900P-00-L16), geometric optics suggest that the MA-PMT can simultaneously detected light within a $\Delta\Lambda = 210$ nm region (Appendix C). The channel detection range $\delta\lambda_j$ of the j-th sensor channel varies; channels that detect blue illumination detect a slightly wider region of the spectrum: $\delta\lambda_1 = 13.2$ nm (blue-most channel), $\delta\lambda_8 = 13$ nm (intermediate wavelength channel), $\delta\lambda_{16}$=12.6 nm (red-most channel). Analytical estimations agree well with spectral calibration results based on fluorescent dye standard images. Spectral calibration is the process of estimating the wavelength region [$\lambda_{min,j}$, $\lambda_{max,j}$] (j=1, 2, …, M) that is detected by each channel of the MA-PMT (Appendix C).

2.3.4 Spectral Response

The spectral response function $h_s(j)$ of a spectral microscope is described by the probability that an emitted photon of wavelength λ that reaches the MA-PMT is detected at its i-th channel. In this paragraph $j = 0$ denotes the sensor channel where the emission should be detected based on geometric optics. Ideally, all photons of the same wavelength λ should be detected in this channel $h_s(j) = \delta(j)$. In practice, due the finite resolution of the emission path optics, optical alignment errors, and the probabilistic nature of electron amplification and propagation in PMT dynodes [Hakamata et. al. 2006], emission photons of the same wavelength can be detected in several channels centered around j=0, i.e. $h_s(j) \neq 0$ for more than one channels. The spectral response function $h_s(j)$ of a spectral microscope broadens the detected spectrum $\boldsymbol{s}_i$ of each source, which deteriorates the precision to resolve the emission of multiple sources (Section 2.4.5).

2.3.5 Channel Gain

The response of different channels of a MA-PMT is not identical due to two major reasons: i) incident photons are amplified differently by different MA-PMT channels, and ii) PMT output pulses are amplified differently by different discriminator card channels. These two steps are described in Figure 2.3.2c by two different probabilities: $p_{p,j}$ is the probability that an photon arriving in the j-th PMT channel will generate a voltage pulse, and $p_{D,j}$ is the probability that a voltage pulse generated by the j-th PMT channel will pass through the discriminator and generate a photon count [Hakamata et. al. 2006]. The product $p_{C,j} = p_{p,j} \cdot p_{D,j}$ is the probability that a photon arriving at the j-th MA-PMT channel will produce a photon count by the corresponding photon counting channel circuitry. $p_{C,j}$ is also the total gain for channel j. The differential gain of a MA-PMT is described by a diagonal gain matrix $\boldsymbol{\Lambda}_G = diag(p_{C,j})$, which is estimated during instrument characterization (Appendix B).

2.3.6 Spectral Signatures

Each emission source has a unique emission spectrum $p_{em}(\lambda)$, which can be described as the PDF that a photon emitted by the source will have energy equal to $E = h \cdot c/\lambda$ (h is Plank's constant and c is the speed of light). The emission spectrum $p_{em}(\lambda)$ of pure fluorophores is a molecular property, which does not depend on the excitation wavelength (although in some cases it depends on its environment, e.g. for NADH) [Lakowicz 2006]. The emission spectrum of molecules that contain multiple fluorescent entities (e.g. collagen) varies with excitation wavelength. The emission spectra of pure chemical components after two-photon excitation are theoretically identical to the emission spectra after single-photon excitation [Lakowicz 2006].

This has been experimentally verified for several fluorophores by [Xu and Webb 1996; Bestvater et al. 2002]. However, there are cases of fluorophores whose emission spectrum after two-photon excitation is significantly different (small differences are attributed to calibration errors) compared to single-photon emission [Bestvater et al. 2002]. The emission spectrum of second harmonic scattering is concentrated at half the excitation wavelength $\lambda_{em}/2$ [Campagnola and Loew 2003].

While the emission spectrum $p_{em}^{i}(\lambda)$ of source i is a continuous function of wavelength λ characteristic of the source, the spectral signature $\boldsymbol{s}_i$ detected by a M-channel spectral microscope is a M-element vector. $\boldsymbol{s}_i$ can be thought of as a probability mass function whose j-th element describes the probability that a photon emitted by source i and detected by the PMT will be detected at j-th channel (j=1, 2,..., M). The spectral signature $\boldsymbol{s}_i$ depends on the emission spectrum $p_{em}^{i}(\lambda)$, the emission path optics, the spectral response of the microscope and the differential gains in the MA-PMT & photon counting card channels.

2.3.7 Estimation of Spectral Signatures

Accurate spectral unmixing requires accurate knowledge of the spectral signature $\boldsymbol{s}_i$ for each source in the image, see Sections 2.4.5 and 2.4.7. $\boldsymbol{s}_i$ can be either measured experimentally (by imaging solutions of pure sources) or can be estimated based on the known emission spectrum $p_{em}^{i}(\lambda)$ of the source, the measured spectral response of the instrument $h_s(j)$, the optical efficiency of the emission light path $\eta_{opt}(\lambda)$, and the geometry of the detection system. Accurate computation of $\boldsymbol{s}_i$ from $p_{em}^{i}(\lambda)$ is hard due to the complex nature of photon conversion and amplification in a MA-PMT. In this thesis $\boldsymbol{s}_i$ is approximated in three steps:

1) Calculate the relative emission that arrives at the j-th MA-PMT channel as:

$$r_i(j) = \int_{\lambda_{min,j}}^{\lambda_{max,j}} p_{em}^{i}(\lambda) \cdot \eta_{opt}(\lambda) \cdot d\lambda \quad j = -k, \dots, 1, 2, \dots, M, \dots, k+1$$

where $[\lambda_{min,j}\ ,\ \lambda_{max,j}]$ is the wavelength range detected by channel j based on geometric optics. $\eta_{opt}(\lambda)$ describes the transmissivity of detection optics and the quantum efficiency of the MA-PMT. This calculation is done for the M channels of the MA-PMT (j=1, 2, ..., M) as well as for k dummy channels that correspond to locations next to the PMT sensor aperture.

2) The emission $r_i(j)$ that arrives in the PMT is convolved with the spectral response of the microscope $h_s(j)$ and amplified by the total gain $p_{C,j}$ of channel j.

$$\hat{s}_{ji} = p_{C,j} \cdot (r_i(j) * h_s(j))$$

where $*$ is the convolution operator.

3) Finally, the spectral signature $\boldsymbol{s}_i$ is calculated by normalizing the outcome so that the sum of its elements equals 1:

$$\boldsymbol{s}_i = [s_{1i} \quad \dots \quad s_{Mi}]^T$$

$$s_{ji} = {}^{\hat{s}_{ji}}\Big/_{\sum_{j=1}^{M} \hat{s}_{ji}}$$

The spectral signature $\boldsymbol{s}_i$ is instrument-dependent and depends on the properties of the optical components (dichroic filters, MA-PMTs, diffraction gratings) and the spectral calibration of the instrument [Tsurui et al. 2000; Berg 2004]. Also, when imaging deep inside scattering samples,

spectral signature $\boldsymbol{s}_i$ is expected to progressively broaden due to the effect of the scattering, which deviating photons from their ballistic trajectories and therefore randomizes the trajectories of photons collected by the objective lens [Helmchen and Denk 2005].

2.4 Spectral Unmixing

2.4.1 Definition and Nomenclature

Spectral unmixing is the mathematical process that estimates the contribution of each source in the emission detected by a M-channel detector, based on the distinct emission spectra $\boldsymbol{s}_i$ of the sources [Keshava 2003].

The measured signal $\boldsymbol{y}$ in every pixel of a spectral image is a M×1 vector, whose j-th element contains the emission detected in the j-th channel of the detector. The detected light was originally emitted by N sources in the sample. The light emitted by each source propagates through the microscope emission optics and reaches the sensor (Figure 2.4.1). It is assumed that a single measurement of the vector $\boldsymbol{y}$ is acquired per image (the usual practice). Let λ_i denote the emission rate of light arriving at the detector originally emitted by the i-th source (i=1, 2, 3..., N) that arrive in all M detector channels. The N×1 detected source emission rate vector

$$\boldsymbol{\lambda} = [\lambda_1 \quad \dots \quad \lambda_N]^T$$

contains the emission rates of all sources that eventually arrive in the detector.

The spectral signature of each source is described by the M×1 vector:

$$\boldsymbol{s}_i = [s_{1i} \quad \dots \quad s_{Mi}]^T$$

where s_{ji} refers to the probability that a detected photon will be detected in channel j provided that the photon is emitted by source i (Section 2.3.6). By definition $s_{ji} \geq 0$ and $\sum_{j=1}^{M} s_{ji} = 1$ (the elements are non-negative and their sum equals 1). The spectral properties of a particular set of sources are described by the M×N spectral matrix:

$$\boldsymbol{S} = [\boldsymbol{s}_1 \quad \dots \quad \boldsymbol{s}_N] = [\boldsymbol{t}_1 \quad \dots \quad \boldsymbol{t}_N]^T$$

The i-th column contains the spectral signature $\boldsymbol{s}_i$ of the i-th source. The j-th row $\boldsymbol{t}_j$ provides the contribution the source emissions $\boldsymbol{\lambda}$ to the emission detected at channel j.

Mixing models, observation models, and estimators

The objective of spectral unmixing is to estimate the detected emission rates $\boldsymbol{\lambda}$ of the sources given a measurement $\boldsymbol{y}$ and the spectral matrix $\boldsymbol{S}$. Spectral unmixing utilizes a mixing model that describes how the fluorescent emission of the N sources generates the photon counts detected in the M-channel detector. The detected signal depends on the properties of the fluorescent emission of the sources, the properties of the intermediate optics, and the properties of the sensor, all of which are incorporates in the $\boldsymbol{\lambda}$ and $\boldsymbol{S}$ (Section 2.3.3). A mixing model is described mathematically by the observation model $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})$ (or equivalently the log-likelihood $\mathrm{L}(\boldsymbol{\lambda}|\boldsymbol{y}) \equiv \log(p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}))$), which is the PDF for detecting a measurement $\boldsymbol{y}$ in the M-channel detector when the detected emission rates of the N sources are $\boldsymbol{\lambda}$.

Bayes law provides the a-posteriori PDF for $\boldsymbol{\lambda}$ given a measurement $\boldsymbol{y}$:

$$p_{\boldsymbol{\Lambda}|\boldsymbol{Y}}(\boldsymbol{\lambda}|\boldsymbol{y}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})}{p_{\boldsymbol{Y}}} p_{\boldsymbol{\Lambda}}(\boldsymbol{\lambda})$$

Bayesian law fuses the information incorporated in the measured data vector $\boldsymbol{y}$ with the a-priori knowledge about $\boldsymbol{\lambda}$ described by the a-priori PDF $p_{\Lambda}(\boldsymbol{\lambda})$. The posterior $p_{\Lambda|Y}(\boldsymbol{\lambda}|\boldsymbol{y})$ can be used to infer $\boldsymbol{\lambda}$.[MacKay 2003; Sivia and Skilling 2006]. One possible point estimator $\hat{\boldsymbol{\lambda}}$ of $\boldsymbol{\lambda}$ is the maximum a-posteriori estimator (MAP) $\hat{\boldsymbol{\lambda}}_{MAP}$, which is the value of $\boldsymbol{\lambda}$ that maximizes $p_{\Lambda|Y}(\boldsymbol{\lambda}|\boldsymbol{y})$:

$$\hat{\boldsymbol{\lambda}}_{MAP} = \mathrm{argmax}_{\boldsymbol{\lambda}} \; p_{\Lambda|Y}(\boldsymbol{\lambda}|\boldsymbol{y})$$

When there is no a-priori information about $\boldsymbol{\lambda}$ (a usual assumption done in image processing), $p_{\Lambda}(\boldsymbol{\lambda})$ is chosen as a non-informative constant, and $\hat{\boldsymbol{\lambda}}_{MAP}$ is identical to the maximum likelihood (ML) estimator $\hat{\boldsymbol{\lambda}}_{ML}$, which is the value of $\boldsymbol{\lambda}$ that maximizes the likelihood $p_{Y|\Lambda}(\boldsymbol{y}|\boldsymbol{\lambda})$ or equivalently the log-likelihood $\mathrm{L}(\boldsymbol{\lambda}|\boldsymbol{y}) = \log(p_{Y|\Lambda}(\boldsymbol{y}|\boldsymbol{\lambda}))$:

$$\hat{\boldsymbol{\lambda}}_{ML} = \mathrm{argmax}_{\boldsymbol{\lambda}} \; \mathrm{L}(\boldsymbol{\lambda}|\boldsymbol{y})$$

Two mixing models (Poisson model, Gaussian model) are described in Sections 2.4.3 and 2.4.4. The performance of the models is discussed in Sections 2.4.5.

Sources – Classes – Compartments

A source is defined as a chemical entity that emits light, which is detected by the spectral microscope. Sources considered in multi-photon microscopes include fluorophores as well as crystalline structures of biopolymers that emit second harmonic scattering.

A class is defined as a physical entity that can contain a particular set of sources. For example, in the fluorescent imaging of cells inside collagen scaffolds (Chapter 3), the class "nucleus" contains the sources "Hoechst33342" (a fluorescent nucleic acid stain) and CMTMR (a fluorescent cell tracker), the class "cytoplasm" contains the sources CMTMR and Alexa Fluor 488 (the fluorescence dye conjugates to the secondary antibody that binds to a particular protein located in cell cytoplasm), the class "matrix" contains the autofluorescence and the second harmonic emission of the collagen located in the scaffold's struts, and the class "void" does not contain any source (any signal is attributed to detector noise) and corresponds to the medium volume between cells and the scaffold.

A compartment is defined as a region that can contain a particular set of classes. For example, in the fluorescent imaging of *ex vivo* transected rat peripheral nerve treated with collagen scaffolds (Chapter 4), the compartment "nerve tissue" can contain the classes "myofibroblast cytoplasm", "nucleus", "Schwann cell cytoplasm", "axon", "blood vessel", "stroma" and "void".

In this thesis, it is assumed that

- The number of compartments, classes and sources per class are known
- The set of classes for each compartment, and the set of sources for each class are known
- There is some limited a-priori knowledge about the range of the detected emission rate λ_i for each source of each class, expressed as a uniform PDF between a lower and an upper bound.

Result interpretation

The interpretation of the emission rate vector $\boldsymbol{\lambda}$ depends on the application. Usually the entity of interest is the density ρ_i (i=1, 2, 3..., N) of the sources in the sample. In the absence of energy transfer mechanisms (e.g. quenching, FRET), photobleaching, and depletion due to high excitation light intensity [Lakowitz 2006], ρ_i is approximately analogous to the rate of light λ_i^0

emitted originally by the i-th source [Xu and Webb 1996]. It should be noted that λ_i is less than λ_i^0 because only a fraction of the light emitted by each source is collected by the microscope objective, propagates through the detection optics and reaches the detector.

2.4.2 The Multiple-Source Multiple-Detector Inverse Problem

Spectral unmixing is a special case of a more general "multiple source multiple-detector" inverse problem, found in many signal processing problems. The objective of this inverse problem is to estimate the signal emitted by N sources based on measurements from M detectors, Figure 2.4.1. The signal at each detector is the outcome a mixing process that linearly combines the signals emitted by the sources, described by mixing parameters p_{ij} . When the emission signal is continuous, λ_i is the emission magnitude of the i-th source, and p_{ij} correspond to the fraction of λ_i that reaches detector j. When the emission signal is discrete, λ_i is the Poisson rate for the emission of the i-th source, and p_{ij} is probability that an emission event from source i will be detected by detector j. In both cases $\sum_{j=1}^{M} p_{ij} = 1$, $\forall\, i = 1,2, \ldots N$.

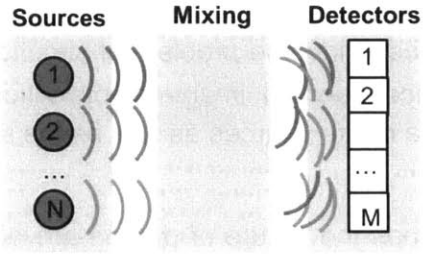


Figure 2.4.1: Schematic of the multiple source – multiple detector problem. The emission of N sources is mixed and detected by an array of M detectors. The objective of the multiple source – multiple detector problem is to estimate the emission rates of the N sources based on the measured signal in the M detectors.

This chapter addresses the problem of spectral unmixing in spectral microscopy data. For every pixel in the spectral image, the objective is to estimate the emission rates λ_i of the N emission sources based on the detected signal $\boldsymbol{y}$ in the M channels of the sensor. Usually N is known (in the order of 2-5), M is in the order of 4 to 32, and the emission spectra of the sources (correspond to the mixing parameters p_{ij}) are known quite well. As long as sources have distinct emission spectra the problem is well-posed. The accuracy of the unique solution depends on the accuracy of our knowledge for p_{ij} and on the noise that corrupts the detected signal. The inverse problem needs to be solved for each pixel in the image. The appropriate unmixing implementation must consider the large number of pixels found in modern 3D images of biological samples (10^6 - 10^7), and that for each pixel there is usually a single measurement.

Various forms of the multiple-source multiple-detector inverse problem appear in different engineering problems. These forms differ in the magnitudes of N and M, whether the problem is mathematically well-posed (well-posed problems have unique solutions), on the accuracy of the knowledge of the parameters p_{ij} , on whether p_{ij} are known or also need to be estimated, on the fraction of the M detectors that are measured, and on the nature of the noise present. Examples include various kinds of tomographic imaging (Position Emission Tomography (PET), single-photon emission computed tomography (SPECT)), and image reconstruction (e.g. image deconvolution, photon re-assignment). Even though the conceptual solution to the multiple-source multiple-detector inverse problem is common, the unique features of each case (Table

2.4-1) affect critically the numerical implementation (EM algorithms in PET imaging [Dempster et al. 1977; Shepp and Vardi 1982], Richardson-Lucy algorithm for image deconvolution [Richardson 1972; Lucy 1974]).

Application	Sources i	Detectors j	p_{ij}	Runs
Spectral unmixing	Fluorophores 2<N<8 Usually N known	PMT channels 2<M<32 M known	Emission spectra Relatively well known	10^5 - 10^7
PET, SPECT	Emission at body position i (10^5 <N< 10^7) N depends on grid	Signal at channel pairs j (10^2<M<10^3) M known	Geometry factor Estimated, maybe poor estimates	1
Image restoration	Real Intensity at pixel I (10^3 <N< 10^6)	Detected Intensity at pixel I (10^3 <M< 10^6)	PSF, may be position-dependent	1

Table 2.4-1: Properties of different multiple-source multiple-detector problems.

In some cases the mixing process is unknown (p_{ij} are unknown). Blind signal separation (BSS) methods attempt to solve the ill-posed inverse problem of simultaneously estimating p_{ij} and λ_i, see Section 2.2.3. In fluorescence spectral imaging applications, BSS methods attempt to estimate both the emission spectra of the sources as well as the source signal contributions.

2.4.3 Poison Mixing Model

The Poisson mixing model describes the nature of photon arrivals at each detector channel as the origin of stochastic nature of the detected signal $\boldsymbol{y}$. There is no analytic solution for the ML estimator $\hat{\boldsymbol{\lambda}}_{ML}$, which instead is calculated numerically by iterative optimization (EM algorithm).

Poison observation model

Due to the nature of photon arrival at each detector channel, the photon count at each channel follows Poisson distribution. Photon arrivals at channel j equals the sum of the contributions of photons emitted by each source. Since each one of these contributions is a Poisson process of rate $\lambda_i p_{ij}$, the photon count at channel j also follows a Poisson distribution $y_j \sim P(y_j; \boldsymbol{t}_j^T \boldsymbol{\lambda})$ with rate $d_j = \boldsymbol{t}_j^T \boldsymbol{\lambda}$.. The rate of photon arrivals at each channel equals:

$$\boldsymbol{d} = \boldsymbol{S} \cdot \boldsymbol{\lambda} \qquad [2.4\text{-}1]$$

so that d_j is the j-th element of $\boldsymbol{d}$. Since the photon count of each channel is an independent random variable, the probability of observing the measurement $\boldsymbol{y}$ (observation model) is:

$$p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}) = \prod_{j=1}^{\boldsymbol{M}} P(y_j; \boldsymbol{t}_j^T \boldsymbol{\lambda}) = \prod_{j=1}^{\boldsymbol{M}} \frac{e^{-d_j} \cdot d_j^{y_j}}{y_j!} = e^{-\|\boldsymbol{\lambda}\|_1} \cdot \prod_{j=1}^{\boldsymbol{M}} \frac{d_j^{y_j}}{y_j!} \qquad [2.4\text{-}2]$$

The log-likelihood of the emission model is:

$$L(\boldsymbol{\lambda}|\boldsymbol{y}) = log\left(p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})\right) = -\|\boldsymbol{\lambda}\|_1 + \sum_{j=1}^{M} \{y_j \cdot \log(d_j) - \log(y_j!)\} \qquad [2.4\text{-}3]$$

Maximum likelihood estimation

The gradient and Hessian matrix $\boldsymbol{H}(\boldsymbol{\lambda})$ of the log-likelihood function $L(\boldsymbol{\lambda}|\boldsymbol{y})$ with respect to $\boldsymbol{\lambda}$ are:

$$\frac{\partial L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}} = \boldsymbol{S}^T \cdot diag(d_j^{-1}) \cdot \boldsymbol{y}$$

$$\boldsymbol{H}(\boldsymbol{\lambda}) = \frac{\partial^2 L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial\boldsymbol{\lambda}\partial\boldsymbol{\lambda}^T} = -\boldsymbol{S}^T \cdot diag(\frac{y_j}{d_j^2}) \cdot \boldsymbol{S}$$

where $d_j = \boldsymbol{t}_j^T \cdot \boldsymbol{\lambda}$. The Hessian has been shown to be negatively defined ($\boldsymbol{c}^T \cdot \boldsymbol{H} \cdot \boldsymbol{c} < 0$ $\forall$ N×1 vector $\boldsymbol{c}$) [Shepp and Vardi 1982]. It follows that $L(\boldsymbol{\lambda}|\boldsymbol{y})$ is a concave function that possesses a global maximum. The convexity of $L(\boldsymbol{\lambda}|\boldsymbol{y})$ is critical for efficient computational spectral unmixing (Section 2.4) and pixel-wide image segmentation (Section 2.5.2) [Boyd and Vandenberghe 2004; Nocedal and Wright 2006]. A choice of the prior $p_{\Lambda}(\boldsymbol{\lambda})$ other than the non-informative constant prior could result in a non-concave posterior $p_{\Lambda|\boldsymbol{Y}}(\boldsymbol{\lambda}|\boldsymbol{y})$. In this case the computation of the MAP estimator $\hat{\boldsymbol{\lambda}}_{MAP}$ wound be much more computationally expensive

The ML estimator $\hat{\boldsymbol{\lambda}}_{ML}(\boldsymbol{y})$ is the non-negative vector $\boldsymbol{\lambda}$ that maximizes $L(\boldsymbol{\lambda}|\boldsymbol{y})$. Based on the Karush–Kuhn–Tucker (KTT) conditions, when all elements of $\hat{\boldsymbol{\lambda}}_{ML}$ are larger than zero (the peak of $L(\boldsymbol{\lambda}|\boldsymbol{y})$ lies within the region $\lambda_i > 0$), then the gradient of $L(\boldsymbol{\lambda}|\boldsymbol{y})$ at $\hat{\boldsymbol{\lambda}}_{ML}$ equals zero [Boyd and Vandenberghe 2004], or equivalently:

$$\boldsymbol{S}^T \cdot diag\left(\boldsymbol{t}_j^T \cdot \boldsymbol{\lambda}_{ML}\right) \cdot \boldsymbol{y} = \boldsymbol{1}_N$$

where $\boldsymbol{1}_N$ is a N×1 vector that contains unity elements. Because there is no analytical solution to this system of nonlinear equations and its solution is not guaranteed to be non-negative, the most efficient way to calculate $\hat{\boldsymbol{\lambda}}_{ML}$ is by numerically maximization of $L(\boldsymbol{\lambda}|\boldsymbol{y})$ [Nocedal and Wright 2006; Press et al. 2007].

Numerical Implementation

Calculating $\hat{\boldsymbol{\lambda}}_{ML}$ requires finding the value of $\boldsymbol{\lambda}$ that maximizes $L(\boldsymbol{\lambda}|\boldsymbol{y})$ numerically. A valid numerical solution should satisfy the non-negativity constraints $\hat{\lambda}_i \geq 0$ (valid emission rates take non-negative values). Such constraints can be explicitly applied using a general-purpose constrained nonlinear optimization algorithm [Press et al. 2007], however this is not advisable due to the complex nature of constrained optimization algorithms and the need to run a different optimization run for each pixel. An alternative is to calculate $\hat{\boldsymbol{\lambda}}_{ML}$ by applying the expectation-maximization (EM) algorithm [Dempster et al. 1977], developed for numerical optimization of likelihood functions. The EM algorithm applied here was developed for calculating the ML estimate of a related multiple-source multiple-detector inverse problem in PET imaging [Shepp and Vardi 1982]. The algorithm has the desirable feature that it provides non-negative results using a modified unconstrained optimization algorithm. Since $L(\boldsymbol{\lambda}|\boldsymbol{y})$ is concave (see previous paragraph) the algorithm is guaranteed to find the global maximum of $L(\boldsymbol{\lambda}|\boldsymbol{y})$.

The application of the EM method in finding the ML estimate in the Poisson mixing model is described in Appendix D, and is based on [Candy 2009]. The EM algorithm is an iterative algorithm that updates the current estimate of the ML solution until the solution converges. At the k+1 iteration, the new estimate of the emission rate of the i-th source is updated based on the previous estimate $\hat{\boldsymbol{\lambda}}^{(k)}$ as follows:

$$\hat{\lambda}_i^{(k+1)} = \hat{\lambda}_i^{(k)} \cdot \sum_{j=1}^{M}\{s_{ji}\frac{y_j}{\hat{d}_j^{(k)}}\} \qquad [2.4\text{-}4]$$

$$\hat{d}_j^{(k)} = \sum_{l=1}^{N} s_{jl} \cdot \hat{\lambda}_l^{(k)} = \boldsymbol{t}_j^T \cdot \hat{\boldsymbol{\lambda}}^{(k)} \qquad [2.4\text{-}5]$$

This implementation of the EM algorithm assures that the results will satisfy non-negativity (if $\hat{\lambda}_i^{(0)} > 0$ for the initial guess, then $\hat{\lambda}_i^{(k)} \geq 0$, $\forall k > 0$) and additivity ($\sum_{i=1}^{N} \lambda_i^{(k)} = \sum_{j=1}^{M} y_j$, $\forall k > 1$).

The main disadvantages of this EM algorithm are:

- it was shown to be a scaled steepest ascent algorithm [Kaufman 1993].

$$\hat{\lambda}_i^{(k+1)} = \hat{\lambda}_i^{(k)} + \hat{\lambda}_i^{(k)} \cdot \left. \frac{\partial L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}} \right|_{\boldsymbol{\lambda}=\hat{\boldsymbol{\lambda}}^{(k)}}$$

  therefore its convergence rate can be poor, especially when $L(\boldsymbol{\lambda}|\boldsymbol{y})$ contains "valleys" (when the eigenvalues of the Hessian $\boldsymbol{H}(\boldsymbol{\lambda})$ are very different) [Boyd and Vandenberghe 2004; Press et al. 2007]. The convergence becomes slow also when the values of $\hat{\lambda}_i^{(k)}$ approach zero, because the effective "step" of each iteration becomes very small.
- It is not possible to implement the algorithm batch-wise to simultaneously unmix groups of pixels. A distinct run of the EM algorithm is required for each measurement $\boldsymbol{y}$ (each pixel).
- The algorithm cannot be implemented in a "pyramid configuration", where groups of pixels are unmixed simultaneously so that the ML $\hat{\boldsymbol{\lambda}}_{ML}$ of the pixels are constrained based on the ML of the binned pixels (see Section 2.4.4).

In order to avoid the slow convergence problem of steepest ascent algorithms, several modifications of the standard EM algorithm have been proposed [Kaufman 1987, 1993; Tsuiet al. 1991]. The standard EM algorithm updates $\hat{\lambda}_i^{(k+1)}$ along the direction defined by the gradient $\left. \frac{\partial L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}} \right|_{\boldsymbol{\lambda}=\hat{\boldsymbol{\lambda}}^{(k)}}$ using the step $\hat{\lambda}_i^{(k)} \left\| \left. \frac{\partial L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}} \right|_{\boldsymbol{\lambda}=\hat{\boldsymbol{\lambda}}^{(k)}} \right\|$. The "line search" modification suggests to update the ML estimates as

$$\hat{\lambda}_i^{(k+1)} = \hat{\lambda}_i^{(k)} + \alpha \cdot \hat{\lambda}_i^{(k)} \cdot \left. \frac{\partial L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}} \right|_{\boldsymbol{\lambda}=\hat{\boldsymbol{\lambda}}_{ML}}$$

This approach keeps the same update direction as the standard EM algorithm but modifies the step size. The value of α is chosen by a line search method (e.g. the backtracking method [Nocedal and Wright 2006; Press et al. 2007]) to maximize $L(\boldsymbol{\lambda}|\boldsymbol{y})$ while $\hat{\lambda}_i^{(k+1)} \geq 0$. The preconditioned conjugate gradient (PCG) modification [Kaufman 1987] picks the update direction using a conjugate gradient method for nonlinear systems, such as the Fletcher-Reeves algorithm [Shewchuk 1994; Press et al. 2007], and then runs a line search method to find the optimal step that maximizes $L(\boldsymbol{\lambda}|\boldsymbol{y})$ while $\hat{\lambda}_i^{(k+1)} \geq 0$. Both modified EM algorithms provide results that satisfy the non-negativity (as long as the initial guess is non-negative) and additivity (as long as $\sum_{i=1}^{N} \lambda_i^{(0)} = \sum_{j=1}^{M} y_j$) constraints [Shepp and Vardi 1982]. For all three EM algorithms it is important to make sure that none of the initial guess elements are zero $\lambda_i^{(0)} \neq 0$ otherwise the algorithms will "stuck" and may not reach the ML solution.

Even though the standard EM algorithm suffers from slow convergence, the computation cost at each iteration step is not significant. On the other hand, the two modified EM algorithms require significantly more computation per iteration. It is not sure that the added computation will result in sufficiently less iteration steps so that overall they require less computation per pixel. These two modified EM algorithms were extremely effective in accelerating the processing of PET and SPECT imaging, where a multiple source-multiple detector problem of large dimensions (0^5 <N<

10^7, 10^2<M<10^3) needs to be solved once for each imaging session [Kaufman 1987]. In the case of spectral unmixing, processing each image requires solving million distinct small scale (usually 2<N<6, 8<M<32) multiple source-multiple detector problems.

Including Detector Noise

For low-signal level data, it is necessary to include detector noise in the observation model, because significant noise can corrupt unmixing results. The detector noise is described by the spectrum vector $\boldsymbol{s}_N$ and the noise rate λ_N. The rate of photon arrivals at each channel becomes:

$$\boldsymbol{d} = \boldsymbol{S} \cdot \boldsymbol{\lambda} + \boldsymbol{s}_N \cdot \lambda_N \qquad [2.4\text{-}6]$$

In this case, noise rate λ_N should be treated as known. Including noise as an unknown quantity will deteriorate significantly the estimation of source emission rates $\boldsymbol{\lambda}$ in low-signal levels. The reason is that because the emission vector $\boldsymbol{s}_N$ is usually approximately constant over all M channels, including $\boldsymbol{s}_N$ in the spectral matrix and treating λ_N as unknown will deteriorate significantly unmixing precision (by reducing the $\bar{\xi}$ and the G_i metrics of all sources, see Sections 2.4.5 and 2.4.6). When λ_N is treated as a known quantity, the derivation of the EM algorithm provides a result identical to Eq. 2.4-4, however in this case instead of Eq. 2.4-5, the term $\hat{d}_j^{(k)}$ should include the effect of noise:

$$\hat{d}_j^{(k)} = \sum_{l=1}^{N} s_{jl} \cdot \hat{\lambda}_l^{(k)} + s_{N,j} \cdot \lambda_N = \boldsymbol{t}_j^T \cdot \hat{\boldsymbol{\lambda}}^{(k)} + s_{N,j} \cdot \lambda_N \qquad [2.4\text{-}7]$$

In this case the EM algorithm still satisfies non-negativity for $\hat{\lambda}_i^{(k)}$.

Poisson mixing model summary

Observation model: $L(\boldsymbol{\lambda}|\boldsymbol{y}) = -\|\boldsymbol{\lambda}\|_1 + \sum_{j=1}^{M}\{y_j \cdot \log(d_j) - \log(y_j!)\}$

ML estimation: $\hat{\lambda}_i^{(k+1)} = \hat{\lambda}_i^{(k)} \cdot \sum_{j=1}^{M}\{s_{ji}\frac{y_j}{\hat{d}_j^{(k)}}\}$

$\hat{d}_j^{(k)} = \boldsymbol{t}_j^T \cdot \hat{\boldsymbol{\lambda}}^{(k)} + s_{N,j} \cdot \lambda_N$

2.4.4 Gaussian Mixing Model

The Gaussian model is an approximation to the Poisson mixing model. It exploits the fact that under appropriate conditions the Poisson probability mass function (PMF) can be approximated by a Gaussian probability density function (PDF).

The Gaussian mixing model describes the origin of stochastic nature of the detected signal $\boldsymbol{y}$ as a Gaussian zero-mean added noise. Several Gaussian mixing models exist that differ on the assumed properties of the Gaussian noise. The main advantage of Gaussian mixing models is that they provide linear equations for ML estimation, which can be solved very efficiently and be orders of magnitude faster than unmixing calculations based on the Poisson mixing model.

Gaussian approximations to Poisson distribution

The Gaussian model can be derived from the Poisson model, by approximating the Poisson probability distribution that describes the probability of detecting y_j photons in each channel. The cumulative distribution function (CDF) of the Poisson distribution $\wp_P(y;\lambda)$ of rate λ is approximated via the CDF $\Phi(w)$ of a standard normal distribution:

$$\wp_P(y;\lambda) = \sum_{x=1}^{y} P(x;\lambda) \approx \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-\frac{w(y,\lambda)^2}{2}} dw = \Phi(w(y,\lambda))$$

where $P(x;\lambda)$ is the PMF of a Poisson distribution of rate λ. The approximation of the Poisson PMF is found by Taylor expanding the CDF $\wp_P(y-1;\lambda)$ around y and keeping the first terms:

$$P(y;\lambda) = \wp_P(y;\lambda) - \wp_P(y-1;\lambda) \approx \Phi\big(w(y,\lambda)\big) - \Phi(w(y-1,\lambda)) \Rightarrow$$

$$P(y;\lambda) \approx \frac{d\Phi}{dw} \cdot \frac{\partial w}{\partial y} = \varphi(w(y,\lambda)) \cdot \frac{\partial w}{\partial y}$$

where $\varphi(w) = (2\pi)^{-0.5}\exp(-0.5w^2)$ is the PDF of the standard normal distribution. Several approximations of the Poisson vary in the way that they define $w(y,\lambda)$ as a function of y and λ (summarized in [Rich 2009]). In this thesis, there is emphasis in computationally-efficient approximations of the Poisson PMF at low (λ<50) and moderate (λ<200) photon counts. Since the approximations will be used for ML inference in millions of pixels, it is preferred to use approximations that result in closed-form solutions of ML estimates in order to avoid iterative numerical methods. Such approximations are the ones where the derivative $\frac{\partial w}{\partial y}$ is not a function of the parameter $\boldsymbol{\lambda}$. One such approximation is $w(y,\lambda) = \frac{y-\lambda}{\sigma}$. Applying this approximation for the emission detected at each channel provides the Gaussian observation model.

Gaussian observation model

According to the Gaussian observation model, the detected signal y_j at channel j is the sum of a deterministic part that depends on the source emission rates $d_j = \boldsymbol{t}_j^T\boldsymbol{\lambda}$, and a stochastic part described as zero-mean Gaussian noise of standard deviation σ_j. The measured signal $y_j \sim N(y_j; d_j, \sigma_j)$ at channel j follows normal distribution with mean d_j and standard deviation σ_j, and the resulting observation model is:

$$p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}) = \prod_{j=1}^{\boldsymbol{M}} P\big(y_j; \boldsymbol{t}_j^T\boldsymbol{\lambda}\big) = \prod_{j=1}^{\boldsymbol{M}} \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{1}{2}\frac{(y_j-d_j)^2}{\sigma_j^2}\right) \qquad [2.4\text{-}8]$$

The log-likelihood is:

$$L(\boldsymbol{\lambda}|\boldsymbol{y}) = log\left(p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})\right) = -\frac{M}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{M}\left\{\frac{(y_j-d_j)^2}{\sigma_j^2} + \log(\sigma_j)\right\} \qquad [2.4\text{-}9]$$

Maximum likelihood solution

The ML estimator $\hat{\boldsymbol{\lambda}}_{ML}$ for the Gaussian mixing model is the value of $\boldsymbol{\lambda}$ that maximizes log-likelihood $L(\boldsymbol{\lambda}|\boldsymbol{y})$, or equivalently minimizes the corresponding chi-square metric [Sivia and Skilling 2006]:

$$\mathrm{X}^2(\boldsymbol{\lambda}) = \sum_{j=1}^{M} \frac{1}{2}\frac{(y_j - d_j)^2}{\sigma_j^2}$$

Setting the gradient of $\mathrm{X}^2(\boldsymbol{\lambda})$ equal to zero, provides the following set of N equations for $\hat{\boldsymbol{\lambda}}_{ML}$:

$$\nabla\mathrm{X}^2(\boldsymbol{\lambda}) = \boldsymbol{0} \;\Rightarrow\; \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_\sigma^{-2} \cdot \boldsymbol{S} \cdot \hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_\sigma^{-2} \cdot \boldsymbol{y} \qquad [2.4.10]$$

where $\boldsymbol{\Lambda}_\sigma = diag(\sigma_j)$. The intuitive approximation for the noise variance $\sigma_j = \sqrt{d_j}$ leads to the Gaussian variable noise (GVN) ML unmixing equations:

$$\boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_d^{-1} \cdot \boldsymbol{S} \cdot \hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_d^{-1} \cdot \boldsymbol{y} \qquad [2.4.11]$$

where $\Lambda_d = diag(d_j)$. The Hessian matrix $\boldsymbol{H}(\boldsymbol{\lambda})$ of the log-likelihood function $L(\boldsymbol{\lambda}|\boldsymbol{y})$:

$$\boldsymbol{H}(\boldsymbol{\lambda}) = \boldsymbol{S}^T \cdot diag(d_j^{-1}) \cdot \boldsymbol{S}$$

is positive definite because $d_j > 0$, therefore $\mathrm{X}^2(\boldsymbol{\lambda})$ is a convex function with a global minimum.

The assumption that noise variance is common in all channels ($\sigma_j = \sigma$), leads to the Gaussian common noise (GCN) ML unmixing equations:

$$\boldsymbol{S}^T \cdot \boldsymbol{S} \cdot \hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{S}^T \cdot \boldsymbol{y} \qquad [2.4.12]$$

Numerical Implementation – Solving Systems of Linear Equations

The inference of $\hat{\boldsymbol{\lambda}}_{ML}$ via the Gaussian mixing model requires the solution of a system of linear equations of the form $\boldsymbol{A}\hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{b}$ (eq. 2.4.11 or 2.4.12). When the N×N coefficient matrix $\boldsymbol{A}$ of this linear system is full-rank (see discussion in next section) a unique solution exists. When the non-negativity constraints on the elements of vector $\hat{\boldsymbol{\lambda}}_{ML}$ are ignored, the ML estimate can be calculated either by Gauss elimination $\hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{A}\backslash\boldsymbol{b}$, or by matrix inversion $\hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{A}^{-1}\boldsymbol{b}$, where $\boldsymbol{A}^{-1}$ is calculated using matrix computation schemes such as singular value decomposition (SVD).

When non-negativity constraints for the elements of $\hat{\boldsymbol{\lambda}}_{ML}$ are considered in the ML solution, the ML estimate can be calculated using the method of non-negative least squares (NNLS). NNLS is an iterative algorithm that finds the vector $\hat{\boldsymbol{\lambda}}_{ML}$ that minimizes the cost $J = \left\|\boldsymbol{A}\hat{\boldsymbol{\lambda}}_{ML} - \boldsymbol{b}\right\|$ subject to the constraints $\lambda_{ML,i} \geq 0$. NNLS is an active-passive set method: at each iteration a set of $\lambda_{ML,i}$ is set to 0 (active set), and the remaining $\lambda_{ML,i}$ (passive set) are calculated via the left pseudo-inverse $\boldsymbol{A}_p^+$ of the matrix $\boldsymbol{A}_p$ (the matrix made of the columns of $\boldsymbol{A}$ that correspond to the passive set). NNLS iterations search for the passive set that minimizes the cost function J while satisfying the non-negativity constraints. NNLS finds the optimal solution within no more iterations than the number of $\boldsymbol{A}$ rows [Lawson & Hanson 1974].

The standard NNLS algorithm needs to be implemented once per pixel. Lately, a modification of the standard NNLS algorithm enables the simultaneous solution of the NNLS problem (minimize $J = \left\|\boldsymbol{A}\hat{\boldsymbol{\lambda}}_{ML} - \boldsymbol{b}\right\|$ subject to $\lambda_{ML,i} \geq 0$) for a large number of vectors $\boldsymbol{b}$ as long as the coefficient matrix $\boldsymbol{A}$ is common [Bro and de Jong 1997; van Benthem and Keenan 2004, 2008]. Fast implementations of NNLS (fNNLS) take into advantage that i) the major computation cost of NNLS is the calculation of the pseudo-inverse $\boldsymbol{A}_p^+$ for all the possible passive sets, and that ii) the optimal solution for many $\boldsymbol{b}$ vectors corresponds to the same passive set therefore it is not necessary to calculate $\boldsymbol{A}_p^+$ more than once for each passive set.

Numerical Implementation - Common channel noise

In the case of common noise variance ($\sigma_j = \sigma$) the set of ML estimation equations (Eq. 2.4.12) has a coefficient matrix $\boldsymbol{A} = \boldsymbol{S}^T\boldsymbol{S}$ that is constant and independent of data $\boldsymbol{y}$. This results in very efficient computations for both cases of ignoring or including non-negativity constraints for $\hat{\boldsymbol{\lambda}}_{ML}$.

When non-negativity constraints for $\hat{\boldsymbol{\lambda}}_{ML}$ are ignored, eq. 2.4.12 can be solved by inverting the coefficient matrix:

$$\hat{\boldsymbol{\lambda}}_{ML} = (\boldsymbol{S}^T \cdot \boldsymbol{S})^{-1} \cdot \boldsymbol{S}^T \cdot \boldsymbol{y} = \boldsymbol{S}^+ \cdot \boldsymbol{y} \qquad [2.4.13]$$

$\boldsymbol{S}^+ = (\boldsymbol{S}^T \cdot \boldsymbol{S})^{-1} \cdot \boldsymbol{S}^T$ is the left pseudo-inverse of the M×N spectral matrix $\boldsymbol{S}$, which can be calculated via singular value decomposition [Strang 1988; Press et al. 2007]:

$$\boldsymbol{S} = \boldsymbol{U} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}^T \quad , \quad \boldsymbol{\Sigma} = \left[diag(\varsigma_j) \quad \mathbf{0}_{(M-N)\times N}\right]^T$$
$$\boldsymbol{S}^+ = \boldsymbol{V} \cdot \boldsymbol{\Sigma}^+ \cdot \boldsymbol{U}^T \quad , \quad \boldsymbol{\Sigma}^+ = \left[diag(\varsigma_j^{-1}) \quad \mathbf{0}_{N\times(M-N)}\right] \qquad [2.4.14]$$

Where $\mathbf{0}_{i\times j}$ is an i×j matrix of zero elements, $\boldsymbol{U}$ and $\boldsymbol{V}$ are M×M and N×N orthogonal matrices respectively, and ς_j (j = 1, 2,..., N) are the singular values of $\boldsymbol{S}$. When M>N (number of detectors is larger than number of sources) and $rank(\boldsymbol{S}) = N$ (the columns of $\boldsymbol{S}$ are independent; equivalently the emission spectra of the sources are distinct and not linearly-dependent), there exist N non-zero singular values ς_j for $\boldsymbol{S}$, the left inverse $\boldsymbol{S}^+$ is full rank, and $\hat{\boldsymbol{\lambda}}_{ML}$ is unique. Since the coefficient matrix $\boldsymbol{A} = \boldsymbol{S}^T\boldsymbol{S}$ is data-independent, $\boldsymbol{S}^+$ needs to be computed only once for all image pixels that have common spectral matrix $\boldsymbol{S}$. Once the pseudo-inverse $\boldsymbol{S}^+$ is calculated, spectral unmixing for each pixel involves a single matrix multiplication (Eq. 2.4.13) and can be implemented in batch (unmixing large numbers of pixels using a single matrix multiplication), resulting in very low computational cost.

When the non-negativity constraints for $\hat{\boldsymbol{\lambda}}_{ML}$ are considered, Eq. 2.4.12 can be solved iteratively using the NNLS algorithm. Since the coefficient matrix $\boldsymbol{S}^T \cdot \boldsymbol{S}$ is common to all measurements, it is possible to apply the fast NNLS (fNNLS) implementations to solve Eq. 2.4.12 for thousands of pixels simultaneously as long as all pixels share a common spectral matrix $\boldsymbol{S}$. It turns out that such batch processing can provide non-negative spectral unmixing in a very cost-effective way, especially if implemented in GPU-based parallel computer architectures [Luo and Duraiswami 2011].

As a final note for the Gaussian common-noise mixing model, many papers in the literature describe linear unmixing as the solution to the least-squares problem

$$\boldsymbol{S} \cdot \hat{\boldsymbol{\lambda}}_{ML} \approx \boldsymbol{y} \qquad [2.4.15]$$

The solution of this system of M equations for N unknowns is not necessary identical to the solution of Eq. 2.4.12 (N equations for N unknowns). The solutions of Eq. 2.4.12 and 2.4.15 are identical when their unconstrained solutions contain no negative element; in this case Eq. 2.4.12 corresponds to the canonical equations of least squares problems. The solutions of Eq. 2.4.12 and 2.4.15 differ when the non-negativity constrains are imposed, which is usually the case at low signal levels. This can be proven by the Karush–Kuhn–Tucker (KKT) conditions when the active constraint set in non-empty (i.e. the non-negativity constraint is imposed for some element of the $\hat{\boldsymbol{\lambda}}_{ML}$ vector).

Numerical Implementation – Variable channel noise

In the case of channel-dependent noise variance ($\sigma_j = \sqrt{d_j}$) the coefficient matrix $\boldsymbol{A} = \boldsymbol{S}^T\boldsymbol{D}_{\hat{d}}^{-1}\boldsymbol{S}$ of the ML estimation equation (Eq. 2.4.11) depends on the channel emission rate d_j. Since d_j are not known a-priori, they can be estimated as either $\hat{d}_j = y_j$ or $\hat{d}_j = \boldsymbol{t}_j^T\hat{\boldsymbol{\lambda}}^{(0)} + s_{N,j} \cdot \lambda_N$, where $\hat{\boldsymbol{\lambda}}^{(0)}$ is an easy to compute estimate of $\hat{\boldsymbol{\lambda}}_{ML}$ (for example the solution of the common noise Gaussian mixing model of Eq. 2.4.12).

$$\boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\hat{d}}^{-1}(\boldsymbol{y}) \cdot \boldsymbol{S} \cdot \hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\hat{d}}^{-1}(\boldsymbol{y}) \cdot \boldsymbol{y} \qquad [2.4.16]$$

where $\boldsymbol{\Lambda}_{\hat{\boldsymbol{d}}} = diag(\hat{d}_j(\boldsymbol{y}))$. Eq. 2.4.11 can be solved by either Gauss elimination (when non-negativity constraints for $\hat{\boldsymbol{\lambda}}_{ML}$ are not considered) or by NNLS (when non-negativity constraints for $\hat{\boldsymbol{\lambda}}_{ML}$ are considered). Since the coefficient matrix $\boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\hat{\boldsymbol{d}}}^{-1}(\boldsymbol{y}) \cdot \boldsymbol{S}$ is data-dependent, it is not possible to process pixels in batch (either by Gauss elimination, or by fNNLS). However, Eq. 2.4.16 can be re-written as:

$$\boldsymbol{S}^T \cdot \boldsymbol{S} \cdot \hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{S}^T \cdot \boldsymbol{S} \cdot \psi(\boldsymbol{y}) \qquad [2.4.17]$$

where

$$\psi(\boldsymbol{y}) = (\boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\hat{\boldsymbol{d}}}^{-1}(\boldsymbol{y}) \cdot \boldsymbol{S})^{-1} \cdot \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\hat{\boldsymbol{d}}}^{-1}(\boldsymbol{y}) \cdot \boldsymbol{y}$$

The solution of Eq. 2.4.17is implemented in two steps. First, $\psi(\boldsymbol{y})$ is calculated pixel-wise by Gauss elimination

$$\psi(\boldsymbol{y}) = \left(\boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\hat{\boldsymbol{d}}}^{-1}(\boldsymbol{y}) \cdot \boldsymbol{S}\right) \backslash \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\hat{\boldsymbol{d}}}^{-1}(\boldsymbol{y}) \cdot \boldsymbol{y} \qquad [2.4.18]$$

The coefficient matrix $\boldsymbol{S}^T \cdot \boldsymbol{S}$ in Eq. 2.4.17is identical to Eq. 2.4.12 and data-independent, therefore Eq. 2.4.17can be solved in batch mode either by SVD decomposition or by fNNLS. KKT conditions prove that Eq. 2.4.11 and 2.4.17 provide identical solutions $\hat{\boldsymbol{\lambda}}_{ML}$ when $\boldsymbol{\Lambda}_{\hat{\boldsymbol{d}}}^{-1}(\boldsymbol{y})$ is full-rank ($\hat{d}_j \neq 0$) and all elements $\hat{\lambda}_{ML,i} > 0$ are non-negative. However, when in the solution there exists at least one i so that $\hat{\lambda}_{ML,i} = 0$, then the solutions of Eq. 2.4.11 and 2.4.17are different.

Including Detector Noise

As it was discussed in Section 2.4.3, when the signal level is close to the level of background detector noise, then it is necessary to include the detector noise signal in the observation model, because otherwise noise can corrupt unmixing results. Incorporating the signal properties ($\boldsymbol{s}_N$, λ_N) in the observation model results in the identical ML estimator equations (Eq 2.4.10, 2.4.12, 2.4.13 and 2.4.17) where $\hat{d}_j^{(k)}$ is calculated via Eq. 2.4-7 instead of Eq. 2.4-5, and $\boldsymbol{y}$ is substituted by $\boldsymbol{y}^*$:

$$\boldsymbol{y}^* = \boldsymbol{y} - \boldsymbol{s}_N \cdot \lambda_N$$

Incorporating noise does not affect the discussion on the application of appropriate numerical solution algorithms.

Gaussian mixing model

Observation model: $L(\boldsymbol{\lambda}|\boldsymbol{y}) = log\left(p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})\right) = -\frac{M}{2}\log(2\pi) - \sum_{j=1}^{M}\{\frac{1}{2}\frac{(y_j - d_j)^2}{\sigma_j^2} + \log(\sigma_j)\}$

ML estimation (general case): $\boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\sigma}^{-2} \cdot \boldsymbol{S} \cdot \hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\sigma}^{-2} \cdot \boldsymbol{y}^*$

$\boldsymbol{y}^* = \boldsymbol{y} - \boldsymbol{s}_N \cdot \lambda_N$

Common noise ($\sigma_j = \sigma$): $\boldsymbol{S}^T \cdot \boldsymbol{S} \cdot \hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{S}^T \cdot \boldsymbol{y}^*$

SVD solution: $\hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{S}^+ \cdot \boldsymbol{y}^*$

$\boldsymbol{S}^+ = \boldsymbol{V} \cdot \boldsymbol{\Sigma}^+ \cdot \boldsymbol{U}^T$

fNNLS solution: $\boldsymbol{S}^T \cdot \boldsymbol{S} \cdot \hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{S}^T \cdot \boldsymbol{y}^* \quad , \quad \hat{\boldsymbol{\lambda}}_{ML,i} \geq 0$

Variable noise ($\sigma_j = \sqrt{\hat{d}_j}$): $\boldsymbol{S}^T \cdot diag^{-1}(\hat{\boldsymbol{d}}(\boldsymbol{y})) \cdot \boldsymbol{S} \cdot \hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\hat{\boldsymbol{d}}}^{-1}(\boldsymbol{y}) \cdot \boldsymbol{y}^*$

	$\boldsymbol{\Lambda}_{\hat{d}} = diag(\hat{\boldsymbol{d}}(\boldsymbol{y}))$, $\hat{\boldsymbol{d}} = \boldsymbol{S} \cdot \hat{\boldsymbol{\lambda}}^{(0)}(\boldsymbol{y}) + \boldsymbol{s}_N \cdot \lambda_N$
SVD solution:	$\hat{\boldsymbol{\lambda}}_{ML} = \psi(\boldsymbol{y})$
	$\psi(\boldsymbol{y}) = (\boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\hat{d}}^{-1}(\boldsymbol{y}) \cdot \boldsymbol{S}) \backslash (\boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\hat{d}}^{-1}(\boldsymbol{y}) \cdot \boldsymbol{y}^*)$
Approx. fNNLS solution:	$\boldsymbol{S}^T \cdot \boldsymbol{S} \cdot \hat{\boldsymbol{\lambda}}_{ML} = \boldsymbol{S}^T \cdot \boldsymbol{S} \cdot \psi(\boldsymbol{y})$, $\hat{\lambda}_{ML,i} \geq 0$

2.4.5 Theoretic Analysis of Spectral Unmixing Performance

Data-generating model

The measurement $\boldsymbol{y}$ at each pixel is generated by the data-generating model:

$$L^0(\boldsymbol{\lambda}|\boldsymbol{y}) = log\left(p^0_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})\right) = -(\|\boldsymbol{\lambda_0}\|_1 + \lambda_N) + \sum_{j=1}^{M}\{y_j \cdot \log\left(\boldsymbol{t}_j^{0\,T}\boldsymbol{\lambda}_0 + s_{N,j} \cdot \lambda_N\right) - \log(y_j!)\}$$

which corresponds to a Poisson mixing model Section 2.4.3. $\boldsymbol{\lambda}_0$ is the pixel-dependent actual source emission rate at the detector to be estimated, and $\boldsymbol{t}_j^0$ is the j-th row of the real spectral matrix $\boldsymbol{S}_0$. The data-generating model incorporates detector noise described by the emission spectrum $\boldsymbol{s}_N$, and the noise rate λ_N.

Estimation Error Definition

The performance of a spectral unmixing algorithm is evaluated based on two kinds of metrics:

- Estimation error: describes the statistical properties of estimation error $\tilde{\boldsymbol{\lambda}}(\boldsymbol{y}) = \hat{\boldsymbol{\lambda}}_{ML}(\boldsymbol{y}) - \boldsymbol{\lambda}_0$. Any ML estimate $\hat{\boldsymbol{\lambda}}_{ML}$ of $\boldsymbol{\lambda}_0$ is a random variable since it is a function of measurement **y** which is a random variable. The properties of $\tilde{\boldsymbol{\lambda}}(\boldsymbol{y})$ that quantify estimation error are:
  - accuracy (trueness), quantified by the estimation bias
    $$\boldsymbol{b} = E[\tilde{\boldsymbol{\lambda}}(\boldsymbol{y})]$$
  - precision (reproducibility), quantified by the estimation covariance matrix
    $$\boldsymbol{C}_{\hat{\lambda}} = E[\hat{\boldsymbol{\lambda}}_{ML} \cdot \hat{\boldsymbol{\lambda}}_{ML}^T].$$
- Computation cost: the CPU time that is required to unmix each pixel of a spectral image.

Sources of Estimation Error

ML estimation provides an estimate $\hat{\boldsymbol{\lambda}}_{ML}(\boldsymbol{y})$ of $\boldsymbol{\lambda}_0$ based on a measurement $\boldsymbol{y}$. The resulting estimation error $\tilde{\boldsymbol{\lambda}}(\boldsymbol{y}) = \hat{\boldsymbol{\lambda}}_{ML}(\boldsymbol{y}) - \boldsymbol{\lambda}_0$ arises due to three reasons:

- Poisson noise: Due to the stochastic nature of the photon detection process, each measurement $\boldsymbol{y}$ is a random vector whose elements depend on $\boldsymbol{\lambda}_0$ and $\boldsymbol{S}_0$. Since $\hat{\boldsymbol{\lambda}}_{ML}(\boldsymbol{y})$ is a function of $\boldsymbol{y}$, $\hat{\boldsymbol{\lambda}}_{ML}(\boldsymbol{y})$ is also a random vector. This randomness induced by the random nature of photon arrival causes estimation error.
- Spectral mixing: When photons emitted by N sources of overlapping emission spectra are detected by the same M-channel detector, the mixing process deteriorates the estimation performance compared to the case where each source could be detected separately.
- Model uncertainty: modeling errors (such as imperfect knowledge of the source number N or detected spectral matrix $\boldsymbol{S}_0$) and detector noise increase estimation error.

These three origins of estimation error $\tilde{\boldsymbol{\lambda}}(\boldsymbol{y})$ affect both the bias $\boldsymbol{b} = E[\tilde{\boldsymbol{\lambda}}(\boldsymbol{y})]$ and covariance matrix $\boldsymbol{C}_{\hat{\lambda}} = E[\tilde{\boldsymbol{\lambda}}_{ML} \cdot \tilde{\boldsymbol{\lambda}}_{ML}^T]$ of the estimator in different ways analyzed below.

Poisson Noise Imposes a Performance Bound on Spectral Unmixing

Given the source emission vector $\boldsymbol{\lambda}_0$ at the detector, any estimator for $\boldsymbol{\lambda}_0$ based on a measurement $\boldsymbol{y}$ would be limited by the Poisson nature of photon arrival. An "ideal" estimation implemented by an "ideal" (non-existing) detector that could somehow distinguish and detect the photons of each source in a separate channel would estimate $\lambda_{0,i}$ with variance equal to $\lambda_{0,i}$. Poisson noise does not induce estimation bias, however imposes a lower bound on the covariance matrix $\boldsymbol{C}_{\hat{\boldsymbol{\lambda}}}^{poisson} = diag(\boldsymbol{\lambda}_0)$ that depends only on the signal level $Y = \sum_i \lambda_{0,i}$. It is not possible to design a spectral unmixing algorithm that estimates the rate $\lambda_{0,i}$ with variance less than $\lambda_{0,i}$.

Analyzing the Effects of Mixing and Model Uncertainty

In addition to the limitation imposed by Poisson noise, the performance of any estimator $\hat{\boldsymbol{\lambda}}$ is deteriorated further due to source signal mixing and model uncertainty. The effect of the last two factors on the ML estimation error $\tilde{\boldsymbol{\lambda}}(\boldsymbol{y}) = \hat{\boldsymbol{\lambda}}_{ML}(\boldsymbol{y}) - \boldsymbol{\lambda}_0$ can be studied analytically by expanding the derivative of the log-likelihood around the actual emission rate vector $\boldsymbol{\lambda}_0$.

$$\frac{\partial L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}} \approx \left.\frac{\partial L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}}\right|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}_0} + \left.\frac{\partial^2 L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}\partial \boldsymbol{\lambda}^T}\right|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}_0} (\boldsymbol{\lambda} - \boldsymbol{\lambda}_0) + O((\boldsymbol{\lambda} - \boldsymbol{\lambda}_0)^2)$$

At the maximum likelihood solution $\hat{\boldsymbol{\lambda}}_{ML}(\boldsymbol{y})$ the gradient of the log-likelihood is zero, therefore after neglecting the $O((\boldsymbol{\lambda} - \boldsymbol{\lambda}_0)^2)$ terms, this provides a linear equation of the form

$$\boldsymbol{F} \cdot \tilde{\boldsymbol{\lambda}} = \boldsymbol{g} \qquad [2.4.19]$$

$$\boldsymbol{F} = -\left.\frac{\partial^2 L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}\partial \boldsymbol{\lambda}^T}\right|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}_0} \qquad [2.4.20]$$

$$\boldsymbol{g} = \left.\frac{\partial L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}}\right|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}_0} \qquad [2.4.21]$$

The following two sub-sections derive expressions for the estimation bias $\boldsymbol{b} = E[\tilde{\boldsymbol{\lambda}}(\boldsymbol{y})]$ and covariance $\boldsymbol{C}_{\hat{\boldsymbol{\lambda}}} = E[\hat{\boldsymbol{\lambda}}_{ML} \cdot \hat{\boldsymbol{\lambda}}_{ML}^T]$ based on the exact form of $\boldsymbol{F}$ and $\boldsymbol{g}$, which depend on the mixing model utilized for ML estimation.

Estimation Error Analysis – Poisson mixing model

When ML estimation is based on the Poisson mixing model (Eq. 2.4-3), $\boldsymbol{F}$ and $\boldsymbol{g}$ become:

$$\boldsymbol{F}(\boldsymbol{y}) = -\left.\frac{\partial^2 L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}\partial \boldsymbol{\lambda}^T}\right|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}_0} = \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_y \cdot \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-2} \cdot \boldsymbol{S}$$

$$\boldsymbol{g}(\boldsymbol{y}) = \left.\frac{\partial L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}}\right|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}_0} = -\boldsymbol{1}_{N\times 1} + \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-1} \cdot \boldsymbol{y}$$

Where $\boldsymbol{1}_{i\times j}$ is a i×j matrix of unitary elements, $\boldsymbol{\Lambda}_d = diag(\boldsymbol{d})$, $\boldsymbol{d} = \boldsymbol{S}\boldsymbol{\lambda}_0$, and $\boldsymbol{\Lambda}_y = diag(\boldsymbol{y})$.

When modeling errors are small and source rates $\boldsymbol{\lambda}_0$ are large enough so that $y_j \approx \boldsymbol{t}_j^T \boldsymbol{\lambda}_0$, then the $\boldsymbol{F}$ matrix becomes data-independent:

$$\boldsymbol{F} \approx \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-1} \cdot \boldsymbol{S} \qquad [2.4.22]$$

In this case the elements of the channel rate vector $\boldsymbol{d}$ are large enough so that the PDF of measurement y_j at each channel can be approximated by a normal distribution of mean and variance equal to $\boldsymbol{t}_j^T\boldsymbol{\lambda}_0$, and $\boldsymbol{g}$ becomes a vector of Gaussian random variables, therefore:

$$E[\boldsymbol{g}] \equiv \boldsymbol{\mu}_{\boldsymbol{g}} = -\mathbf{1}_{N\times 1} + \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-1} \cdot E[\boldsymbol{y}] = -\boldsymbol{S}^T \cdot \boldsymbol{e} \qquad [2.4.23]$$

$$E[\boldsymbol{g}\boldsymbol{g}^T] \equiv \boldsymbol{J} = \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-1} \cdot \boldsymbol{C}_{\boldsymbol{y}} \cdot \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-1} \cdot \boldsymbol{S} + \boldsymbol{S}^T \cdot \boldsymbol{e} \cdot \boldsymbol{e}^T \cdot \boldsymbol{S} \qquad [2.4.24]$$

where $\boldsymbol{C}_{\boldsymbol{y}} = diag\ (\boldsymbol{S}_{\mathbf{0}} \cdot \boldsymbol{\lambda}_{\mathbf{0}})$ is the covariance matrix of $\boldsymbol{y}$. The modeling error vector $\boldsymbol{e}$ describes modeling errors due to imperfect knowledge of the spectral matrix $\boldsymbol{S}_0$:

$$\boldsymbol{e} = \mathbf{1}_{M\times 1} - \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-1} \cdot \boldsymbol{S}_{\mathbf{0}} \cdot \boldsymbol{\lambda}_{\mathbf{0}} \qquad [2.4.25]$$

where the j-th element $e_j = 1 - \boldsymbol{t}_j^{0T}\boldsymbol{\lambda}_{\mathbf{0}}/\boldsymbol{t}_j^T\boldsymbol{\lambda}_{\mathbf{0}}$ describes the imperfect knowledge of source emission spectra at channel j. $\boldsymbol{J}$ is the Fisher information matrix of the Poisson model.

Combining Eq. 2.4.19-24, since $\boldsymbol{g}$ is a Gaussian random vector, then estimation error $\tilde{\boldsymbol{\lambda}}$ becomes a Gaussian random vector of mean $\boldsymbol{F}^{-1}\boldsymbol{\mu}_{\boldsymbol{g}}$ and covariance $(\boldsymbol{F} \cdot (\boldsymbol{J} - \boldsymbol{\mu}_{\boldsymbol{g}} \cdot \boldsymbol{\mu}_{\boldsymbol{g}}^T)^{-1} \cdot \boldsymbol{F})^{-1}$:

$$\tilde{\boldsymbol{\lambda}} \sim N\left(\boldsymbol{F}^{-1}\boldsymbol{\mu}_{\boldsymbol{g}}, (\boldsymbol{F} \cdot (\boldsymbol{J} - \boldsymbol{\mu}_{\boldsymbol{g}} \cdot \boldsymbol{\mu}_{\boldsymbol{g}}^T)^{-1} \cdot \boldsymbol{F})^{-1}\right)$$

where the Fisher information matrix can be written as:

$$\boldsymbol{J} = \boldsymbol{S}^T \cdot ((\boldsymbol{I} - diag(\boldsymbol{e})) \cdot \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-1} + \boldsymbol{e} \cdot \boldsymbol{e}^T) \cdot \boldsymbol{S}$$

In this case, the estimator bias equals $\boldsymbol{b} = E[\tilde{\boldsymbol{\lambda}}] = \boldsymbol{F}^{-1}\boldsymbol{\mu}_{\boldsymbol{g}}$.

When $\boldsymbol{\lambda}_{\mathbf{0}}$ are large and there are no modeling errors ($\boldsymbol{S} = \boldsymbol{S}_0$), then $\boldsymbol{e} = \mathbf{0}$, $\boldsymbol{\mu}_{\boldsymbol{g}} = \mathbf{0}$, and $\boldsymbol{J} = \boldsymbol{F}$ [Konishi and Kitagawa 2008], and estimation error is a zero-mean Gaussian vector of covariance $\boldsymbol{J}^{-1} = \boldsymbol{F}^{-1}$:

$$\tilde{\boldsymbol{\lambda}} \sim N(\mathbf{0}, \boldsymbol{J}^{-1})$$

Equivalently, the estimator bias equals zero (the estimator is unbiased) and the estimator variance equals $\boldsymbol{C}_{\hat{\boldsymbol{\lambda}}} = \boldsymbol{J}^{-1}$. In summary, when the elements of $\boldsymbol{\lambda}_{\mathbf{0}}$ are large enough and there is perfect knowledge of the source emission spectra, the ML estimator based on Poisson mixing model is unbiased and the error covariance equals the inverse of the Fisher information matrix that is approximated as $\boldsymbol{J} = \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-1} \cdot \boldsymbol{S}$.

When the elements of $\boldsymbol{\lambda}_{\mathbf{0}}$ are not large enough, the above approximations are not valid. Nevertheless, it is useful to know how "large" $\boldsymbol{\lambda}_0$ elements need to be or how small $\boldsymbol{\Lambda}$ elements should be so that the above analytic relations provide reasonable approximations.

Estimation Error Analysis – Gaussian mixing model

When ML estimation is based on the Gaussian mixing model, $\boldsymbol{F}$ and $\boldsymbol{g}$ become:

$$\boldsymbol{F} = -\left.\frac{\partial^2 L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial\boldsymbol{\lambda}\partial\boldsymbol{\lambda}^T}\right|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}_0} = \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{\sigma}}^{-2} \cdot \boldsymbol{S}$$

$$\boldsymbol{g}(\boldsymbol{y}) = \left.\frac{\partial L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial\boldsymbol{\lambda}}\right|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}_0} = -\boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{\sigma}}^{-2} \cdot (\boldsymbol{S} \cdot \boldsymbol{\lambda}_0 - \boldsymbol{y})$$

In this case, $\boldsymbol{F}$ is always data-independent. The noise approximation $\sigma_j = \sqrt{\boldsymbol{t}_j^T \boldsymbol{\lambda_0}}$ provides the identical $\boldsymbol{F} \approx \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-1} \cdot \boldsymbol{S}$ result as in the case of the Poisson mixing model.

When $\boldsymbol{\lambda_0}$ are large (see discussion on the Poison mixing model) the measurement y_j can be approximated as a random variable that follows normal distribution of mean and variance equal to $\boldsymbol{t}_j^T \boldsymbol{\lambda_0}$. In this case $\boldsymbol{g}$ is a vector of Gaussian random variables where:

$$E[\boldsymbol{g}] \equiv \boldsymbol{\mu_g} = -\boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{\sigma}}^{-2} \cdot (\boldsymbol{S} \cdot \boldsymbol{\lambda}_0 - \boldsymbol{S}_0 \cdot \boldsymbol{\lambda}_0)$$

$$E[\boldsymbol{g}\boldsymbol{g}^T] \equiv \boldsymbol{J} = \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{\sigma}}^{-2} \cdot (\boldsymbol{C_y} + \boldsymbol{\Lambda}_d \cdot \boldsymbol{e} \cdot \boldsymbol{e}^T \cdot \boldsymbol{\Lambda}_d) \cdot \boldsymbol{\Lambda}_{\boldsymbol{\sigma}}^{-2} \cdot \boldsymbol{S}$$

These two equations become identical to Eq. 2.4.23, 2.4.24 when $\sigma_j = \sqrt{\boldsymbol{t}_j^T \boldsymbol{\lambda_0}}$. The estimation error $\tilde{\boldsymbol{\lambda}}(\boldsymbol{y})$ is a Gaussian vector of mean $\boldsymbol{F}^{-1}\boldsymbol{\mu_g}$ and covariance matrix $\boldsymbol{F}^{-1}(\boldsymbol{J} - \boldsymbol{\mu_g} \cdot \boldsymbol{\mu_g}^T)\boldsymbol{F}^{-1}$.

In the absence of modeling errors ($\boldsymbol{e} = \boldsymbol{0}$), then $\boldsymbol{\mu_g} = \boldsymbol{0}$ (the ML estimator becomes unbiased) and to very good approximation $\boldsymbol{J} = \boldsymbol{F}$ (when $\boldsymbol{\lambda} = \boldsymbol{\lambda_0}$ the Gaussian mixing model $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})$ fits the "real" distribution $p^0_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})$ very well), and the covariance of the estimator error equals $\boldsymbol{C}_{\hat{\boldsymbol{\lambda}}} = \boldsymbol{J}^{-1} = (\boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-1} \cdot \boldsymbol{S})^{-1}$. These results suggest that when source rates $\boldsymbol{\lambda_0}$ are large and the spectral matrix is known exactly, the Gaussian mixing model provides the same ML estimation performance as the more complex and computationally expensive Poison mixing model.

Analysis of the Fisher information matrix

The previous sub-sections demonstrate that the effects of mixing and model uncertainty on spectral unmixing performance depend strongly on the properties of the Fisher information matrix $\boldsymbol{J} = E[\boldsymbol{g}\boldsymbol{g}^T]$, where $\boldsymbol{g} = \frac{\partial L(\boldsymbol{\lambda}|\boldsymbol{y})}{\partial \boldsymbol{\lambda}}\Big|_{\boldsymbol{\lambda}=\boldsymbol{\lambda_0}}$ and the expectation $E[\cdot]$ is calculated with respect to data $\boldsymbol{y}$. The Fisher information matrix is an important concept in statistics, involved in the Cramer-Rao bound (CRB) on the covariance $\boldsymbol{C}_{\hat{\boldsymbol{\lambda}}}(\boldsymbol{\lambda})$ of an unbiased estimator $\hat{\boldsymbol{\lambda}}(\boldsymbol{y})$ [Rao 2001]:

$$\boldsymbol{C}_{\hat{\boldsymbol{\lambda}}}(\boldsymbol{\lambda}) - \boldsymbol{J}^{-1}(\boldsymbol{\lambda}) \geq 0$$

where $\geq$ means that the matrix $\boldsymbol{C}_{\hat{\boldsymbol{\lambda}}}(\boldsymbol{\lambda}) - \boldsymbol{J}^{-1}(\boldsymbol{\lambda})$ is positive semi-definite [Strang 1988]. The Fisher information matrix $\boldsymbol{J}(\boldsymbol{\lambda})$ describes how the gradient magnitude of the log-likelihood of a probabilistic model varies with respect to the model variables $\boldsymbol{\lambda}$ that need to be inferred, and therefore describes the ease of performing inference based on this model.

In the previous sub-sections, it was shown that when source rates $\boldsymbol{\lambda}_0$ are large enough and there is no model uncertainty, then both mixing models (Poison, Gaussian) are efficient: they provide unbiased estimation with covariance matrix that approaches the CRB $\boldsymbol{C}_{\hat{\boldsymbol{\lambda}}} = \boldsymbol{J}^{-1}$ where approximately $\boldsymbol{J} \approx \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-1} \cdot \boldsymbol{S}$.

Eigenvalue analysis of the Fisher information matrix $\boldsymbol{J}$ can provide further insight. $\boldsymbol{J}$ is a function of the source emission rate vector $\boldsymbol{\lambda}_0$ through the matrix $\boldsymbol{\Lambda}_d = diag(\boldsymbol{t}_j^T \boldsymbol{\lambda_0})$, and also depends on the emission spectra of the sources through matrices $\boldsymbol{S}$ and $\boldsymbol{\Lambda}_d$. This suggests that the ability to estimate $\boldsymbol{\lambda_0}$ precisely depends on both the source emission rates $\boldsymbol{\lambda}_0$ and the source emission spectra $\boldsymbol{s}_i$. The analysis of estimation variance for each $\hat{\lambda}_i$ starts by considering the ellipsoid $\mathcal{E}_{\boldsymbol{J}}$: $(\boldsymbol{\lambda} - \boldsymbol{\lambda_0})^T \cdot \boldsymbol{J} \cdot (\boldsymbol{\lambda} - \boldsymbol{\lambda_0}) = 0$. The eigenvalues ξ_i and eigenvectors $\boldsymbol{\psi}_i$ of $\boldsymbol{J}$ provide the

magnitudes and directions of $\mathcal{E}_J$ axes (eigenvectors are normalized so that $\boldsymbol{\psi}_j^T \boldsymbol{\psi}_i = \delta_{ij}$, where δ_{ij} is Kronecker's delta) [Strang 1988]. The magnitude of $\boldsymbol{\xi}_i$ is inversely related to the magnitude of the corresponding axis of the covariance matrix $\boldsymbol{C}_{\hat{\boldsymbol{\lambda}}}$. Therefore, the larger the j coordinate where the ellipsoid $\mathcal{E}_J$ crosses axis j, the smaller the estimation variance for λ_i. The volume of ellipsoid $\mathcal{E}_J$ provides a metric that is inversely related to the overall covariance of $\hat{\boldsymbol{\lambda}}_{ML}$. The ellipsoid volume is analogous to the product of $\boldsymbol{J}(\boldsymbol{\lambda}_0)$ eigenvalues. The geometric mean of $\boldsymbol{J}$ eigenvalues

$$\xi_J = \sqrt[N]{\prod_{i=1}^{N} \xi_i}$$

is a metric analogous to the volume of the $\mathcal{E}_J$ ellipsoid (therefore inversely analogous to the geometric mean of the variance of $\hat{\lambda}_{ML,i}$), normalized over the number of sources N. The geometric mean of the ellipsoid $\mathcal{E}_{Poisson}$ that describes the inverse of the variance due to just Poisson noise equals $\xi_{Poisson} = \sqrt[N]{\lambda_{0,i}^{-1}}$. The effect of the mixing process is to shrink the ellipsoid $\mathcal{E}_{Poisson}$ into the ellipsoid $\mathcal{E}_J$ (therefore increase the axes of the covariance matrix $\boldsymbol{C}_{\hat{\boldsymbol{\lambda}}}$). A metric that describes the effects of the mixing process on $\boldsymbol{C}_{\hat{\boldsymbol{\lambda}}}$ and excludes the effects of Poisson noise is the ratio of $\mathcal{E}_J$ volume $\mathcal{E}_{Poisson}$ volume which equals the emission-normalized geometric mean of $\boldsymbol{J}$ eigenvalues:

$$\bar{\xi} = \sqrt[N]{\prod_{i=1}^{N} \{\xi_i \cdot \lambda_{0,i}\}} \qquad [2.4.26]$$

where $0 \leq \bar{\xi} \leq 1$. When $\bar{\xi} = 1$ spectral mixing has no effect on the estimation of source rates $\boldsymbol{\lambda}$. When $\bar{\xi} = 0$ mixing prevents the estimation of the source rates. Since the metric is normalized with respect to the number of sources N, it is possible to use $\bar{\xi}$ to compare the ease of spectral unmixing in different models that contain different number of sources N.

The metric $\bar{\xi}$ describes the overall effect of mixing on the estimation of all source rates $\boldsymbol{\lambda}_0$. It is possible though that estimating the rate of each source λ_i is affected in a different way. The effect of spectral mixing is to increase the estimator variance of source i emission $\hat{\lambda}_{ML,i}$ from $\lambda_{0,i}$ (achieved using the "ideal" sensor described earlier in this section) to $C_{\hat{\lambda},ii}$. This deteriorating effect of spectral mixing on the estimation variance $\hat{\lambda}_{ML,i}$ is described by the variance factor G_i:

$$G_i(\boldsymbol{\lambda}_0) = \frac{\lambda_{0,i}}{C_{\hat{\lambda},ii}} \qquad [2.4.27]$$

In the literature it has been proposed to analytically approximate the estimator variance $\hat{\lambda}_{ML,i}$ as $C_{\hat{\lambda},ii} = J_{ii}^{-1}$, where J_{ii} is the i-th diagonal element of $\boldsymbol{J}(\boldsymbol{\lambda}_0)$ [Neher and Neher 2004]. The resulting analytic expression for G_i is:

$$G_i(\boldsymbol{\lambda}_0) = \left(\sum_{j=1}^{N} \left\{ \frac{s_{ji}}{1 + \sum_{\substack{k=1 \\ k \neq i}}^{N} \left\{ \frac{\lambda_{0,k}}{\lambda_{0,i}} \frac{s_{jk}}{s_{ji}} \right\}} \right\} \right)^{-1} \qquad [2.4.28]$$

Since $\lambda_{0,i} \geq 0$, $s_{ij} \geq 0$ and $\|\boldsymbol{s}_j\|_1 = \sum_{j=1}^{N} s_{ji} = 1$, then $0 \leq G_i(\boldsymbol{\lambda}_0) \leq 1$. The magnitude of G_i (and the corresponding loss of estimation precision for $\hat{\lambda}_{ML,i}$) depends on the emission spectra $\boldsymbol{s}_j$ and the relative magnitudes of the source rates $\lambda_{0,k}/\lambda_{0,i}$. Eq. 2.4.28 suggests that:

- if there is a source whose emission rate is much larger compared to the rate of other sources ($\lambda_{0,i} \gg \lambda_{0,j}\ \forall i \neq j$) then $G_i \to 1$ (the rate $\lambda_{0,i}$ will be estimated with precision close to the maximum possible) and $G_j \to 0$ (the rates $\lambda_{0,j}$ of the remaining sources will be estimated with precision much worse compared to maximum possible). This suggests that spectral unmixing is not precise when source rates differ significantly.
- each source affects the estimation variance for the rates of the remaining sources. The effect of source k on the estimation of $\lambda_{0,i}$ ($i \neq k$) depends on their relative emission rates $\lambda_{0,k}/\lambda_{0,i}$ and on the emission spectra through s_{jk}/s_{ji} at each detector channel j=1,2,…,M.
- For sources i,j of similar emission spectra $s_{jk}/s_{ji} \approx 1$ which increases the variance of their estimation.
- Variance factors $G_i(\boldsymbol{\lambda}_0)$ and the metric $\bar{\xi}$ do not depend on the absolute value of $\boldsymbol{\lambda}_0$.
- When the emission spectra of sources i, j are "very different" it is still possible to get good unmixing performance even when $\lambda_{0,k}/\lambda_{0,i} \gg 1$.
- As the number of sources N increases, it is not possible to prevent spectrum cross-talk and the estimation of all source rates will be deteriorated.

The approximation $C_{\hat{\lambda},ii} \approx J_{ii}^{-1}$ is accurate only when $\boldsymbol{J}$ is near-diagonal (when its eigenvalues lie close to the unit axes). A more accurate calculation of $\hat{\lambda}_{ML,i}$ should result from a more general calculation of the diagonal elements of t$\boldsymbol{J}^{-1}$ [Sivia and Skilling 2006]:

$$C_{\hat{\lambda},ii} \approx \frac{|\boldsymbol{J}_{(i,i)}|}{|\boldsymbol{J}|} \qquad [2.4.29]$$

where $|\cdot|$ stands for the determinant of a matrix, and $\boldsymbol{J}_{(i,i)}$ is the matrix obtained from $\boldsymbol{J}$ by removing its i-th row and column [Strang 1988]. The resulting analytic expression for G_i is:

$$G_i(\boldsymbol{\lambda}_0) \approx \frac{\lambda_{0,i} \cdot |\boldsymbol{J}|}{|\boldsymbol{J}_{(i,i)}|} \qquad [2.4.30]$$

Simulation results show that many of the conclusions obtained from the analytic expression of Eq. 2.4.28 are true for the more general expression of Eq. 2.4.29 (see Section 2.4.6).

In summary, the performance of a spectral unmixing ML estimator is evaluated by quantifying its variance using two metrics: $\bar{\xi}$ describes the overall "ease of spectral unmixing" in a particular system consisting of N sources. G_i describes the deterioration of estimation variance $C_{\hat{\lambda},ii}$ for the i-th component of the system compared to the Poisson-noise limited case. Both metrics take vales in the $[0,1]$ region, where 1 stands for optimal and 0 for worse case.

Quantifying the Similarity of Sources' Emission Spectra

The previous sub-section suggests that ML spectral unmixing estimation variance depends on the similarity of the emission spectra of the sources involved. In the literature, the similarity of the emission spectra of two sources has been quantified through several metrics such as the eucledian distance between normalized spectra, spectral correlation similarity, spectral angle measure (SAM) metric $SAM = \operatorname{acos}\left(\frac{\boldsymbol{s}_1^T \boldsymbol{s}_2}{\|\boldsymbol{s}_1\| \cdot \|\boldsymbol{s}_2\|}\right)/\pi$, and the spectral information divergence [Chang 2000; Landsford et al. 2001; Schwarz and Staenz 2001 ;Garini et al. 2006; Burton et al.

2009]. However, these metrics suffer from several limitations: i) they cannot be used to estimate the ML estimator variance, ii) they do not incorporate the effect of the relative source emission rates $\lambda_{0,k}/\lambda_{0,i}$ on ML estimation variance, and iii) they cannot quantify systems containing more than N>2 sources.

Here, it is proposed to use the emission-normalized geometric mean of $\boldsymbol{J}$ eigenvalues $\bar{\xi}$ provided by Eq. 2.4.26 to describe the degree of similarity in a system consisting of an arbitrary number N of sources. $\bar{\xi}$ describes how the geometric mean of $\boldsymbol{J}$ eigenvalues are reduced due to the mixing process excluding the effects of Poisson noise. When $\bar{\xi} \to 1$ the emission spectra of sources are very "different" and the presence of each source does not affect the estimation of the other sources' rates. When $\bar{\xi} \to 0$ some or all emission rates cannot be resolved accurately either because the sources have similar emission spectra, or because their emission rates are very different.

Figure 2.4.2 shows the emission spectra of 59 fluorophores whose emission peak maximum is located within the [400, 630] nm range. The fluorophores consist of 22 fluorescent dyes (cascade blue, alexa fluor 405, alexa fluor 350, Quinine sulfate, POPOP, FITC, BodipyFL, alexa fluor 488, rhodamine green, oregongreen488, FlAsH, alexa fluor 514, alexa fluor 532, 5-CTMR, Cy3, Dylight549, alexa fluor 555, TRITC, alexa fluor 546, alexa fluor 568, ReAsH, Texas red), 13 fluorescent proteins (BFP, AmCyan, CFP, GFP, sapphire, YFP, mhoneydew, mbanana, morange, RFP, mstrawberry, mtangerine, mcherry), 6 cell tracker dyes (calcein blue, CalceinAM, CMFDA, CMR, CMTMR, Calcein red), 5 phenazines (endogenous fluorophores in *Pseudomonas aeruginosa* bacteria: PCN, PVD, 1OHPhz, PYO, PCA), 4 fluorescent beads (fluorospheres blue, yellow-green, orange, and red), 4 nucleic acid dyes (hoechst33342, DAPI, syto13, syto82), 2 endogenous fluorphores found in tissues (collagen, NADH), 2 cell component dyes (Fluoromyelin, Cellmask orange), and 1 quantum dot (E560).

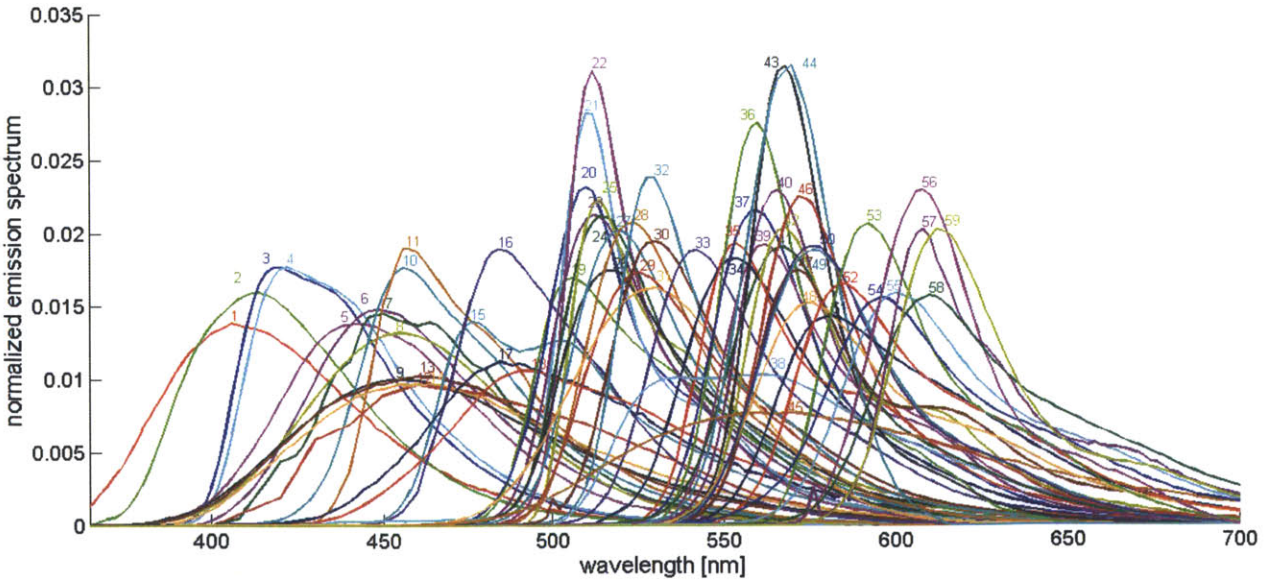


Figure 2.4.2: Emission spectra of 59 fluorophores whose emission spectrum peak lies between 400 and 625nm. 1: PCN, 2: fluorospheres blue, 3: cascade blue, 4: alexa fluor 405, 5: alexa fluor 350, 6: calcein blue, 7: collagen, 8: hoechst3334, 9: Quinine sulfate, 10: PVD, 11: DAPI, 12: BFP, 13: NADH, 14: POPOP, 15: 1OHPhz, 16: AmCyan, 17: CFP, 18: PYO, 19: GFP, 20: sapphire, 21: syto13, 22: FITC, 23: BodipyFL, 24: CalceinAM, 25: CMFDA, 26: Fluoromyelin, 27: alexa fluor 488, 28: Fluospheres yellow-green, 29: rhodamine green, 30: oregongreen488, 31:

PCA, 32: FlAsH, 33: YFP, 34: mhoneydew, 35: alexa fluor 514, 36: alexa fluor 532, 37: mbanana, 38: syto82, 39: Fluospheres orange, 40: morange, 41: 5-CTMR, 42: Cy3, 43: E560, 44: Dylight549, 45: alexa fluor 555, 46: CMR, 47: Cellmask orange, 48: CMTMR, 49: TRITC, 50: alexa fluor 546, 51: RFP, 52: mstrawberry, 53: mtangerine, 54: alexa fluor 568, 55: mcherry, 56: ReAsH, 57: Calcein red, 58: Texas red, 59: Fluospheres red.

Figure 2.4.3 shows the detected emission spectra $\mathbf{s}_i$ of the 59 fluorophores shown in Figure 2.4.2. These particular spectra $\mathbf{s}_i$ are obtained for the 16-channel instrument utilized in this thesis (Section 2.3.1) calibrated to detect emission in the [381,576] nm range (16 channels, 12.8 nm range per channel). While the emission spectra shown in Figure 2.4.2 are instrument-independent (a chemical property of fluorophores), the emission spectra of Figure 2.4.3 depend on both the instrument used and the instrument calibration.

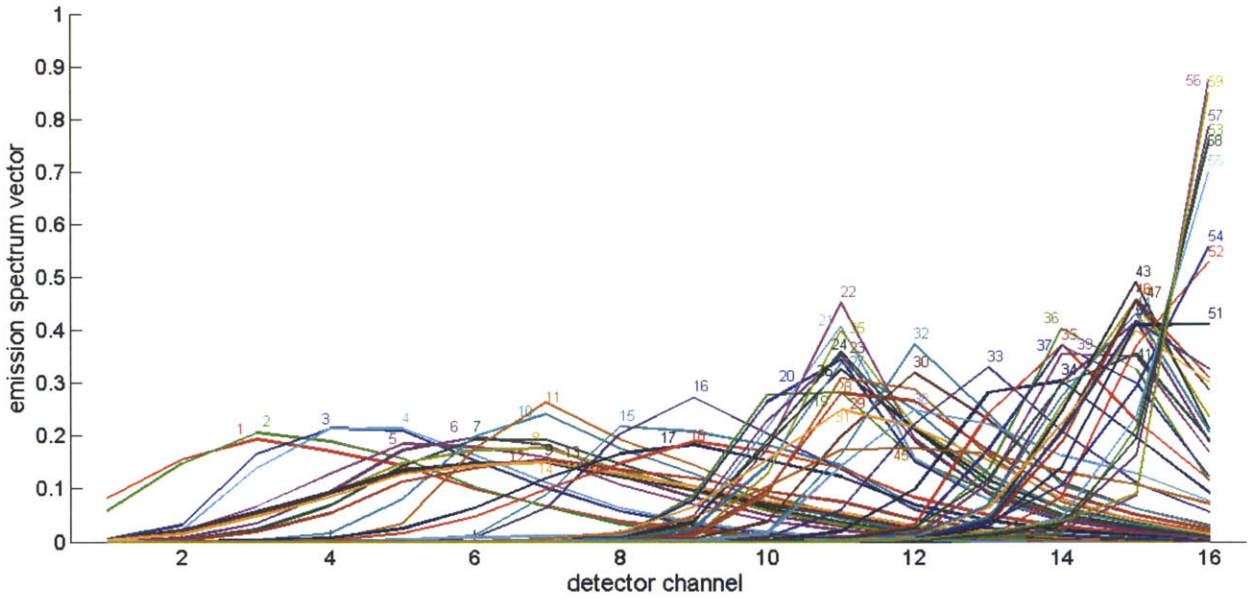


Figure 2.4.3: Emission spectra of the 59 fluorophores shown in Figure 2.4.2 as detected by the 16-channel spectral microscope used in this thesis (Section 2.3). Each channel samples approximately a 12.8nm region of the EM spectrum. Calculations are performed when the detector is calibrated so that, the 16 channels sample the [381, 586] nm EM region. 1: PCN, 2: fluorospheres blue, 3: cascade blue, 4: alexa fluor 405, 5: alexa fluor 350, 6: calcein blue, 7: collagen, 8: hoechst3334, 9: Quinine sulfate, 10: PVD, 11: DAPI, 12: BFP, 13: NADH, 14: POPOP, 15: 1OHPhz, 16: AmCyan, 17: CFP, 18: PYO, 19: GFP, 20: sapphire, 21: syto13, 22: FITC, 23: BodipyFL, 24: CalceinAM, 25: CMFDA, 26: Fluoromyelin, 27: alexa fluor 488, 28: Fluospheres yellow-green, 29: rhodamine green, 30: oregongreen488, 31: PCA, 32: FlAsH, 33: YFP, 34: mhoneydew, 35: alexa fluor 514, 36: alexa fluor 532, 37: mbanana, 38: syto82, 39: Fluospheres orange, 40: morange, 41: 5-CTMR, 42: Cy3, 43: E560, 44: Dylight549, 45: alexa fluor 555, 46: CMR, 47: Cellmask orange, 48: CMTMR, 49: TRITC, 50: alexa fluor 546, 51: RFP, 52: mstrawberry, 53: mtangerine, 54: alexa fluor 568, 55: mcherry, 56: ReAsH, 57: Calcein red, 58: Texas red, 59: Fluospheres red.

Pair-wise spectral similarity in a system of N sources can be described using a distance matrix $\boldsymbol{\Xi}$ whose (i,j) element contains the metric $\bar{\xi}$ of a N=2 system that consists of sources i and j (which is denoted as $\bar{\xi}\{i,j\}$). Since $\bar{\xi}$ depends on the emission rate ratios $\lambda_{0,i}/\lambda_{0,j}$ in the simplest case it is assumed that $\lambda_{0,i} = \lambda_{0,j}$.

$$\boldsymbol{\Xi}(i,j) = \bar{\xi}\{i,j\}$$

Figure 2.4.4 displays the pairwise distance matrix $\boldsymbol{\Xi}$ for the 59 fluorophores shown in Figure 2.4.3 when their emission rates are equal $\lambda_{0,i} = \lambda_{0,j}$. The distance matrix depends on the

instrument and the instrument calibration. Instruments of different channel number M, channel region width, and different emission optics would result in different $\bar{\xi}\{i,j\}$.

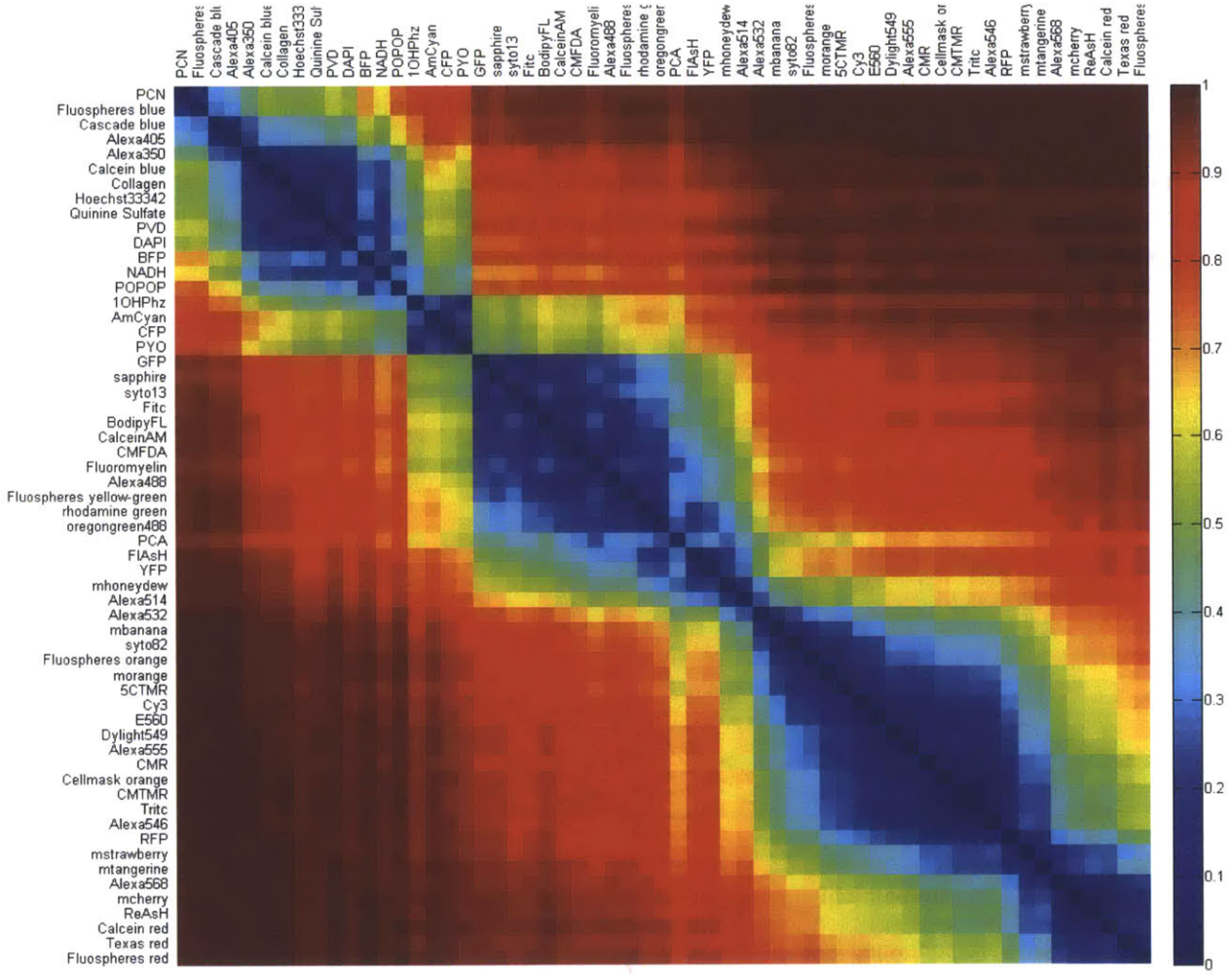


Figure 2.4.4: Pairwise distance matrix Ξ between the emission spectra of the 59 fluorophores of Figure 2.4.2 based on the 16-channel detected emission spectra shown in Figure 2.4.3. The distance is defined as the emission-normalized geometric mean $\bar{\xi}$ of J eigenvalues provided by Eq. 2.4.26 for the case when fluorophores have equal emission rate.

The pattern of the pairwise distance matrix Ξ shown in Figure 2.4.4 suggests that these 59 fluorescent dyes can be classified into seven clusters that contain sources of similar emission spectra: the "far blue" cluster (PCN, fluorospheres blue, cascade blue, alexa fluor 405), the "blue" cluster (alexa fluor 350, calcein blue, collagen, hoechst3334, Quinine sulfate, PVD, DAPI, BFP, NADH, POPOP), the "cyan" cluster (1OHPhz, AmCyan, CFP, PYO), the "green" cluster (GFP, sapphire, syto13, FITC, BodipyFL, CalceinAM, CMFDA, Fluoromyelin, alexa fluor 488, Fluospheres yellow-green, rhodamine green, oregongreen488), the "yellow" cluster (PCA, FlAsH, YFP, mhoneydew, alexa fluor 514), the "orange" cluster (alexa fluor 532, mbanana, syto82, Fluospheres orange, morange, 5-CTMR, Cy3, E560, Dylight549, alexa fluor 555, CMR, Cellmask orange, CMTMR, TRITC, alexa fluor 546), the "orange-red" cluster (RFP, mstrawberry, mtangerine), and the "red cluster" (alexa fluor 568, mcherry, ReAsH, Calcein red, Texas red, Fluospheres red). Generally, the dyes of each cluster are quite similar to the dyes of

the same cluster ($\bar{\xi} < 0.3$), relatively similar to dyes of neighboring clusters ($0.3 < \bar{\xi} < 0.6$) and different than dyes of other clusters ($\bar{\xi} > 0.6$). Dye clustering can be utilized to design staining protocols for high content imaging experiments (Section 2.4.8).

Figure 2.4.5 compares the $\bar{\xi}$ metric and the published SAM metric applied in quantifying the difference between the emission spectra of all possible combinations of the 59 fluorophores shown in Figure 2.4.3. The $\bar{\xi}$ metric is calculated assuming that the emission rates of the two fluorophores are equal. Results show that the SAM metric correlates very well to the $\bar{\xi}$ metric for two fluorophores of equal emission rates. Nevertheless, $\bar{\xi}$ is amore generic metric, as it can describe cases where there is an arbitrary number N of sources of arbitrary emission rates.

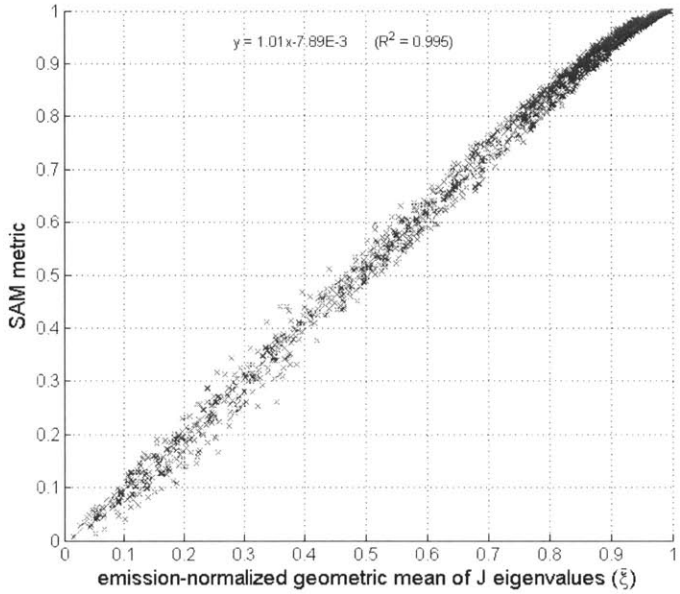


Figure 2.4.5: Relationship between the $\bar{\xi}$ metric and the spectral angle measure (SAM) metric for quantifying the similarity of the emission spectra of two fluorophores. Metrics are calculated for all pairs between the 59 fluorophores displayed in Figure 2.4.4. Both metrics are calculated based on the detected spectra in the 16-channel spectral multi-photon microscope utilized in this thesis (Section 2.3). The $\bar{\xi}$ metric is calculated assuming the two fluorophores have equal emission rates.

2.4.6 Simulation Results: No Model Uncertainty

This section discusses the performance of the three ML spectral unmixing algorithms in the case when there is no model uncertainty. It is assumed that noise characteristics and source properties (number N, emission spectra vectors $\boldsymbol{s}_i$) are known. This section presents simulation results that describe how estimation variance, estimation bias, and computation cost depend on experimental parameters (choice of sources, emission rates of sources, signal level), imaging parameters (channel number), and spectral unmixing algorithm utilized.

Estimation Variance

As described in Section 2.4.5, under appropriate assumptions (large $\boldsymbol{\lambda}_0$, no model uncertainty) the ML estimation error $\tilde{\boldsymbol{\lambda}}(\boldsymbol{y}) = \hat{\boldsymbol{\lambda}}_{ML}(\boldsymbol{y}) - \boldsymbol{\lambda}_0$ is a Gaussian vector $\tilde{\boldsymbol{\lambda}} \sim N(\boldsymbol{0}, \boldsymbol{J}^{-1})$ of zero mean and covariance matrix $\boldsymbol{J}^{-1}$. Estimation variance is quantified using two metrics. First, the ease of spectral unmixing of a signal generated by mixing N sources of overlapping spectra is described by the emission-normalized geometric mean of $\boldsymbol{J}$ eigenvalues $\bar{\xi}$ (Eq. 2.4.26). Second, the

estimation variance $C_{\hat{\lambda},ii}$ for the estimate $\hat{\lambda}_{ML,i}$ of $\lambda_{0,i}$ is quantified by the variance factor $G_i(\boldsymbol{\lambda}_0) = \lambda_{0,i}/C_{\hat{\lambda},ii}$ (Eq. 2.4.27), which describes how much estimation variance for component i increases due to mixing, compared to the best-case scenario where variance is due to the Poisson-noise ($\lambda_{0,i}$). The Fisher information matrix $\boldsymbol{J} = \boldsymbol{S}^T \cdot \boldsymbol{\Lambda}_{\boldsymbol{d}}^{-1} \cdot \boldsymbol{S}$ is key in calculating both $\bar{\xi}$ and G_i.

The properties of the Fisher information matrix $\boldsymbol{J}$ depend on the sources (number N, emission spectra $\boldsymbol{s}_i$, emission rates $\boldsymbol{\lambda}_0$) that contribute to the mixed signal. Given a set of sources ($\boldsymbol{s}_i$) and their emission rates ($\boldsymbol{\lambda}_0$) Eq. 2.4.29provides a theoretic estimate of the estimator variance $C_{\hat{\lambda},ii}$ for the i-th source, and Eq. 2.4.30describes estimator variance through G_i. Theoretic estimates of $\bar{\xi}$ and G_i (Eq. 2.4.26, 2.4.30) depend indirectly on the instrument configuration and alignment through the emission spectra vectors $\boldsymbol{s}_i$, and do not depend on the signal level ($Y = \|\boldsymbol{\lambda}_0\|_1 = \sum \lambda_{0,i}$), just the emission rate ratios $\lambda_{0,i}/\lambda_{0,j}$.

Figure 2.4.6 compare the theoretic calculations of the variance factors G_i (Eq. 2.4.30) with simulation results $G_i^{SIM} = \lambda_{0,i}/var(\hat{\lambda}_{ML,i}(\boldsymbol{y}))$. Each subplot shows results for 50 cases, where each case corresponds to a mixture generated by randomly picking 2≤N≤6 sources from the 59 sources shown in Figure 2.4.3, where the randomly-generated emission vector has a specific signal level Y (units: photons/pixel) and maximum emission rate ratio $R = max(\lambda_{0,i}/\lambda_{0,j})$. For each mixing case (a particular set of sources $\boldsymbol{S}$ and emission rates $\boldsymbol{\lambda}_0$), $var(\hat{\lambda}_{ML,i}(\boldsymbol{y}))$ is calculated from the ML solutions $\hat{\lambda}_{ML,i}$ of 64000 randomly simulated measurements $\boldsymbol{y}$ generated for that particular case ($\boldsymbol{S}$ and $\boldsymbol{\lambda}_0$). ML estimates are calculated using the EM algorithm described in Section 2.4.3.

Figure 2.4.6 results suggest that theoretically predicted variance factors G_i become larger (closer to 1) as the metric $\bar{\xi}$ increases. For each one of the 50 cases (source sets) shown in Figure 2.4.6 subplots, there are N points (G_i,G_i^{SIM}) that correspond to the theoretically predicted and ML simulated variance factors of the N sources of that case. The N points of each case are highlighted with a marker that corresponds to the theoretical $\bar{\xi}$ for this case (Eq. 2.4.26).

Results suggest that the validity of theoretic G_i predictions (Eq. 2.4.30) depends strongly on the G_i itself, the signal level Y, and the maximum emission rate ratio R. When signal level Y is large and the max ratio R approaches 1, then there is very good agreement between G_i and G_i^{SIM} over a wider region of G_i values. As the signal level Y decreases, G_i predicts G_i^{SIM} well over a smaller range of R values away from 1 and over a smaller region of G_i away from 1. For example, when $Y = 250$ photons/pixel, G_i approaches G_i^{SIM} for values as small as $G_i = 0.05$. When $Y = 50$ photons/pixel, G_i^{SIM} deviates significantly from G_i when $G_i < 0.3$. Increasing R makes the prediction less reliable. The deviation G_i-G_i^{SIM} becomes significant for sources of small G_i value, where G_i under-estimates G_i^{SIM} (simulated estimation variance is much less than the one predicted). Results also show that the validity of theoretic G_i predictions depends weakly on $\bar{\xi}$, since deviation G_i-G_i^{SIM} becomes larger in cases of small $\bar{\xi}$ (denoted by the "x" mark in Figure 2.4.6).

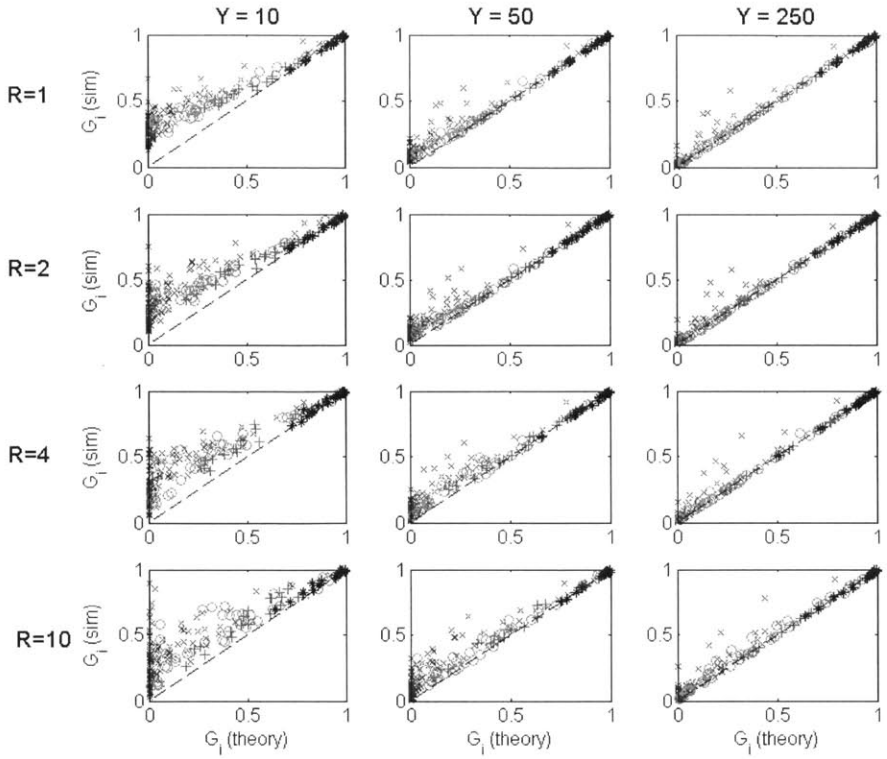


Figure 2.4.6: Comparison of the theoretic variance factors $\boldsymbol{G_i}$ calculated by Eq. 2.4.30 with simulated $\boldsymbol{G_i^{SIM}}$ for three values of signal level $\boldsymbol{Y = \sum \lambda_{0,i}}$ (columns) and four values of maximum emission rate ratio $\boldsymbol{R = \max(\lambda_{0,i}/\lambda_{0,j})}$ (rows). $\boldsymbol{G_i^{SIM}}$ is calculated based on 64000 random samples per case. Each plot shows 50 cases generated by randomly picking $\boldsymbol{2 \le N \le 6}$ sources from the 59 sources shown in Figure 2.4.3 and a random emission vector $\boldsymbol{\lambda_0}$ based on the specific combination of signal level $\boldsymbol{Y}$ and max emission ratio $\boldsymbol{R}$. For each case there are N points ($\boldsymbol{G_i}$, $\boldsymbol{G_i^{SIM}}$) that correspond to the predicted and actual ML estimation variance of the N sources. The N points of each case are highlighted with a different marker depending on the calculated $\bar{\xi}$ (Eq. 2.4.26): x ($\boldsymbol{0 \le \bar{\xi} \le 0.25}$), o ($\boldsymbol{0.25 \le \bar{\xi} \le 0.5}$), + ($\boldsymbol{0.5 \le \bar{\xi} \le 0.75}$), and * ($\boldsymbol{0.75 \le \bar{\xi} \le 1}$). ML estimates are calculated using the EM algorithm.

The maximum emission rate ratio

$$R = max(\lambda_{0,i}/\lambda_{0,j})$$

describes how uniform are the emission rates present in the pixel. It is affected by the nature of the sample and the experimental staining protocol. The effect of R in estimation variance is further demonstrated in Figure 2.4.7, which shows how the distance matrix $\boldsymbol{\Xi}$ between the 59 fluorophores of Figure 2.4.3 varies as a function of the emission rate ratio λ_1/λ_2 (rows: source 1, columns: source 2).

Figure 2.4.7 shows that the magnitude of $\boldsymbol{\Xi}$ elements $\bar{\xi}\{i,j\}$ depends on the emission rate ratio λ_1/λ_2. While $\boldsymbol{\Xi}(\lambda_1/\lambda_2 = 1)$ is symmetric, $\boldsymbol{\Xi}(\lambda_1/\lambda_2 \neq 1)$ is not symmetric anymore: due to the non-symmetric shape of fluorophore emission spectra (tail at large wavelengths), increasing the emission rate λ_i of source i affects fluorophores whose emission peaks are located at longer wavelengths more compared to fluorophores whose emission peaks are located in shorter wavelengths. Figure 2.4.7 also shows that $\boldsymbol{\Xi}(\lambda_1/\lambda_2 = 5)$ is quite similar to $\boldsymbol{\Xi}(\lambda_1/\lambda_2 = 1)$ in the sense that the pairwise metric $\bar{\xi}\{i,j\}$ between fluorophores that belong to neighboring dye clusters is not reduced significantly ($\bar{\xi}\{i,j\}$ is reduced by up to 0.25). However $\boldsymbol{\Xi}(\lambda_1/\lambda_2 = 20)$ is significantly different than $\boldsymbol{\Xi}(\lambda_1/\lambda_2 = 1)$ in the sense that the metric $\bar{\xi}\{i,j\}$ between fluorophores

that belong to neighboring dye clusters takes small values (less than 0.3). This figure suggests that when $\lambda_1/\lambda_2 \neq 1$ the elements of the Ξ matrix get smaller, therefore spectral unmixing becomes harder. This does not mean that the estimation variance of all source rates increases.

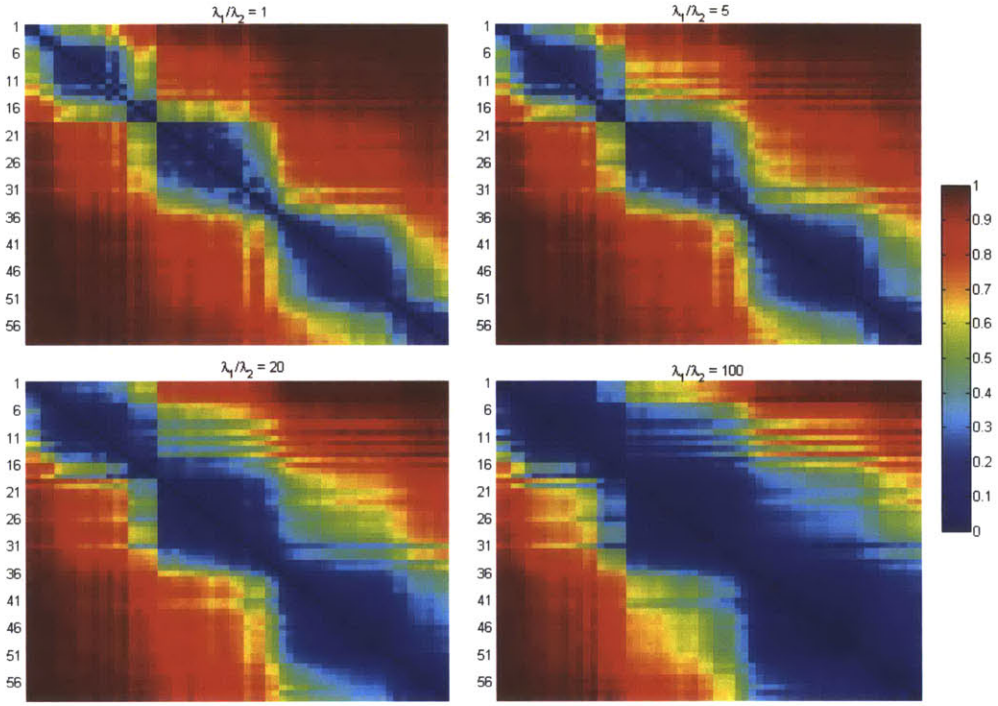


Figure 2.4.7: The effect of the emission rate ratio λ_1/λ_2 in the pairwise distance matrix Ξ shown in Figure 2.4.4. The distance matrix is calculated based on the emission-normalized geometric mean of the eigenvalues $\bar{\xi}$ of the Fisher Information matrix $\boldsymbol{J}$ (Eq. 2.4.26). Source 1 corresponds to rows, source 2 corresponds to columns. In each case, the sensor is calibrated to detect emission in the [381, 586] nm EM region.

Figure 2.4.8 shows that in a mixture of N sources, as the rate λ_i of one source i increases, then the variance factor G_i of this element increases and the variance factors G_j $(j \neq i)$ of the remaining sources decrease. Overall for a given set of fluorophores, the $\bar{\xi}$ metric decreases as the maximum emission rate ratio R deviates from 1,

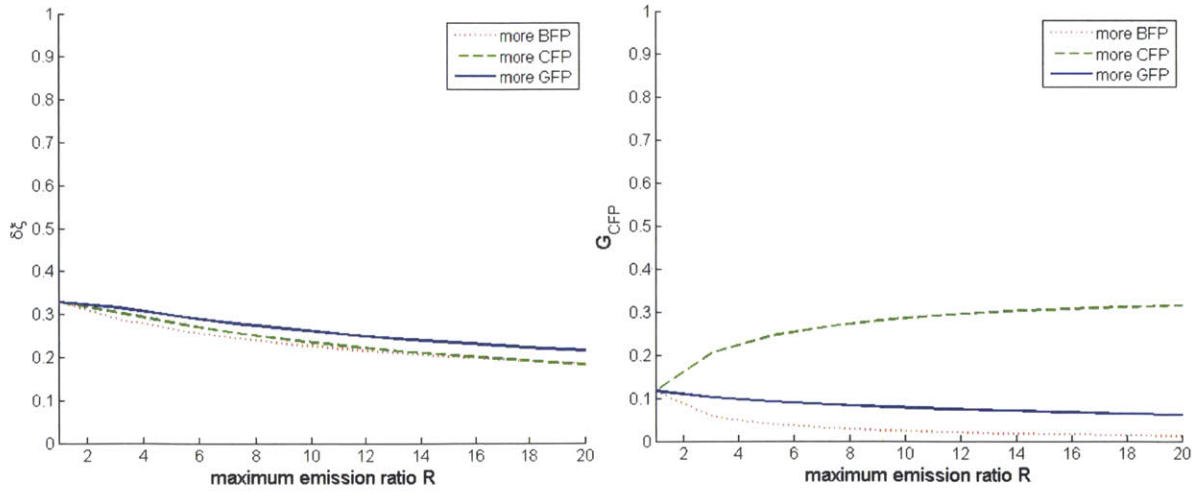


Figure 2.4.8: The spectral unmixing performance in the six fluorescent protein set (BFP, CFP, GFP, YFP, morange, mcherry, see Figure 2.4.18) deteriorates as the maximum emission rate ratio $\boldsymbol{R}$ deviates away from 1. **Left:** Value of

the $\bar{\xi}$ metric when the emission of one of the sources is R times the common emission rate of the five remaining sources. **Right:** Variance factor $\boldsymbol{G_{CFP}}$ for CFP when the emission rate of one of the sources is R times the common emission rate of the five remaining sources.

Estimation Bias

Under the assumptions of large $\boldsymbol{\lambda_0}$ and no model uncertainty, theoretical results suggest that ML spectral unmixing is unbiased, i.e. $E[\tilde{\boldsymbol{\lambda}}(\boldsymbol{y})] = \mathbf{0}$ (Section 2.4.5). However, simulation results show that even in the absence of modeling errors ($\boldsymbol{S} = \boldsymbol{S_0}$) estimation bias can be non-zero. In this section, bias is described by the normalized bias factor, which is defined as:

$$B_i = \frac{E[\hat{\lambda}_{ML,i} - \lambda_{0,i}]}{\lambda_{0,i}} \qquad [2.4.31]$$

and is the non-dimensional ratio of estimation bias $E[\hat{\lambda}_{ML,i} - \lambda_{0,i}]$ to the actual value $\lambda_{0,i}$.

Figure 2.4.9 provides simulation results B_i^{SIM} (where the expectation $E[\hat{\lambda}_{ML,i} - \lambda_{0,i}]$ is calculated based on ML estimates obtained by the EM algorithm) for the same cases (source sets, signal level Y, maximum emission rate R) and data as the ones shown in Figure 2.4.6. For each one of the N components of each case, its simulated B_i^{SIM} is plotted versus its theoretic factor G_i using a marker that corresponds to the $\bar{\xi}$ value of that case.

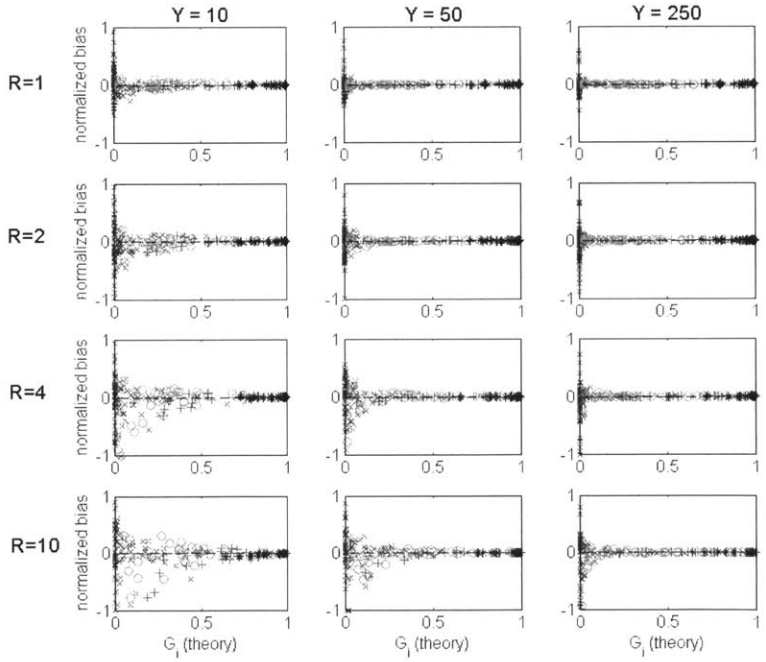


Figure 2.4.9: Simulated normalized bias $\boldsymbol{B_i^{SIM} = E[\hat{\lambda}_{0,i}(y) - \lambda_{0,i}]/\lambda_{0,i}}$ for three values of signal level $\boldsymbol{Y = \sum \lambda_{0,i}}$ (columns) and four values of maximum emission rate ratio $\boldsymbol{R = \max(\lambda_{0,i}/\lambda_{0,j})}$ (rows).These results are obtained from identical simulations as the results for $\boldsymbol{G_i}$ and $\boldsymbol{G_i^{SIM}}$ shown in Figure 2.4.6, calculated based on 64000 random samples per case. Each plot shows 50 cases generated by randomly picking $\boldsymbol{2 \le N \le 6}$ sources from the 59 sources shown in Figure 2.4.3 and a random emission vector $\boldsymbol{\lambda_0}$ based on the specific combination of signal level $\boldsymbol{Y}$ and max emission ratio $\boldsymbol{R}$. For each case there are N points $(\boldsymbol{G_i}, \boldsymbol{B_i^{SIM}})$ that correspond to the predicted and actual ML estimation variance of the N sources. The N points of each case are highlighted with a different marker depending on the calculated $\bar{\xi}$ (Eq. 2.4.26): x ($\boldsymbol{0 \le \bar{\xi} \le 0.25}$), o ($\boldsymbol{0.25 \le \bar{\xi} \le 0.5}$), + ($\boldsymbol{0.5 \le \bar{\xi} \le 0.75}$), and * ($\boldsymbol{0.75 \le \bar{\xi} \le 1}$). ML estimates are calculated using the EM algorithm.

Simulation results show that B_i^{SIM} deviates from 0 as a function of signal level Y, maximum emission ratio R, and the parameters G_i, and $\bar{\xi}$ (describe the system to be unmixed). The pattern of $B_i^{SIM}(Y, R, G_i, \bar{\xi})$ deviation from 0 shares common features with the way $G_i^{SIM}(Y, R, G_i, \bar{\xi})$ deviates from G_i. B_i^{SIM} deviates significantly from 0 over a small region near $G_i = 0$ when Y is large ($Y = 250$ photons/pixel, even $Y = 50$ photons/pixel when R is small) and $R \to 1$. As signal level Y decreases, B_i^{SIM} deviates from 0 over a wider range of ratio R and over a wider region of G_i around 0. B_i^{SIM} deviates from 0 more for sources of small G_i, and this sensitivity becomes more significant for systems of smaller $\bar{\xi}$.

Comparison of ML unmixing algorithms

The estimation variance and bias results shown in Figure 2.4.6 and Figure 2.4.9 have been calculated using ML estimates obtained using the EM algorithm based on the Poisson mixing model (Section 2.4.3). Figure 2.4.10 and Figure 2.4.11 show how results for $G_i^{SIM}(G_i)$ and $B_i^{SIM}(G_i)$ deviate when ML estimates $\hat{\lambda}_{ML,i}$ are calculated using the Gaussian-common noise (GCN) algorithm and the Gaussian-variable noise (GVN) unmixing algorithms (Section 2.4.4) using data from the identical unmixing cases.

Figure 2.4.10 and Figure 2.4.11 show that the performance of the two Gaussian mixing-based ML estimation algorithms (GCN, GVN) is not significantly worse than the performance of the EM algorithm. The simplest algorithm (GCN) has slightly worse performance than the other two more computationally-intensive algorithms. For sources of large G_i, $G_i^{SIM}(\mathrm{GCN})$ is slightly less than G_i indicating that GCN results in 10-20% larger estimation variance compared to the theoretic calculation.

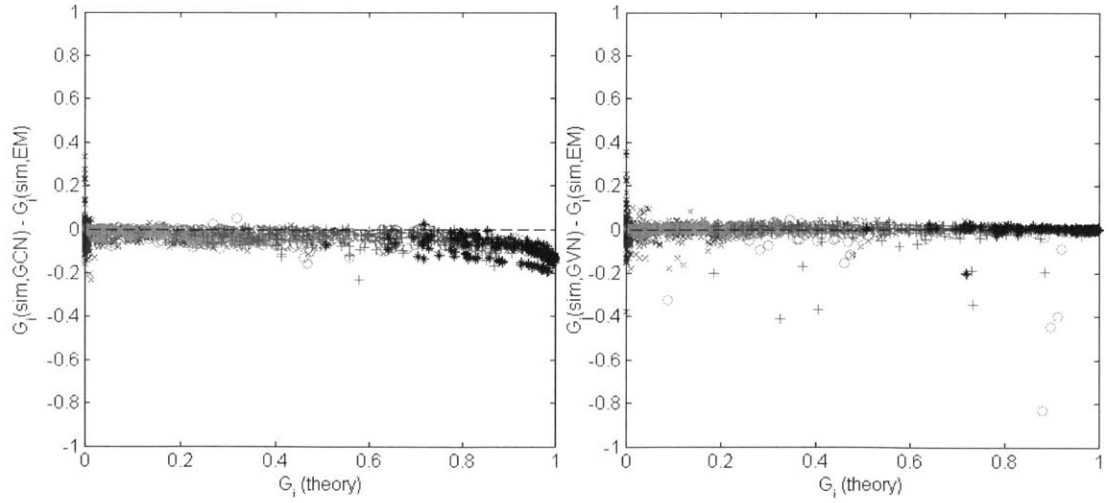


Figure 2.4.10: Comparison of the estimation variance $\boldsymbol{G_i^{SIM}}$obtained by Gauss unmixing ML estimators versus $\boldsymbol{G_i^{SIM}}$ obtained by the EM unmixing ML estimator. **Left**: Difference $\boldsymbol{G_i^{SIM}(GCN) - G_i^{SIM}(EM)}$ between simulation results calculated using Gaussian common-noise (GCN) or the EM algorithm. **Right**: Difference $\boldsymbol{G_i^{SIM}(GVN) - G_i^{SIM}(EM)}$ between simulation results calculated using Gaussian variable-noise (GVN) or the EM algorithm. Simulations correspond to identical cases as the ones shown in Figure 2.4.6. In all cases, results are highlighted with a different marker depending on the $\bar{\xi}$ of each case: x ($\mathbf{0 \le \bar{\xi} \le 0.25}$), o ($\mathbf{0.25 \le \bar{\xi} \le 0.5}$), + ($\mathbf{0.5 \le \bar{\xi} \le 0.75}$), and * ($\mathbf{0.75 \le \bar{\xi} \le 1}$)

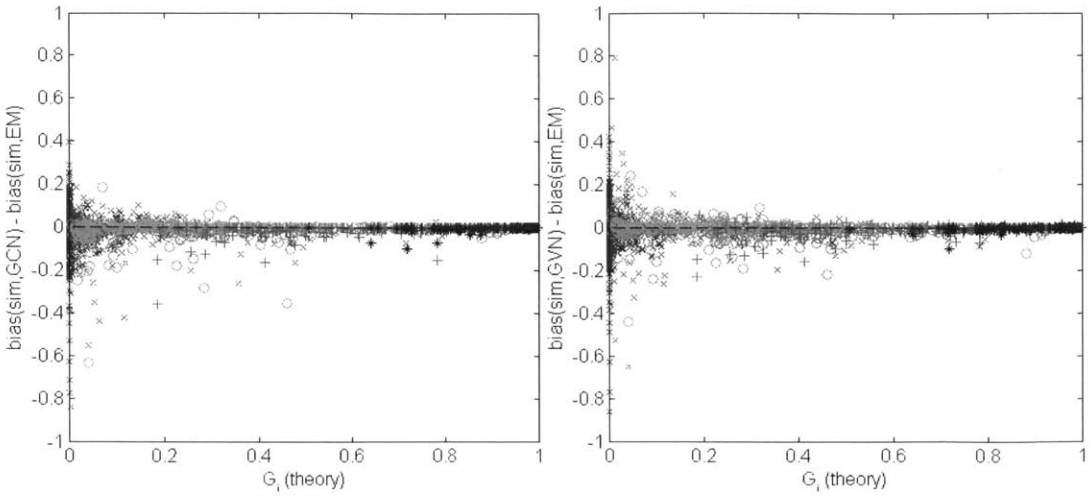


Figure 2.4.11: Comparison of the estimation normalized bias $\boldsymbol{B_i^{SIM}}$ obtained by Gauss unmixing ML estimators versus $\boldsymbol{G_i^{SIM}}$ obtained by the EM unmixing ML estimator. **Left**: Difference $\boldsymbol{B_i^{SIM}(GCN) - B_i^{SIM}(EM)}$ between simulation results calculated by Gaussian common-noise (GCN) or the EM algorithm. **Right**: Difference $\boldsymbol{B_i^{SIM}(GVN) - B_i^{SIM}(EM)}$ between simulation results calculated by Gaussian variable-noise (GVN) or the EM algorithm. Simulations correspond to identical cases as the ones shown in Figure 2.4.9. In all cases, results are highlighted with a different marker depending on the $\bar{\xi}$ of each case: x ($\mathbf{0} \leq \bar{\xi} \leq \mathbf{0.25}$), o ($\mathbf{0.25} \leq \bar{\xi} \leq \mathbf{0.5}$), + ($\mathbf{0.5} \leq \bar{\xi} \leq \mathbf{0.75}$), and * ($\mathbf{0.75} \leq \bar{\xi} \leq \mathbf{1}$).

Unmixing algorithms need to be compared not just in terms of the estimation fidelity they provide, but also in terms of the computational cost they require. Figure 2.4.12 shows the required computation time τ per pixel per source for each one of the ML unxming algorithms (EM, GCN, GVN) considered in this thesis. Results show that the computation power required by each algorithm is significantly different. GCN requires 1-2 orders of magnitude less computation compared to the GVN algorithm, which requires at most 1 order of magnitude less computation compared to the EM algorithm. The computation cost τ of the EM and GCN algorithms depends significantly on the $\bar{\xi}$ of the system, while the computation time for GVN is less sensitive on $\bar{\xi}$. As $\bar{\xi} \to 1$ the required computation cost of the EM algorithm is a bit less than one order of magnitude smaller compared to the case $\bar{\xi} \to 0$, and displays less variance. The required computation cost of GCN decreases as $\bar{\xi}$ increases because fewer iterations are required in NNLS. At $\bar{\xi} > 0.3$ $\tau(\bar{\xi})$ the computation cost of GCN shows a biphasic shape. The lower branch corresponds to cases where unconstrained solution (SVD) is enough, while the top branch corresponds to cases that require constrained solution (NNLS). Computations are implemented in MATLAB R2012b (Mathworks; Natick MA) running in a quad-core CPU (Intel i7-2600 3.4 GHz) with 8GB RAM under the Ubuntu 12.04 (kernel 3.2.0-43) operating system.

Comparing the estimation and computation performance of the three ML unmixing algorithms, the previous figures show that the EM algorithm offers the best ML estimation performance (lowest variance and bias) compared to the Gaussian-based ML unmixing algorithms (GCN, GVN), however at significantly higher computation cost. The improvement in ML estimation provided by the EM algorithm seems not significant enough to justify the increased computation cost. In hard unmixing cases ($\bar{\xi} \to 0$) EM is approximately 50 times shower than GCN and 5 times slower than GVN. In easy unmixing cases ($\bar{\xi} \to 1$) EM is 20-200 times slower than GCN, 2 times slower than GVN without providing significantly improved estimation performance in terms

of estimation variance and bias. None of the algorithms is able to improve the estimation performance in hard unmixing cases.

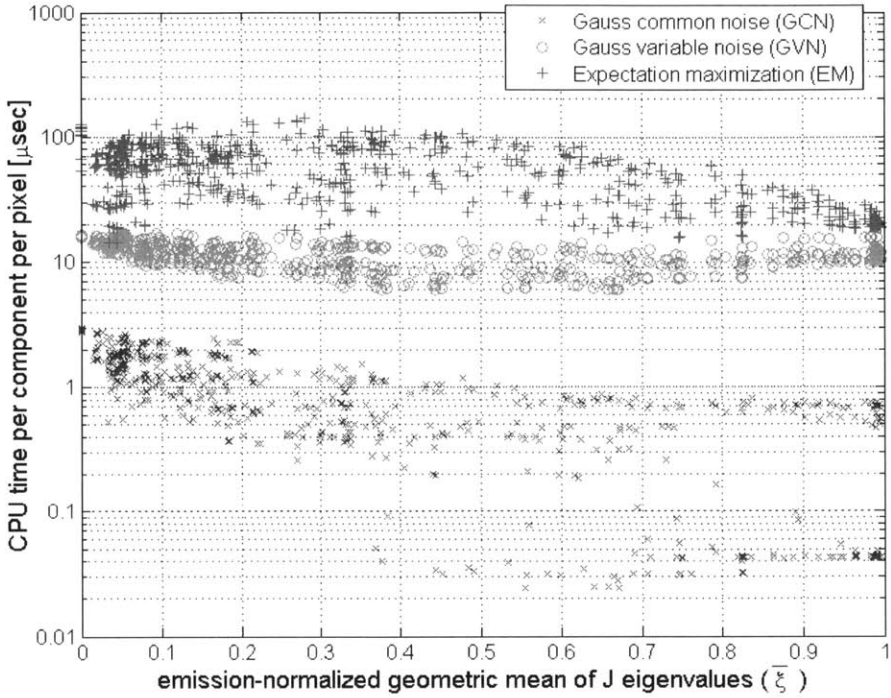


Figure 2.4.12: Mean CPU time per pixel per component required by each ML unmixing algorithm as a function of the emission-normalized geometric mean of **J** eigenvalues $\bar{\xi}$ for the particular case (set of fluorophores, emission ratio).

The effect of detector channel number M

The vast majority of research or commercial spectral fluorescent microscopes, including the one utilized in this study, incorporate detectors that contain M=8 to M=32 channels of $\delta\lambda = 10 - 30$ nm detection range per channel, and sample a $\Delta\Lambda = 200 - 300$ nm region of the EM spectrum. [Haraguchi et al. 2002; Tsurui et al. 2000; Landsford et al. 2001; Buehler et al. 2005; Tsupryk et al. 2008], see Sections 2.2.2 and 2.3.3. There are a few instruments that incorporate EM-CCDs and provide M=80 to 512 channels that detect emission over a 350 nm region ($\delta\lambda = 0.7 - 4.4$ nm per channel) [Sinclair et al. 2006; Im et al. 2010].

The main motivation for increasing channel number M is to be able to resolve the signal of more sources N, either by sampling a larger region of the EM spectrum, or by sampling a given region with finer resolution in order to resolve sources whose emission spectra are quite similar.

Figure 2.4.13 shows how the pairwise distance matrix $\boldsymbol{\Xi}$ between the 59 fluorophores shown in Figure 2.4.3 varies as the source emission is detected by sensors that contain 4, 8, 16 or 32 channels. The case of 16 channels corresponds to the results shown in Figure 2.4.4. For fairness of comparison all sensors detect emission in the (381, 586) nm region of the EM spectrum. The distance matrix is calculated assuming the two sources have equal emission λ_i. ($R = 1$). Results show that increasing the number of channels from 4 to 8 results in significant increase in pairwise $\bar{\xi}$ metrics particularly between fluorophores that belong to neighboring source clusters, leading to improved spectral unmixing performance. Increasing channel number from 8 to 16 results in slight increase in pairwise $\bar{\xi}$, while further increase to 32

channels results in negligible increase in pairwise $\bar{\xi}$. However, increasing the number of channels from 4 to 32 has very limited effect on the ability to resolve pairs of components that belong to the same cluster of the 16-channel distance matrix Ξ.

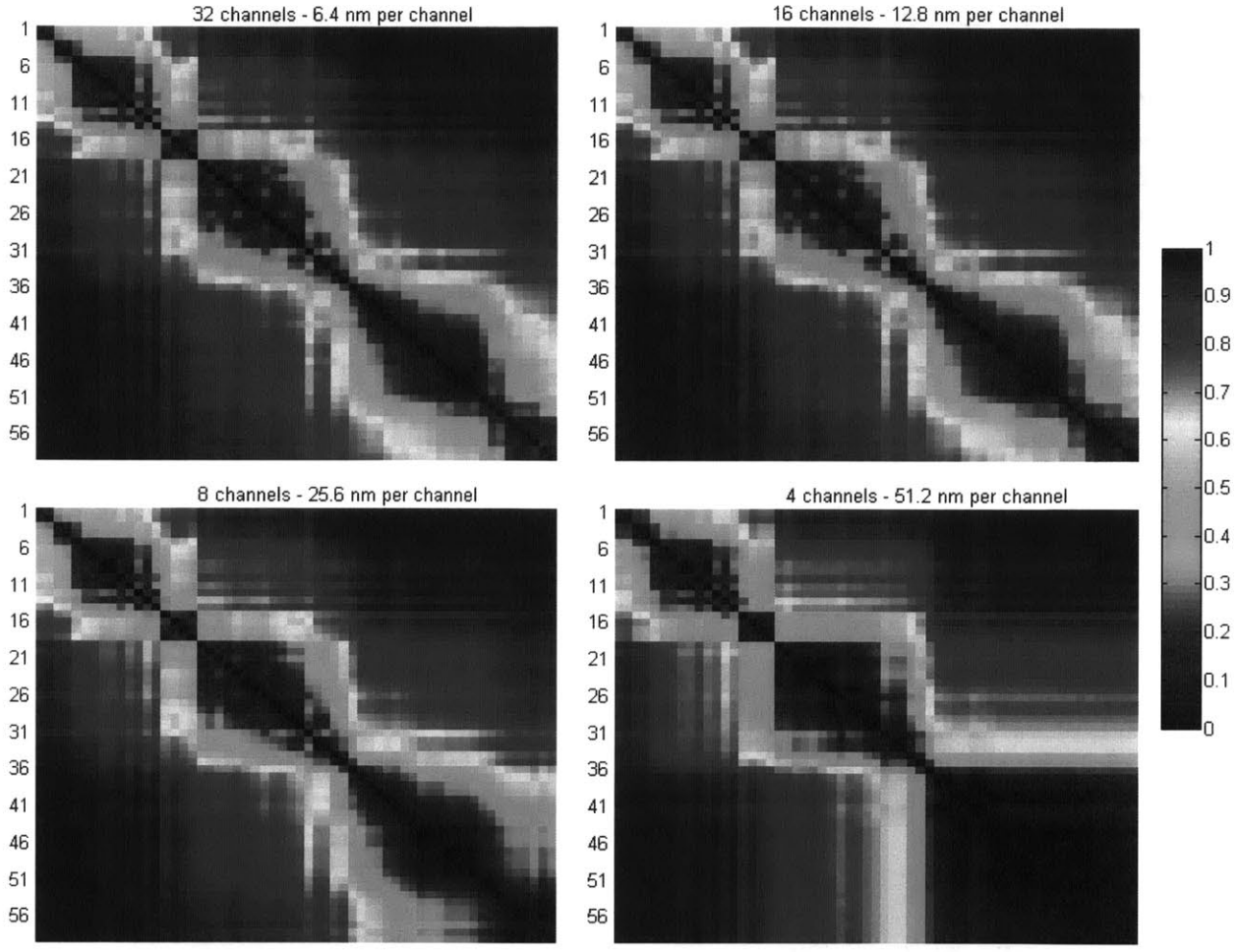


Figure 2.4.13: The effect of the channel number M in the pairwise distance matrix Ξ shown in Figure 2.4.4. The distance matrix is calculated based on the emission-normalized geometric mean of the eigenvalues $\bar{\xi}$ of the Fisher Information matrix $\boldsymbol{J}$ when the two sources have equal emission $\boldsymbol{\lambda}_{\iota}$(Eq. 2.4.26). In each case, the sensor detects emission in the (381,586) nm EM region.

Increasing channel number has a positive effect on accelerate unmixing calculations, as it increases the emission-normalized geometric mean of $\bar{\xi}$. On the other hand, increasing channel number M deteriorates performance by decreasing the signal magnitude per channel, and by increasing memory requirements and array sizes. It seems therefore meaningful to minimize the number of channels utilized in unmixing calculations without reducing $\bar{\xi}$ significantly.

It was shown that as the difference between the source emission rates increase, spectral unmixing becomes harder ($\bar{\xi}$ and G_i decrease). It is of interest to evaluate if signal acquisition by more channels could limit the deterioration of estimation performance that takes place as emission rates differ.

Three cases: easy, hard, challenging

The first case (easy) considers a set of N=3 components, consisting of a cell-tracker dye (CMTMR) and two dyes (Alexa Fluor 350 and 488) used often as secondary antibody conjugates. The spectral signatures of the three sources detected by the spectral microscope described in Section 2.3 are very well separated (Figure 2.4.14b). The distance matrix $\boldsymbol{\Xi}$ components are $\bar{\xi}\{alexa350, alexa488\} = 0.9$, $\bar{\xi}\{alexa350, CMTMR\} = 0.99$, and $\bar{\xi}\{alexa488, CMTMR\} = 0.8$, suggesting components are significantly distinct of each other (Figure 2.4.14c). The $\bar{\xi} = 0.84$ metric of the whole system (common emission rate) suggests that the three components could be estimated without mixing causing major loss of precision. The variance factors are $G_{alexa350} = 0.9$, $G_{alexa488} = 0.75$, and $G_{CMTMR} = 0.84$ (Figure 2.4.14d). This means for example that the estimation variance of alexa488 emission rate will be 33% (1/0.75=1.33) larger compared to the best possible case (Poisson noise-limited).

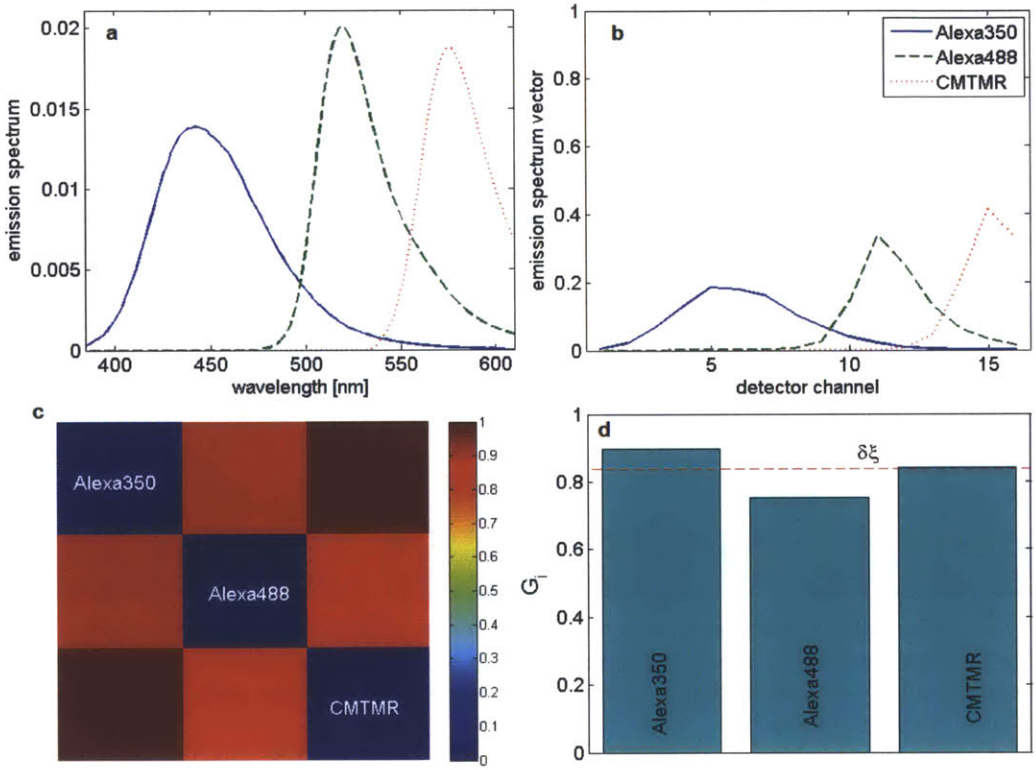


Figure 2.4.14: Theoretic analysis of spectral unmixing estimation variance from data resulting from mixing N=3 fluorophores applied in cell imaging (Alexa Fluor 350, Alexa Fluor 488, and CMTMR). Sources have common emission rates. The signal is acquired by the 16-channel spectral microscope described in Section 2.3. **a**: emission spectra of the fluorophores. **b**: detected emission spectra $\boldsymbol{s_i}$ of the fluorophores by the 16-channel spectral microscope. **c**: Pairwise distance matrix $\boldsymbol{\Xi}$ between the three components. **d:** Bar graph of the variance factors $\boldsymbol{G_i(\lambda_0)}$ for each component. The normalized geometric mean $\boldsymbol{\delta\xi}$ of the eigenvalues of $\boldsymbol{J}$ is superimposed.

It is interesting to note that the variance factors G_i of the three components can be estimated based on the product of the non-diagonal elements of the pairwise distance matrix $\boldsymbol{\Xi}$ (i.e. for alexa 350, this calculation suggests that $G_{Alexa350} \approx 0.90 \cdot 0.99 = 0.89$, $G_{Alexa488} \approx 0.90 \cdot 0.86 = 0.75$, and $G_{Alexa488} \approx 0.99 \cdot 0.86 = 0.85$, values close to the ones obtained using Eq. 2.4.30).

Figure 2.4.15 presents simulation results for the variance factors G_i, estimation bias $\boldsymbol{b}$, and computation cost per pixel when ML spectral unmixing is implemented by the three unmixing

algorithms (EM, GCN, GVN) discussed in this thesis. Results are provided for various signal levels (assuming common emission rates λ_i) and are compared with theoretical predictions for G_i, and $\boldsymbol{b}$. Simulated results for G_i and $\boldsymbol{b}$ agree with theoretical predictions very well over all signal levels. The computation cost for EM and GVN is approximately independent of signal level and approximately one order of magnitude larger than the cost of GCN (0.1 μsec per pixel for signal level larger than 60 photons per pixel).

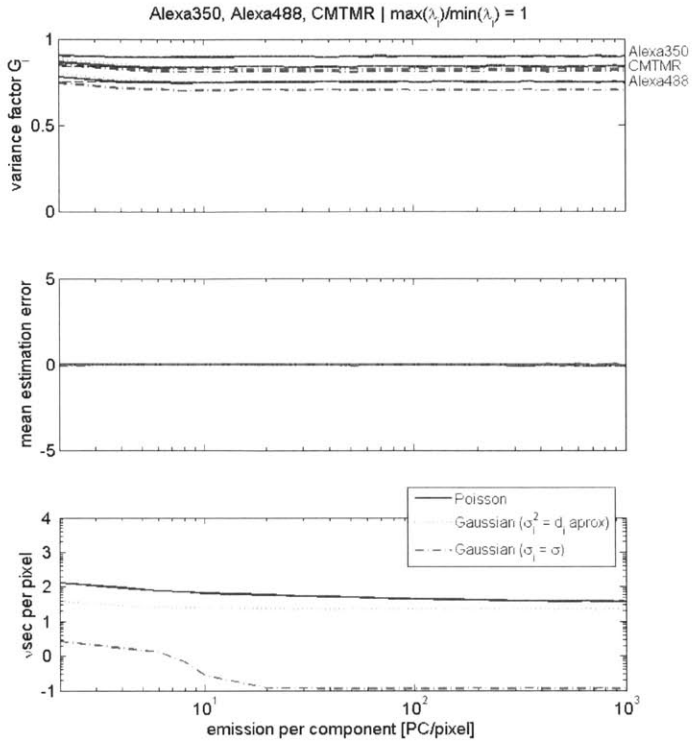


Figure 2.4.15: Simulation results for the estimation variance factor $\boldsymbol{G_i^{SIM}}$ (**top**), estimation bias (**middle**), and computation cost (**bottom**) for ML spectral unmixing of signals generated by mixing emissions of Alexa350, Alexa488 and CMTMR. Results are presented as a function of the common emission rate per component $\boldsymbol{\lambda_i}$. Data are acquired by the 16-channel spectral multi-photon microscope utilized in this thesis (Section 2.3). $\boldsymbol{G_i^{SIM}}$ are superimposed with theoretic estimations based on Eq. 18b. Results are provided for three ML spectral unmixing algorithms.

The second case (hard) considers a N=4 component system, consisting of four intrinsic fluorophores (PVD, NADH, PYO and PCA) expressed in *P. aeruginosa* bacteria (Figure 2.4.16). The emission spectra of the components have significant spectral overlap. The pairwise distance matrix $\boldsymbol{\Xi}$ components are very small. The small $\bar{\xi} = 0.078$ metric for the 4-source system (common emission rates) suggests that it is hard to estimate precisely their emission rates. The element that seems hardest to estimate is NADH because it is very similar to PVD ($\bar{\xi}\{NADH, PVD\} = 0.19$) and PYO ($\bar{\xi}\{NADH, PYO\} = 0.3$) Figure 2.4.16c. The small variance factors $G_{PVD} = 0.003$, $G_{NADH} = 0.002$, $G_{PYO} = 0.02$ and $G_{PCA} = 0.24$(Figure 2.4.16d) probably overestimate estimation variance (see discussion of Figure 2.4.6).

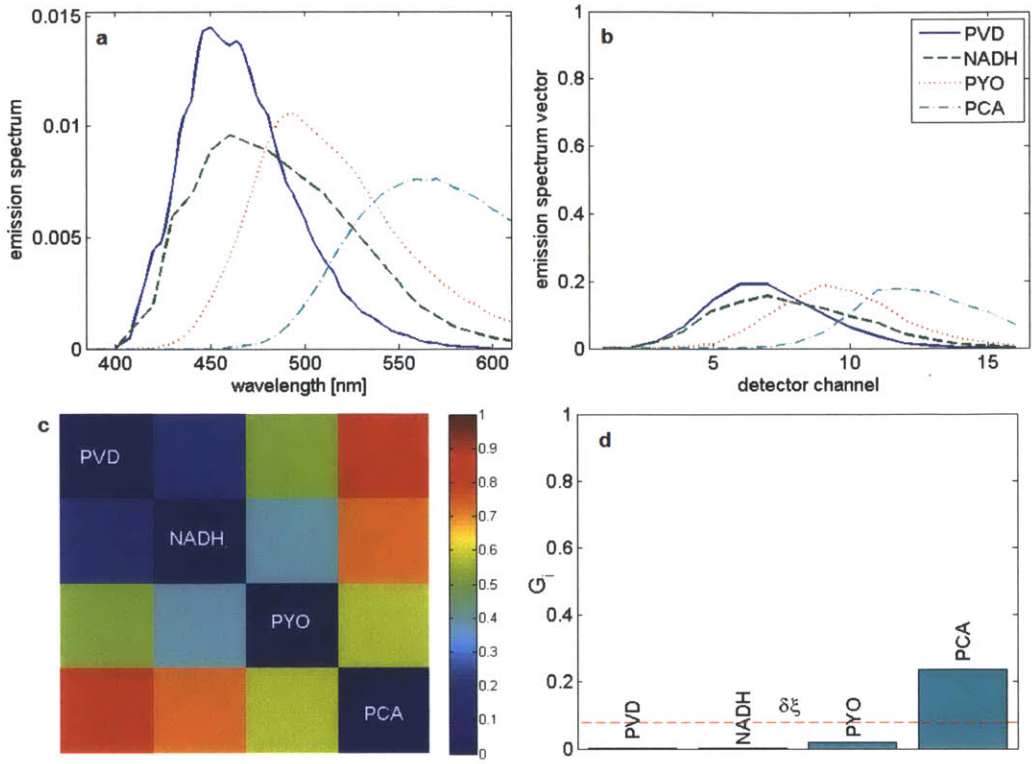


Figure 2.4.16: Theoretic analysis of spectral unmixing estimation variance from data resulting from mixing N=4 fluorophores found in bacteria (PVD, NADH, PYO and PCA). All sources have common emission rates. The signal is acquired by the 16-channel spectral microscope described in Section 2.3. **a**: emission spectra of the fluorophores. **b**: detected emission spectra $\boldsymbol{s_i}$ of the fluorophores by the 16-channel spectral microscope. **c**: Pairwise distance matrix $\boldsymbol{\Xi}$ between the four components. **d:** Bar graph of the variance factors $\boldsymbol{G_i(\lambda_0)}$ for each component. The normalized geometric mean $\boldsymbol{\delta\xi}$ of the eigenvalues of $\boldsymbol{J}$ is superimposed.

Figure 2.4.17 presents simulation results for the variance factors G_i, estimation bias $\boldsymbol{b}$, and computation cost per pixel when ML spectral unmixing is implemented by the unmixing algorithms (EM, GCN, GVN) discussed in this thesis. Results are provided for various signal levels (assuming common emission rates λ_i) and are compared with theoretical predictions for G_i, and $\boldsymbol{b}$. Simulated results for G_i and $\boldsymbol{b}$ agree with theoretical predictions reasonably only for PCA and for signal levels larger than 40 photons/pixel. Theoretical estimations of G_i and $\boldsymbol{b}$ for the remaining components (PVD, NADH and PYO) deviate significantly from simulation results. As signal level decreases, simulation results for G_i increase. As signal level increases the estimation bias of PVD, NADH and PYO deviates significantly from 0, however the simulated bias factors B_i remain under 5%. The computation cost per pixel for solving this case is approximately one order of magnitude larger than the computational cost of solving the easy case shown in Figure 2.4.15. These results agree with the trends shown in Figure 2.4.6, and Figure 2.4.9.

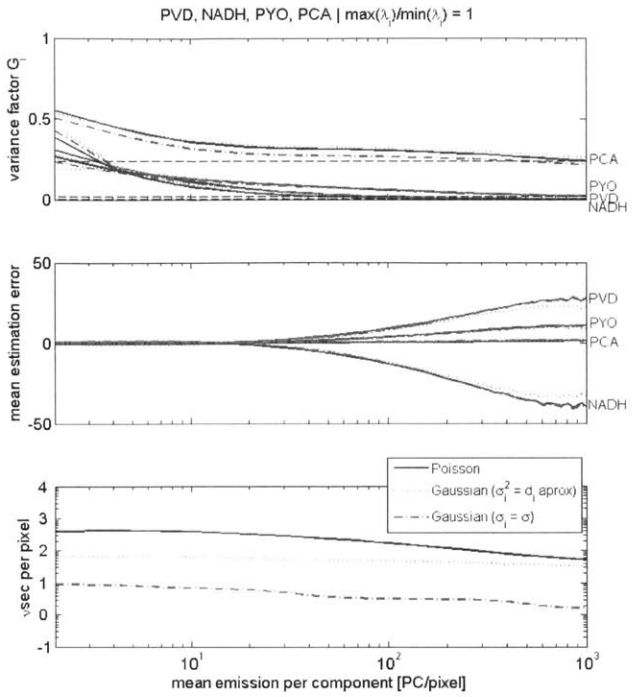


Figure 2.4.17: Simulation results for the estimation variance factor $\boldsymbol{G_i^{SIM}}$ (**top**), estimation bias (**middle)**, and computation cost (**bottom**) for ML spectral unmixing of signals generated by mixing emissions of PVD, NADH, PYO and PCA. Results are presented as a function of the common emission rate per component $\boldsymbol{\lambda_i}$. Data are acquired by the 16-channel spectral multi-photon microscope utilized in this thesis (Section 2.3). $\boldsymbol{G_i^{SIM}}$ are superimposed with theoretic estimations based on Eq. 2.4.30. Results are provided for three ML spectral unmixing algorithms.

The third case (challenging) considers a mixture of N=6 fluorescent proteins (BFP, CFP, GFP, YFP, morange, and mcherry) used extensively in biological and medical research (Figure 2.4.18). This is an example of trying to exploit the capabilities of spectral imaging by careful selection of fluorophores in order to be able to image as many sources as possible without compromising unmixing precision significantly. While the second case of N=4 sources (Figure 2.4.16) is particularly hard to unmix ($\bar{\xi} = 0.078$), the N=6 fluorescent protein case is easier to unmix ($\bar{\xi} = 0.33$). The spectrum of each fluorophore as detected by the 16-ch sensor partially ovarlaps (the values of the pairwise distance matrix $\boldsymbol{\Xi}$ lie between 0.5 and 0.7) partially one or two other spectra (Figure 2.4.18b,c). The resulting variance factors are $G_{BFP} = 0.336$, $G_{CFP} = 0.118$, $G_{GFP} = 0.134$, $G_{YFP} = 0.22$, $G_{morange} = 0.311$ and $G_{mcherry} = 0.503$ (Figure 2.4.18d).

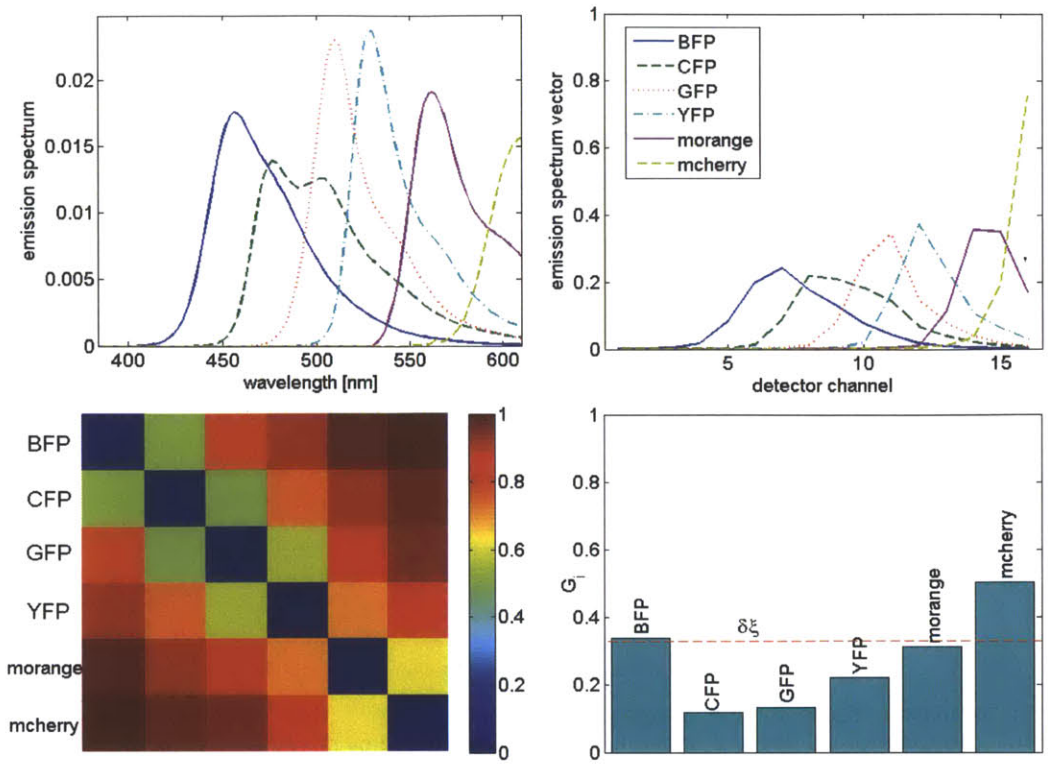


Figure 2.4.18: Theoretic analysis of spectral unmixing estimation variance from data resulting from mixing N=6 fluorescent proteins (BFP, CFP, GFP, YFP, morange, mcherry) used in biological research. All sources have common emission rates. The signal is acquired by the 16-channel spectral microscope described in Section 2.3. **a**: emission spectra of the fluorophores. **b**: detected emission spectra $\boldsymbol{s}_i$ of the fluorophores by the 16-channel spectral microscope **c**: Pairwise distance matrix $\boldsymbol{\Xi}$ between the six components. **d:** Bar graph of the variance factors $\boldsymbol{G}_i(\boldsymbol{\lambda}_0)$ for each component. The normalized geometric mean $\boldsymbol{\delta\xi}$ of the eigenvalues of $\boldsymbol{J}$ is superimposed.

Figure 2.4.19 presents simulation results for the variance factors G_i, estimation bias $\boldsymbol{b}$, and computation cost per pixel when ML spectral unmixing is implemented by the unmixing algorithms (EM, GCN, GVN) discussed in this thesis. Results are provided for various signal levels (assuming common emission rates λ_i) and are compared with theoretical predictions for G_i, and $\boldsymbol{b}$. Simulated results for G_i and $\boldsymbol{b}$ agree very well with theoretical predictions for a wide range of signal level. Only at small signal level (less than 60 photons per pixel) simulation results for G_i deviate from theoretical predictions as described in Figure 2.4.6.

Overall simulation results suggest that it is possible to image simultaneously these six fluorescent proteins via spectral imaging and estimate their emission rates without significant bias but with 2 to 8 times worse precision compared to the case that each fluorescent protein was imaged by itself. In practical applications the performance is estimated to be even worse since the max emission rate ratio R is expected to be larger than 1 (sources have nonequal emission rates). Nevertheless this case demonstrates that it is possible to achieve high information content provided the emission sources of the system are chosen in a smart way.

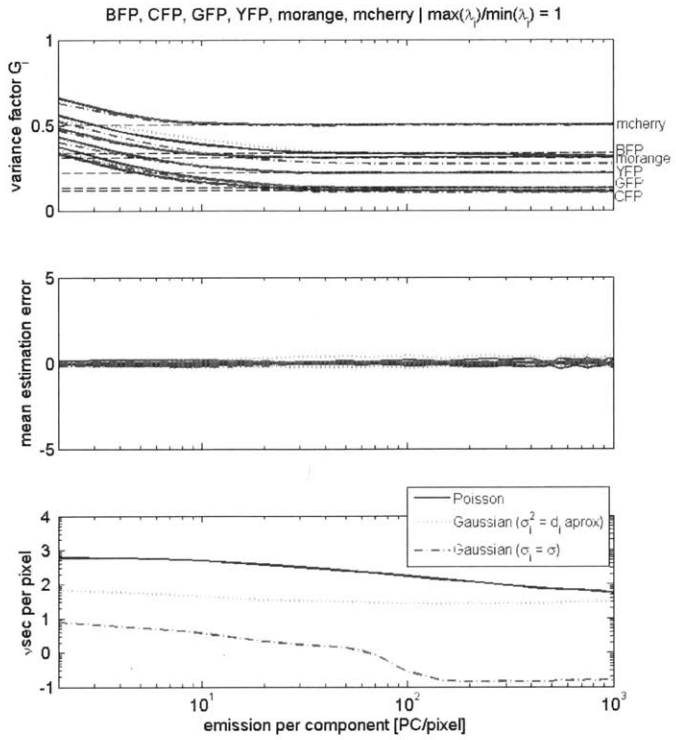


Figure 2.4.19: Simulation results for the estimation variance factor G_i^{SIM} (**top**), estimation bias (**middle**), and computation cost (**bottom**) for ML spectral unmixing of signals generated by mixing emissions of BFP, CFP, GFP, YFP, morange, mcherry. Results are presented as a function of the common emission rate per component λ_i. Data are acquired by the 16-channel spectral multi-photon microscope utilized in this thesis (Section 2.3). G_i^{SIM} are superimposed with theoretic estimations based on Eq. 2.4.26. Results are provided for three ML spectral unmixing algorithms.

The results of the three cases suggest that estimation variance using the Gauss common-noise (GCN)unmixing is always slightly larger compared to the other two algorithms (variance factors G_i are up to 0.1 less). All algorithms also provide similar small bias. Therefore, since GCN is 1-2 order of magnitude faster than the other two algorithms, GCN should be the algorithm of choice even in low level signals.

2.4.7 Simulation Results: The effect of Model Uncertainty

Section 2.4.6 considers the performance of ML spectral unmixing algorithms assuming no model uncertainty, i.e. when the identity of the sources and the detected emission spectra $\boldsymbol{s}_i$ are assumed known. This section considers the effects of model uncertainty in ML estimation.

There are several sources of model uncertainty in spectral microscopy, which can affect the precision and accuracy of spectral unmixing estimates.

- Imperfect knowledge of the spectral matrix $\boldsymbol{S}_0$ due to:
  - spectral position error
  - incomplete knowledge of the spectral response function
  - spatially variations in the spectral matrix
- incomplete knowledge of the number of sources N present in the system due to:
  - incomplete knowledge of the sources present in the system
  - neglecting to include components assuming their emission is negligible

- including more components than necessary

Several of these reasons are described in more detail in the following sub-sections.

Theoretic Description of Estimation Errors due to Model Uncertainty

Based on Section 2.4.5, under appropriate assumptions (large $\boldsymbol{\lambda_0}$) estimation error $\tilde{\boldsymbol{\lambda}}$ is a Gaussian vector $\tilde{\boldsymbol{\lambda}} \sim N\left(\boldsymbol{F}^{-1}\boldsymbol{\mu_g}, \left(\boldsymbol{F}\cdot\left(\boldsymbol{J}-\boldsymbol{\mu_g}\cdot\boldsymbol{\mu_g^T}\right)^{-1}\cdot\boldsymbol{F}\right)^{-1}\right)$, where $\boldsymbol{\mu_g} = -\boldsymbol{S}^T\cdot\boldsymbol{e}$, $\boldsymbol{F}\approx\boldsymbol{S}^T\cdot\boldsymbol{\Lambda_d^{-1}}\cdot\boldsymbol{S}$, and $\boldsymbol{J} = \boldsymbol{S}^T\cdot((\boldsymbol{I}-diag(\boldsymbol{e}))\cdot\boldsymbol{\Lambda_d^{-1}}+\boldsymbol{e}\cdot\boldsymbol{e}^T)\cdot\boldsymbol{S}$.

Model uncertainty is described by the modeling error vector $\boldsymbol{e} = \boldsymbol{1}_{M\times 1} - \boldsymbol{\Lambda_d^{-1}}\cdot\boldsymbol{S_0}\cdot\boldsymbol{\lambda_0}$, whose j-th element describes the effect of imperfect knowledge of emission spectra at channel j.

When model uncertainty is nonzero $\boldsymbol{e}\neq\boldsymbol{0}$, then estimator bias is also nonzero:

$$\boldsymbol{b} = E[\tilde{\boldsymbol{\lambda}}] = \boldsymbol{F}^{-1}\boldsymbol{\mu_g} \qquad [2.4.32]$$

and the covariance matrix of estimation error $\tilde{\boldsymbol{\lambda}}$ equals $\boldsymbol{C_{\hat{\lambda}}} = \boldsymbol{J^{*-1}}$, where

$$\boldsymbol{J^*} = \boldsymbol{F}\cdot\left(\boldsymbol{J}-\boldsymbol{\mu_g}\cdot\boldsymbol{\mu_g^T}\right)^{-1}\cdot\boldsymbol{F} \qquad [2.4.33]$$

In the case of finite model uncertainty, $\boldsymbol{J^*}$ plays the same role in the analysis of estimation variance (the inverse of the estimation error covariance matrix) as the Fisher information matrix $\boldsymbol{J}$ plays in the case of no model uncertainty (Section 2.4.5). In this case, the combined effect of mixing and model uncertainty in the ability to estimate the emission rates of sources is described by the emission-normalized geometric mean of $\boldsymbol{J^*}$ eigenvalues $\xi_i^* = eig(\boldsymbol{J^*})$:

$$\bar{\xi}^* = \sqrt[N]{\prod_{i=1}^{N}\{\xi_i^*\cdot\lambda_{0,i}\}} \qquad [2.4.34]$$

The estimation variance of the i-th source is predicted to be:

$$C_{\hat{\lambda},ii} \approx \frac{\left|\boldsymbol{J}^*_{(i,i)}\right|}{|\boldsymbol{J^*}|} \qquad [2.4.35]$$

Therefore the discussion of Section 2.4.5 about predicting the estimation variance by analyzing the $\boldsymbol{J}$ matrix applies in this section to the matrix $\boldsymbol{J^*}$. The following sub-sections apply this analysis to predict the effect of several forms of model uncertainty and compare theoretic predictions for estimation bias $\boldsymbol{b}$ and estimation error covariance (expressed as variance factor G_i via Eq. 2.4.27).

Effect of Spectral Position Error

Spectral position error describes the situation where the spectral signatures $\boldsymbol{s}_i$ considered in spectral unmixing calculations differ from the actual $\boldsymbol{s}_{0,i}$ due to a shift in the position of the emission on the detector surface (which is equivalent in a shift in the wavelength)h

$$\boldsymbol{s}_i(x(\lambda)) = \boldsymbol{s}_{0,i}(x(\lambda-\lambda_\varepsilon))$$

The effect of the spectral position error λ_ε on $\boldsymbol{s}_i$ is to consider a wavelength range $[\lambda_{min,j}, \lambda_{max,j}]$ for the spectra signature calculations (Section 5.3.7) that differs from the actual one. Such error can occur in spectral imaging due to two major reasons:

- optical aberrations can cause position-dependent shift in the position where the emission signal is focused on the M-channel detector.

- During spectral imaging in scanned configuration [Masters and So 2008] the position $x(\lambda)$ where emission is focused on the multi-channel detector depends on the position (row r, column c) of the pixel on the image. This makes the spectral signature $\boldsymbol{s}_i(r,c)$ of each source to be position-variant, i.e. different at each row r and column c of the image.

Figure 2.4.20a shows that the spectral position error λ_ε induces deviation between the spectral signature $\boldsymbol{s}_i$ used for spectral unmixing, and the actual spectral signature $\boldsymbol{s}_{i,0}$. Positive spectral difference λ_ε causes a red-shift of $\boldsymbol{s}_i$ with respect to $\boldsymbol{s}_{i,0}$, while negative λ_ε causes a blue-shift of $\boldsymbol{s}_i$ with respect to $\boldsymbol{s}_{i,0}$. The results are calculated for the spectral microscope of Section 2.3.1 ($\delta\lambda \approx 13$ nm) and show that a $|\lambda_\varepsilon|$ =10 nm error causes a shift of the signature by approximately one channel. Figure 2.4.20b shows how the 2-norm of the modeling error vector $\boldsymbol{e}$ (calculated by Eq. 2.4.25) varies as a function of spectral position error λ_ε for 50 random sets of fluorophores of $2 \le N \le 6$ sources. Results show that the modeling error is much more sensitive to positive spectral position error λ_ε compared to negative λ_ε.

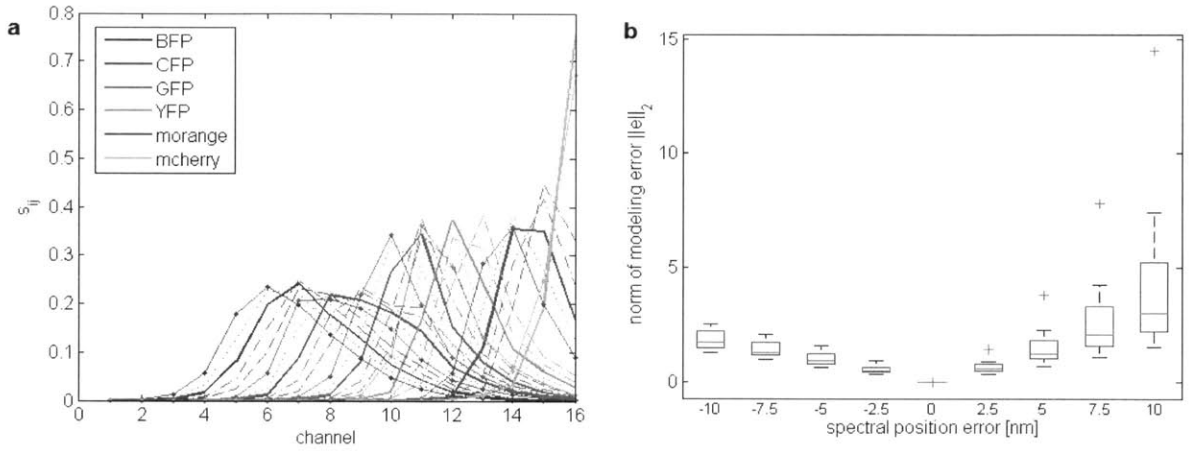


Figure 2.4.20: The effect of spectral position error $\boldsymbol{\lambda_\varepsilon}$ in the 2-norm of the model error vector $\boldsymbol{e}$. **a**: The effect of spectral position error in the emission signatures $\boldsymbol{s_i}$ of six fluorescent proteins. Bold lines show the spectral signatures assumed for spectral unmixing calculations . Light lines show the actual spectral signatures when the spectral position error is -10nm (-- --), -5 nm (.-), 5 nm (..) and 10 nm (solid line plus circular marker). **b**: Box plot for the 2-norm of the modelling error vector $\boldsymbol{e}$ for nine values of spectral position error $\boldsymbol{\lambda_\varepsilon}$. Data are obtained for fifty randomly selected sets of fluorophores of $\boldsymbol{2 \le N \le 6}$ sources. The error vector is calculated via Eq. 2.4.25. Results are presented for the spectral microscope described in Section 2.3.1.

Figure 2.4.21 and Figure 2.4.22 show how the analytic predictions of the variance factors G_i and the normalized bias B_i respectively agree with simulation results G_i^{SIM}, B_i^{SIM} in the presence of spectral position error (-10nm to 10 nm). Results show that there is good agreement between theoretic predictions and simulations, and that the level of agreement depends on the signal level Y, the max emission rate ratio R and the value of theoretic variance G_i. The larger the value of G_i, the better the agreement with simulation results G_i^{SIM}. In both figures, the agreement between theoretic predictions and simulation results improves as $R \to 1$ and as Y increases.

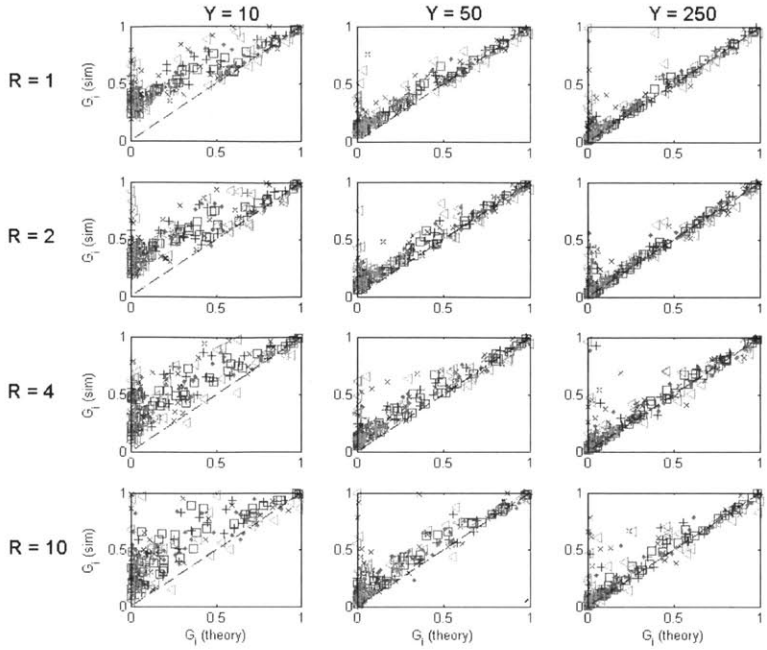


Figure 2.4.21: Comparison of the estimated variance factor $\boldsymbol{G_i}$ (based on Eq. 16, 19b, and19d) with simulation results $\boldsymbol{G_i^{SIM}}$ in the presence of spectral position error (-10 up to 10 nm) for three cases of signal level $\boldsymbol{Y}$ and four cases of maximum emission rate ratio $\boldsymbol{R}$. Simulation results are calculated based on 64000 random samples per case. Each plot shows 50 cases generated by randomly picking $\boldsymbol{2 \leq N \leq 6}$ sources from the 59 sources shown in Figure 2.4.3 and a random emission vector $\boldsymbol{\lambda_0}$ based on the specific combination of $\boldsymbol{Y}$ and $\boldsymbol{R}$. For each case there are N points $(\boldsymbol{G_i, G_i^{sim}})$ corresponding to the estimated and simulated values of the variance factors of the N sources. All ML estimates are calculated by the EM algorithm.

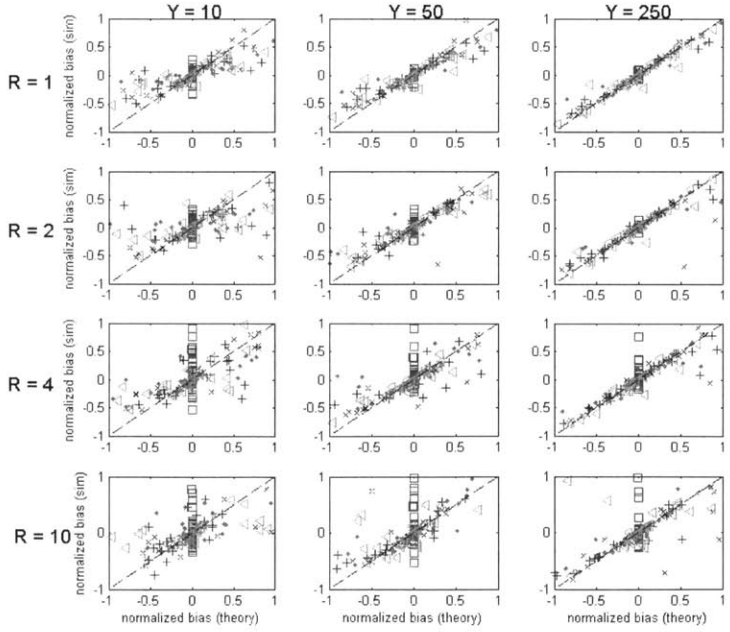


Figure 2.4.22: Comparison of the estimated normalized variance $\boldsymbol{B_i}$ (based on Eq. 2.4.6.1, 19A) with simulation results $\boldsymbol{B_i^{sim}}$ in the presence of spectral position error (-10 up to 10 nm) for three cases of signal level $\boldsymbol{Y}$ and four cases of maximum emission rate ratio $\boldsymbol{R}$. Simulation results are calculated based on 64000 random samples per case. Each plot shows 50 cases generated by randomly picking $\boldsymbol{2 \leq N \leq 6}$ sources from the 59 sources shown in

Figure 2.4.3 and a random emission vector $\boldsymbol{\lambda_0}$ based on the specific combination of $\boldsymbol{Y}$ and $\boldsymbol{R}$. For each case there are N points $(\boldsymbol{B_i}, \boldsymbol{B_i^{sim}})$ corresponding to the estimated and simulated values of the variance factors of the N sources. All ML estimates are calculated from the same simulations as the data of Fig. 2.4.19 by the EM algorithm.

The effect of the spectral position error on the precision and accuracy of the spectral unmixing calculations is shown in Figure 2.4.23. Figure 2.4.23a shows that spectral position error has little effect on estimation variance, since the presence of non-zero spectral position error does not change the value of the variance factors significantly. Figure 2.4.23b,c show that spectral position error induces estimation bias. In easy to estimate sources ($G_i > 0.25$) the effect of spectral position error is small. Spectral position error almost equal to the spectral width of each detector channel leads to bias up to 30%. Hard to estimate sources ($G_i < 0.1$) are very sensitive to spectral position. In this case, theory predicts that even small spectral position errors can induce large bias.

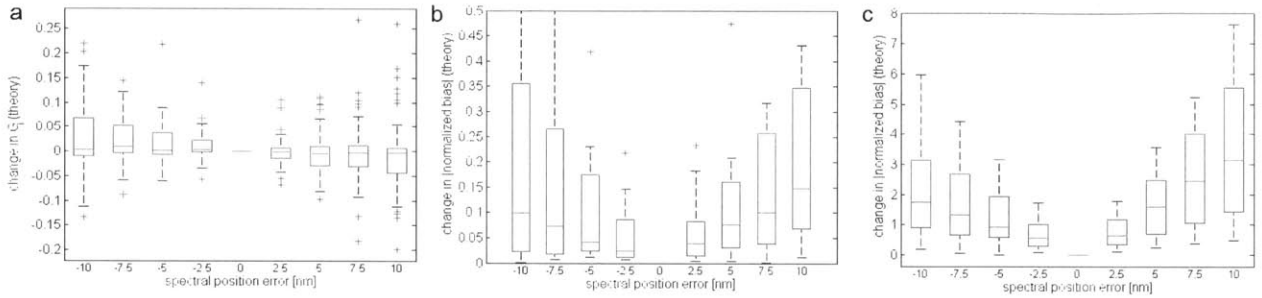


Figure 2.4.23: The effect of spectral position error on the accuracy and precision of spectral unmixing calculations. **a**: Box plot of the deviation of predicted variance factors $\boldsymbol{G_i}$ compared to the error-free case as a function of the spectral position error. **b**: Box plot of the deviation of the absolute value of the predicted normalized bias $\boldsymbol{B_i}$ compared to the error-free case as a function of the spectral position error. This image shows results for sources whose estimated error-free variance factor is $\boldsymbol{G_i > 0.25}$. **c**: Box plot of the deviation of the absolute value of the predicted normalized bias $\boldsymbol{B_i}$ compared to the error-free case as a function of the spectral position error. This image shows results for sources whose estimated error-free variance factor is $\boldsymbol{G_i < 0.1}$ (difficult to estimate sources). These results are based on the same random sets of sources as the ones shown in Figure 2.4.21 and Figure 2.4.22.

Effect of Incomplete Knowledge of the Spectral Response Function

The spectral response function $h_s(j)$ of a spectral microscope describes how the emission of a particular wavelength is distributed among the detector channels (Section 2.3.4). Ideally, emission of a particular wavelength should be detected in a single channel $(h_s(j) = \delta(j))$. However, due to optical and instrumentation reasons $h_s(j)$ is finite over more than one channels. The spectral response $h_s(j)$ affects the shape of the spectral signatures $\boldsymbol{s}_i$ (Section 2.3.6). Broader $h_s(j)$ causes broadening of the $\boldsymbol{s}_i$ shape (spreading the probability to detect the photons emitted by a source into more channels, see Figure 2.4.25), which deteriorates the ability to conduct precise ML spectral unmixing (reduce $\bar{\xi}$ metric) as discussed in Section 2.4.5. Figure 2.4.24 shows a family of spectral response $h_s(j)$ of progressively broader shape and shorter peak. This family of spectral response could correspond to the response for imaging at different planes inside a scattering sample.

Imperfect knowledge of $h_s(j)$ can cause estimation error in ML spectral unmixing because the assumed spectral signatures $\boldsymbol{s}_i$ are different than the actual signatures $\boldsymbol{s}_{0,i}$. In practice, imperfect knowledge of $h_s(j)$ can happen because of aberated focusing of emission light on the detector surface for several reasons such as:

- scattering of emission photons in thick sample imaging causes deviation of emission photons from their ideal ballistic trajectory
- optical aberrations or imperfect optical alignment.

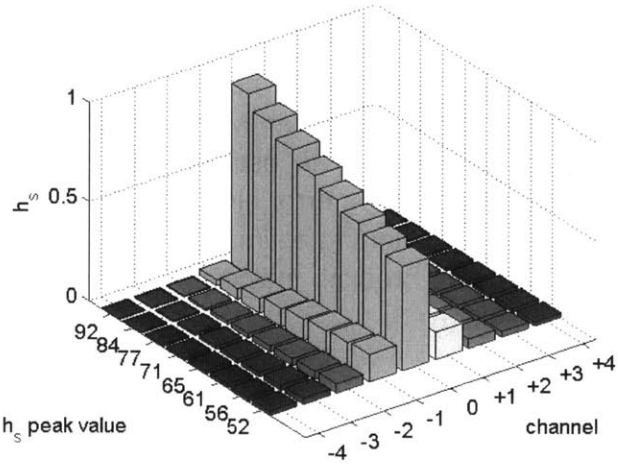


Figure 2.4.24: A family of spectral response $\boldsymbol{h_s}$ profiles. These profiles are used in the simulations of Figure 2.4.25 to Figure 2.4.28. This family could correspond either to different instruments, or could correspond to using the same instrument to image different planes inside a scattering sample. The x axis shows the channel number. The y axis shows the peak of $\boldsymbol{h_s}$ (channel 0).

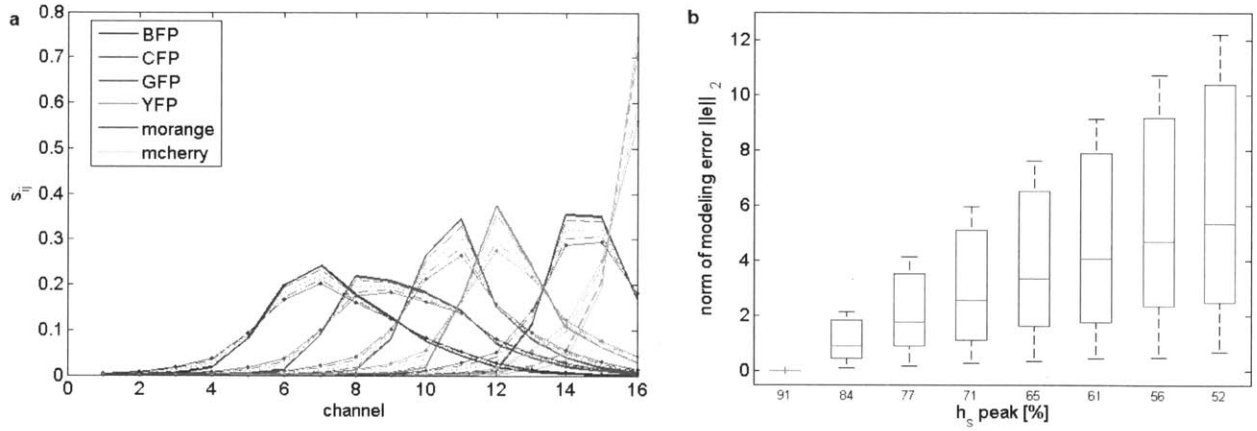


Figure 2.4.25: The effect of imperfect knowledge of spectral response in the 2-norm of the model error vector $\boldsymbol{e}$. **a**: The effect of imperfect knowledge of spectral response in the emission signatures $\boldsymbol{s_i}$ of six fluorescent proteins. Bold lines show the spectral signatures assumed for spectral unmixing calculations (corresponds to the sharpest profile of Figure 2.4.24) . Light lines show spectral signatures for the family of spectral responses shown in Figure 2.4.24. **b**: Box plot for the 2-norm of the modelling error vector $\boldsymbol{e}$ for the spectral response family shown in Figure 2.4.24. Data are obtained for fifty randomly selected sets of fluorophores of $\boldsymbol{2 < N < 6}$ sources. The error vector is calculated via Eq. 2.4.25. Results are presented for the spectral microscope described in Section 2.3.1.

Figure 2.4.26 and Figure 2.4.27 show how the analytic predictions of the variance factors G_i and the normalized bias B_i respectively agree with simulation results G_i^{SIM}, B_i^{SIM} in the case of uncertainty about the spectral response. The conclusions are similar to the ones derived from Figure 2.4.21 and Figure 2.4.22. Results show that there is good agreement between theoretic predictions and simulations, and that the level of agreement depends on the signal level Y, the max emission rate ratio R and the value of theoretic variance G_i. The larger the value of G_i, the better the agreement with simulation results G_i^{SIM}. The range of agreement increases as $R \to 1$ and Y increases.

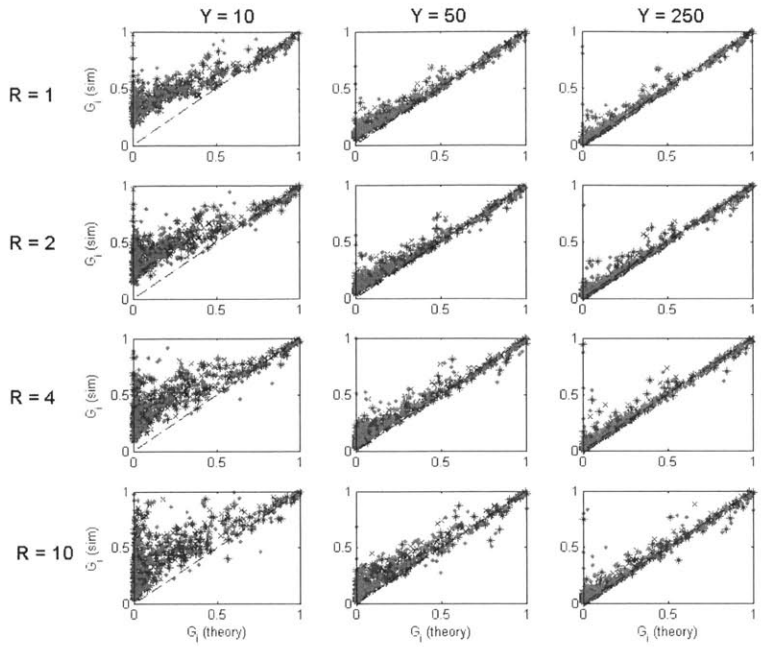


Figure 2.4.26: Comparison of the estimated variance factor $\boldsymbol{G_i}$ (based on Eq. 16, 19b, and19d) with simulation results $\boldsymbol{G_i^{SIM}}$ in the case of spectral response uncertainty (for the family of spectral response described in Figure 2.4.24) for three cases of signal level $\boldsymbol{Y}$ and four cases of maximum emission rate ratio $\boldsymbol{R}$. Simulation results are calculated based on 64000 random samples per case. Each plot shows 50 cases generated by randomly picking $\boldsymbol{2 \leq N \leq 6}$ sources from the 59 sources shown in Figure 2.4.3 and a random emission vector $\boldsymbol{\lambda_0}$ based on the specific combination of $\boldsymbol{Y}$ and $\boldsymbol{R}$. For each case there are N points $(\boldsymbol{G_i}, \boldsymbol{G_i^{sim}})$ corresponding to the estimated and simulated values of the variance factors of the N sources. All ML estimates are calculated using the EM algorithm.

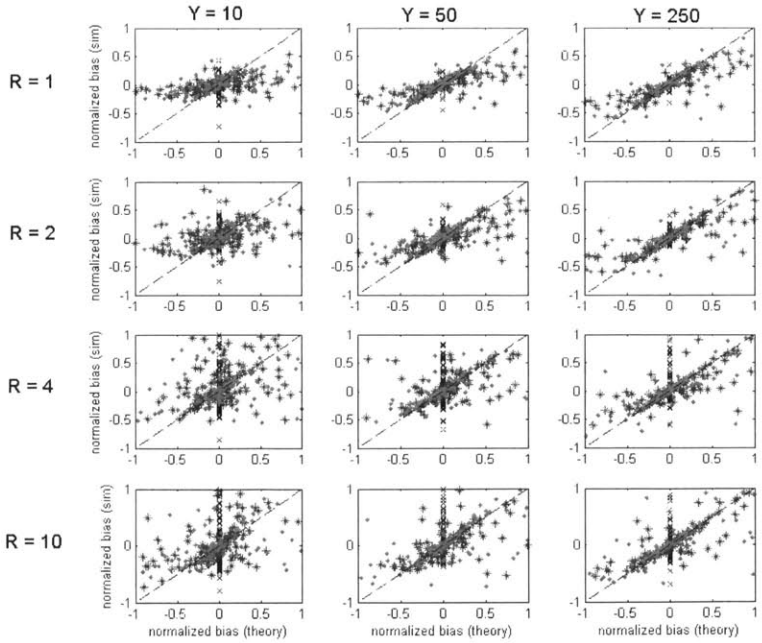


Figure 2.4.27: Comparison of the estimated normalized variance $\boldsymbol{B_i}$ (based on Eq. 2.4.6.1, 19A) with simulation results $\boldsymbol{B_i^{sim}}$ in the case of spectral response uncertainty (for the family of spectral response described in Figure 2.4.24) for three cases of signal level $\boldsymbol{Y}$ and four cases of maximum emission rate ratio $\boldsymbol{R}$. Simulation results are

calculated based on 64000 random samples per case. Each plot shows 50 cases generated by randomly picking $\mathbf{2} \leq \boldsymbol{N} \leq \mathbf{6}$ sources from the 59 sources shown in Figure 2.4.3 and a random emission vector $\boldsymbol{\lambda_0}$ based on the specific combination of $\boldsymbol{Y}$ and $\boldsymbol{R}$. For each case there are N points $(\boldsymbol{B_i}, \boldsymbol{B_i^{sim}})$ corresponding to the estimated and simulated values of the variance factors of the N sources. All ML estimates are calculated from the same simulations as the data of Figure 2.4.21 by the EM algorithm.

The effect of spectral response uncertainty on the precision and accuracy of the spectral unmixing calculations is shown in Figure 2.4.28. Figure 2.4.28a shows that spectral response uncertainty has little effect on estimation variance. Similar to Figure 2.4.23, Figure 2.4.28b,c show that spectral response uncertainty induces little estimation bias in easy to estimate sources ($G_i > 0.25$) and very significant bias in hard to estimate sources ($G_i < 0.1$).

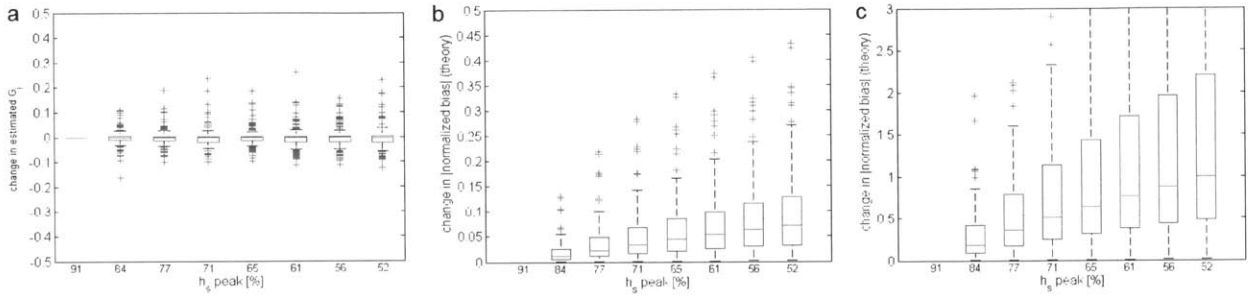


Figure 2.4.28: The effect of uncertainty on spectral response knowledge on the accuracy and precision of spectral unmixing calculations. All results are provided for the family of spectral responses shown in Figure 2.4.24. **a**: Box plot of the deviation of predicted variance factors $\boldsymbol{G_i}$ compared to the error-free case as a function of the spectral position error. **b**: Box plot of the deviation of the absolute value of the predicted normalized bias $\boldsymbol{B_i}$ compared to the error-free case as a function of the peak value of the spectral response $\boldsymbol{h_s}$. This image shows results for sources whose estimated error-free variance factor is $\boldsymbol{G_i > 0.25}$. **c**: Box plot of the deviation of the absolute value of the predicted normalized bias $\boldsymbol{B_i}$ compared to the error-free case as a function of the peak value of the spectral response $\boldsymbol{h_s}$. This image shows results for sources whose estimated error-free variance factor is $\boldsymbol{G_i < 0.1}$ (difficult to estimate sources). All results are based on the same random sets of sources as the ones shown in Figure 2.4.21and Figure 2.4.22.

All-together, the previous Figures suggest that the predictions of estimation bias and variance (Eq. 2.4.32,33) agree with simulation results. Figure 2.4.23 and Figure 2.4.28 suggest that these two modeling error do not affect the precision of spectral unmixing, however they can induce bias. The bias induced due to these two modeling errors are small for easy to unmix sources ($G_i > 0.25$) but can be very significant errors in hard to unmix samples ($G_i < 0.1$) that yield theoretical predictions useless.

Effect of Including the Wrong Set of Sources

Another source of model uncertainty is including the wrong number of sources N in the spectral unmixing calculations. This can happen by either neglecting or including extra sources in the unmixing calculations compared to the actual sources that contribute to the detected signal.

Neglecting to consider sources whose emission is significant, induces error in the estimates of the remaining samples. This kind of model uncertainty could affect the validity of spectral unmixing results particularly when the identity of sources present in the samples is unknown.

A more tricky and common source of model uncertainty is the case when spectral unmixing calculations at some pixel considers more sources than the ones actually present at that pixel. This kind of model uncertainty can take place in complex samples consisting of many classes and compartments (see Section 2.4.1), resulting in the presence of different sets of sources at

different parts of the sample. However, the exact location where each source is expressed is not known a-priori. Including all sources present in the sample in the spectral unmixing calculations of each pixel can induce significant estimation errors, especially when the sample contains sources of similar spectral signatures. An example of the errors that can arise when processing complex samples are shown in Figure 2.4.25. Such complex samples need to processed via the approach described in Section 2.5, where each pixel is eventually assigned to a particular class so that only the sources of this class are considered in the spectral unmixing calculation.

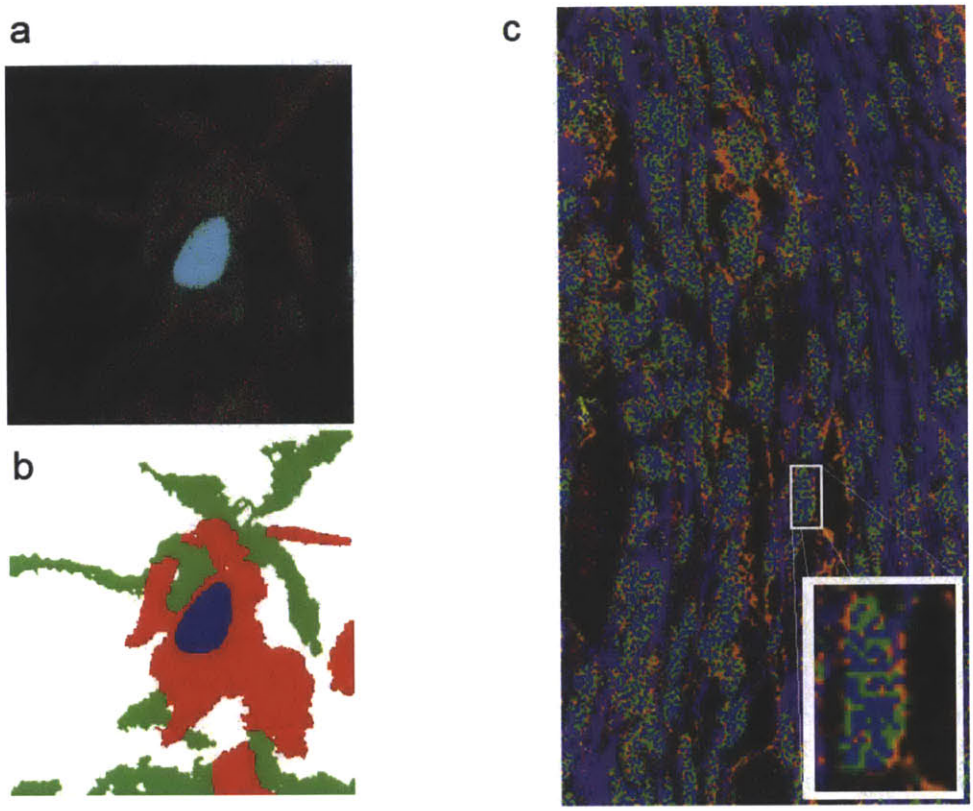


Figure 2.4.29: Including extra sources in spectral unmixing can induce significant errors in source rate estimation. **a**: RGB representation of the spectral unmixing outcome of a spectral image of a cell-seeded scaffold (Chapter 3) stained with Hoechst33342 (shown in the blue channel. Linear range 1-30 photons/pixel) , CMTMR cell tracker (shown in the green channel; linear range 1-10 photons/pixel) and alexa fluor 488-conjugated antibody (not shown here). The sample in addition contains two intrinsic sources: collagen autofluorescecne (shown in the red channel; linear range 1-5 photons/pixel) and second harmonic emission of collagen (not shown). Spectral unmixing was performed by considering for all image pixels all five sources present in the image. **b**: Outcome of global image segmentation. Blue corresponds to pixels classified as "cell nucleus" class, green to pixels classified as "collagen scaffold' class, and red to pixels classified as "cell cytoplasm" class. A comparison of images a and b shows that due to the low signal level and the similarity of the emission signatures of the compounds (Figure 2.4.3) CMTMR signal is present in the scaffold struts and collagen emission is present in the cell cytoplasm. **c:** HSV representation of the spectral unmixing outcome of a spectral image of an *ex vivo* nerve sample (Chapter 4) stained with DAPI (shown blue), phalloidin-alexa fluor 532 (shown purple), alexa fluor 350-conjugated antibody (shown orange). The sample in addition contains two intrinsic sources: collagen autofluorescecne (shown green) and second harmonic emission of collagen (shown red). The image depicts actin-rich contractile cells.). Spectral unmixing was performed by considering for all image pixels all sources present in the image. The insert image zooms in a cell nucleus. It is evident that due to the similarity of the emission signatures of alexa fluor 350, collagen autofluorescent and DAPI, there is signal contribution from all three. A correct calculation would consider only DAPI for unmixing nucleus pixels.

2.4.8 Limitations, Experimental Guidelines, and Extensions

Limitations: How many sources N can be detected in parallel

Based on the discussion of the previous sections, the number of fluorophores N that can be simultaneously detected with adequate precision is limited by the spectral region $\Delta\Lambda$ detected by the multi-channel sensor, the number of detector channels M, the emission signal level Y, the max emission rate ratio R, and the uncertainty in the knowledge of the spectral matrix $\boldsymbol{S}_0$.

The identity of the sources that compose the source set imposes an upper limit in the estimation performance of the ML spectral unmixing algorithms. As ratio R deviates away from 1 and signal level Y decreases, the performance deteriorates and is harder to predict.

Precise and accurate spectral unmixing of spectral imaging data requires careful choice of the sources and the mission rates of all sources to be of comparable magnitude ($R \to 1$). However, this may not be feasible for several reasons:

- Most commercially available fluorescent dyes are chemical derivatives of a few molecules, therefore the emission spectra of many fluorophores are quite similar. This can be observed by the clusters shown in Figure 2.4.4.
- It is not always possible for the experimenter to choose the identity and emission rates of fluorescent sources. For example, many systems of biological interest contain several auto-fluorescent molecules that have overlapping emission spectra and weak emission.
- When the detected signal is emitted by N sets, but the emission of one source is much brighter, the emission of the remaining N-1 sources cannot be estimated with predicted accuracy and precision.

The condition $R \to 1$ often cannot be satisfied in complex samples that contain many sources. In such cases, the successful application of spectral microscopy requires careful design of the staining protocols so that the sample is divided into classes, which of which contains a small (2-3) set of fluorophores and can be estimated accurately over a wider range of R and Y values.

Experimental Design for High-Content Spectral Imaging

In challenging samples that contain multiple sources, careful experimental design can enhance spectral unmixing performance and exploit the full potential of spectral microscopy. This can be accomplished by: i) proper choice of sources, ii) keeping the ratio R close to 1 (the emission rates of sources λ_i are in the same order of magnitude), and iii) increasing the signal level Y as much as possible.

Proper choice of sources requires knowledge of i) endogenous sources present in the sample, ii) available choices of dyes/secondary antibodies/fluorescent protein, iii) localization of sources in the sample, iv) spectral signatures $\boldsymbol{s}_i$ of candidate sources as detected by the particular instrument that will be utilized (e.g. Figure 2.4.3). In order to be able to estimate the emission rates of sources with adequate precision, it is necessary to pick co-localizing sources so that they come from different clusters of the distance matrix $\boldsymbol{\Xi}$, ideally pick sources of as large pairwise distance $\bar{\xi}\{i,j\}$ as possible. While the distance matrix $\boldsymbol{\Xi}$ depends on the number of detector channels M, results show that for sensors that detect a 210nm wide range of the EM spectrum, increasing M from 8 to 32 does not offer significant advantages.

In challenging samples that contain multiple sources, it is very important to keep the maximum emission rate ratio R close to 1 in order to achieve reasonable spectral unmixing performance. It

is necessary to optimize the staining protocol because the emission rates of different sources can be different due to several reasons:

- variations in the spatial distribution of sources inside the sample
- sources usually have different optical properties (cross-section, quantum yield)
- variations in biological expression of fluorescent proteins
- staining variations
- energy transfer phenomena (quenching, FRET),
- photo-bleaching.

The ratio R can be designed to be as close to 1 as possible in several ways:

- Careful design of staining protocols (immunofluorescence or fluorescent dyes) so that the emission rates λ_i due to specific staining are on the same order of magnitude. A staining calibration step needs to proceed staining and imaging large batches of samples in order to optimize staining as a whole.
- When multiple fluorescent proteins (FP) are expressed in cells or tissues, appropriate design of expression vectors (plasmid copies, promoter strength, ribosome binding site (RBS) choice) can tune the expression of fluorescent proteins in a way that counter-balances the different brightness of each FP (due to different cross-section and quantum yield) and result in emission rates on the same order of magnitude.
- Pick longer-wavelength sources to quantify entities that have more extensive spatial variation. This is because the magnitude of longer-wavelength sources affect the estimation of shorter-wavelength sources more compared to the contrary (Figure 2.4.7).

Large signal level Y enhances the performance of spectral unmixing. Larger Y can be achieved by tuning the optical parameters or the sample preparation protocol. Signal level can be increased by increasing the power of the excitation light or by increasing the pixel sampling period. Both methods suffer from severe limitations. Both methods induce photobleaching and photo-damage. Increasing excitation power also suffers from excitation saturation, while increasing pixel sampling period reduces throughput. Larger signal level Y can be also achieved by increasing the amount of sources in the sample. However, it may not be possible to increase signal level Y arbitrarily by increasing the amount of source molecules due to several reasons:

- Using more dye or antibody concentration can cause significant non-specific staining.
- Over-expression of FP or other fluorescent protein-based sensors (e.g. FRET sensors [Zhang et al. 2002]) can be cytotoxic or can perturb the cellular system of interest.
- Increasing the signal level Y must be accomplished by increasing the emission rates of all N sources so that the ratio R remains close to 1. Increasing the emission rate of all sources may not be feasible.
- Changing the emission rates of endogenous sources (auto-fluorescent molecules, multiple harmonic emission) is usually not possible. If it is necessary to estimate accurately the emission rates of weak endogenous fluorophores, then the emission rates of the remaining sources cannot be increased arbitrarily because the ratio R will deviate away from 1, therefore inducing bias and significant estimation error variance.

Implications in Instrument Design: Spectral Multi-focal Multi-photon Microscopy

There is significant research effort towards developing high-throughput spectral imaging microscopes with 3D capability which could be used to study complex biological samples of large volume (tissues, organs, cell-biomaterial constructs) [Kim et al. 2007b; Ragan et al. 2012]. One class of high-throughput instruments are multi-focal multi-photon microscopes (MMM), which increase throughput by imaging multiple locations in a biological sample in parallel. Adding spectral detection capability in MMMs promises to deliver large-area information-rich data. However, due to slight differences in the excitation and detection part of each MMM focus, accurate processing and combination of spectral MMM data acquired by different foci requires careful instrument characterization and calibration. Therefore, spectral MMM instruments can unleash their information content provided they can be calibrated easily.

Implications in Instrument Design: Spectral-Fluorescent Lifetime Imaging Microscopy

One way to increase the information content of a spectral microscope is by combining it with fluorescence lifetime microscopy (FLIM). One such promising implementation combines time-correlated single-photon counting (TCSPC) FLIM imaging with spectral imaging by utilizing a constant-fraction discriminator to process the output of each channel of a MA-PMT that operates in photon-counting mode [Becker 2005]. If the instrument utilizes a MA-PMT of M_1 channels, and classifies the lifetime of the detected photons into M_2 lifetime bins, then the total number of channels available for statistical inference is $M = M_1 \cdot M_2$. In this case the signature $\boldsymbol{s}_i$ of each source is a $M \times 1$ vector ($\|\boldsymbol{s}_i\|_1 = 1$) that corresponds (by reshaping rows) to the equivalent $M_1 \times M_2$ signature matrix $\boldsymbol{S}_i$whose rows correspond to $j = 1,2,\dots,M_1$ spectral channels, and whose columns to $k = 1,2,\dots,M_2$ lifetime channels. By designing the instrument so that elements of source signature matrix $\boldsymbol{S}_i$ are significantly nonzero over distinct rows/columns, it is possible to be increase the information content in order to be able to estimate accurately the emission of more sources or just estimate more accurately the emission of very challenging source sets

As proposed in [Neher and Neher 1994] processing data from spectral-lifetime imaging microscopy can utilize similar tools to the ones used to process spectral microscopy. Indeed it is possible to use the same mixing models, spectral unmixing algorithms and numerical implementations like the ones described in Sections 2.4.3, 2.4.4 whose performance can be analyzed by a procedure similar to the one presented in Section 2.4.5. However, there are several important features of FLIM that need to be considered:

- The fluorescent lifetime τ of a fluorophore can be sensitive to the its environment (e.g. pH, ion concentration, FRET mechanisms). Such sensitivity in the fluorescence lifetime τ induces uncertainty in the elements of signature matrix $\boldsymbol{S}_i$ of the element, which can deteriorate the precision and accuracy of the estimation approaches described in this thesis.
- In many cases, estimating the fluorescence lifetime of sources is essential (e.g. lifetime sensors, FRET experiments). Estimating simultaneously the N emission rates λ_i and the N fluorescent lifetimes τ_i of the sources is a much harder estimation problem, since the elements of $\boldsymbol{s}_i$ are non-linear functions of the unknown τ_i. A more computationally-efficient solution could be an Expectation-Maximization implementation that alternatively estimates λ_i given the current estimate of τ_i, followed by improved τ_i estimation given the current λ_i estimate.

2.5 Segmentation of Multi-Class Spectral Microscopy Images

2.5.1 Overview

Problem Definition

Image segmentation is the process of partitioning a digital image into regions of particular kinds (classes) in order to provide a meaningful representation of an image that is easier to subsequently analyze [Shapiro and Stockman 2001]. Mathematically, image segmentation is called "coloring" because each image pixel is assigned a discrete "label" from a set of possible labels [Strauss 1977].

Supervised image segmentation is based on a-priori information provided by the user about the properties of each class. Unsupervised segmentation attempts to estimate both the correct pixel partition as well as the properties of the different partitions, a much harder inverse problem. This work considers the application of Bayesian supervised segmentation in 3D images acquired by spectral multi-photon microscopy.

Image segmentation in spectral MPM images can be implemented based on three kinds of information:

- *Image intensity*: the intensity level at different classes could be different due to different presence and amount of sources.
- *Emission spectrum*: the emission signatures detected at different classes could be different due to the presence of different kinds of sources or due to the presence of same sources at different composition ratios.
- *Class neighborhood*: classes represent physical entities (e.g. cell nuclei, cell cytoplasm, nerve axons, collagen fibers) whose dimensions lie within specific meaningful limits. Each class can neighbor particular kinds of classes and cannot neighbor others (e.g. a pixel labeled as "nucleus" should not lie next to a pixel labeled as "collagen fiber"). This makes possible to distinguish classes of similar composition (sources, composition ratios) as long as they can neighbor different classes.

In this work, it is assumed that the user has a relatively good knowledge of the chemical composition (equivalently the gene expression pattern) of each class:

- Number of classes present in the image
- Identity of sources present in each class
- A coarse range of expected emission rates for all sources present in each class.

In many biological experiments, these are not demanding assumptions

Exploiting Image Segmentation to Increase the Information Content of Spectral Imaging

The analysis of spectral unmixing (Section 2.4) suggests that there is an upper bound on the number of sources that can be accurately unmixed in a single pixel of an image. This limit arises from several experimental constraints:

- Spectral similarity of source emission spectra
- Spatial extent of the source emission spectra
- Uncertainty in source emission spectra knowledge
- Availability of fluorophores for particular applications (dyes, antibody conjugates, fluorescent proteins)

- Experimental noise: Detector noise, Poisson noise (low emission rate regime)
- Number of sources that can be excited by excitation light
- Detecting range and resolution of the sensor

Therefore extending the information content of an image just by increasing the number of sources present in the image is limited. On the other hand, it is possible to extend the information content by carefully staining the sample so that the image consists of many information-rich yet "distinguishable" classes.

Image Segmentation Pipeline

Image segmentation of spectral images is implemented in two steps:

i) *Local (pixel-wide) segmentation* (Section 2.5.2): For each pixel, this step first calculates the probability (called "evidence") that the pixel belongs to one of the possible classes present in the image. The calculation utilizes only the data measured at that pixel and prior information on the class presence. This calculation is able to identify the coarse location of class regions, however it suffers from noise (especially when the signal level is low or when the composition range of classes have overlap), inaccurate estimation of region boundaries, and lack of ability to incorporate constraints about the spatial relationships of different classes. Part of the noise is removed by processing the evidence calculation using simple morphological-based operations [Soille 2003]. Finally, each pixel is labeled with a class by a Bayesian classifier.

ii) *Global (region-wide) segmentation* (Section 2.5.3): The outcome of the pixel-wide segmentation is refined using Markov random field (MRF) modeling and global optimization [Li 2009]. Region-wide segmentation methods provide the optimal labeling for a neighborhood of pixels by considering the signal detected at all these pixels, and a-priori information about the spatial relationships of different classes [Richards and Jia 2006]. Due to the high computational cost of global segmentation, in this thesis global segmentation is applied to a fraction of the image, particularly at the boundaries of the regions and at locations were valid spatial class neighborhood rules are broken.

2.5.2 Local Image Segmentation by Bayesian Classification

Image Segmentation as a Model Selection Problem

In local (pixel-wide) image segmentation, labeling each pixel with a class can be posed as a model selection problem: Given the emission spectrum $\boldsymbol{y}$ measured at a pixel and prior information about the classes present in the image, the objective is to label the pixel with the class that best describes this emission.

Let D denote the number of classes present in the image, N denote the number of sources present in the image, and N_d denote the set of sources present at class $d = 1,2,\dots,D$. Let $\lambda_{d,i}$ (i = 1, 2, ,...N_d) denote the emission rate of the i-th source present in class d. The following properties are assumed to be known:

- the number of classes D and the elements of sets N_d (sources present in each class)
- the a-priori probability $p_D(d)$ of occurance of class d in the image. If this is unknown then $p_D(d)$ can be the non-informative (contant) prior

- a coarse estimate of the range $[\lambda_{d,i}^L, \lambda_{d,i}^U]$ for the possible values of emission rates $\lambda_{d,i}$ of the sources at each class

Local Segmentation Algorithm

The proposed local segmentation algorithm consists of three parts:

1. for each pixel calculate the calculate the $D \times 1$ evidence vector $\boldsymbol{\varepsilon}_i$
   - for each one of the D classes perform spectral unmixing to calculate the ML estimate $\boldsymbol{\lambda}_d^{ML}$ of the N_d sources present in the class based on the spectral data $\boldsymbol{y}$
   - for each one of the D classes calculate the evidence $P(d|\boldsymbol{y})$ that this pixel belongs to this class. Normalize to calculate the $D \times 1$ evidence vector $\boldsymbol{\varepsilon}_i$
   - The evidence vectors of the $R \times C$ pixels are combined to form the $R \times C \times D$ evidence array $\mathcal{E}$. The d-th plane $\mathcal{E}_d$ contains the evidence for class d.
2. Refine the evidence calculations by morphologic image processing
   - Each array $\mathcal{E}_d$ is treated as grayscale image and is processed via a series of morphological operators that provides a $R \times C \times D$ refined evidence array $\hat{\mathcal{E}}$.
   - Each pixel of the refined evidence array $\hat{\mathcal{E}}$ is normalized so that the sum of its planes equals 1. Therefore each plane $\hat{\mathcal{E}}_d$ contains the updated evidence for class d.
3. Each pixel is classified into a particular class based on the values of the corresponding $\hat{\mathcal{E}}_d$ pixel.

Class Evidence Calculation

Given the measured emission spectrum $\boldsymbol{y}$ at a pixel, Baye's theorem provides the probability that this pixel belongs to class d [Richards and Jia 2006]:

$$P(d|\boldsymbol{y}) = \frac{P(\boldsymbol{y}|d)\cdot p_D(d)}{\sum_{t=1}^{D}\{P(\boldsymbol{y}|t)\cdot p_D(t)\}} \propto P(\boldsymbol{y}|d)\cdot p_D(d) \qquad [2.5\text{-}1]$$

where i refers to the index of the pixel, and $p_D(d)$ is the a-priori probability that a pixel belongs to class d. The denominator is common in all classes and is skipped. The probability

$$P(\boldsymbol{y}|d) = \iiint_{-\infty}^{+\infty} p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}_d)\cdot p_{\boldsymbol{\Lambda}}(\boldsymbol{\lambda}_d)d\boldsymbol{\lambda}_d \qquad [2.5\text{-}2]$$

called the "evidence" of class d, is a partition function-like integral that incorporates the probability of detecting the emission $\boldsymbol{y}$ for all possible emission rates of the sources present in the class [MacKay 2003]. The elements of the $N_d \times 1$ vector $\boldsymbol{\lambda}_d$ are the emission rates $\lambda_{d,i}$ for the sources of class d. The pdf $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}_d)$ is the observation model for class d, providing the probability of detecting the spectrum $\boldsymbol{y}$ when the emission rates of the N_d sources equal $\boldsymbol{\lambda}_d$ (see Sections 2.4.3 and 2.4.4).

The PDF $p_{\boldsymbol{\Lambda}}(\boldsymbol{\lambda}_d)$ describes the possible values for the parameter vector $\boldsymbol{\lambda}_d$ of class d. In this thesis for simplicity, it is assumed that the a-priori knowledge on each $\lambda_{d,i}$ is described as a uniform PDF between boundaries $\lambda_{d,i}^L$ and $\lambda_{d,i}^U$:

$$p_{\boldsymbol{\Lambda}}(\boldsymbol{\lambda}_d) = \prod_{i=1}^{N_d} p_{\Lambda}(\lambda_{d,i}) \qquad [2.5\text{-}3]$$

$$p_{\Lambda}(\lambda_{d,i}) = \begin{cases} (\lambda_{d,i}^U - \lambda_{d,i}^L)^{-1}, & \lambda_{d,i}^L < \lambda_{d,i} < \lambda_{d,i}^U \\ 0, & otherwise \end{cases}$$

So that the class evidence integral becomes:

$$P(\boldsymbol{y}|d) = \frac{I_d}{v_d} \quad [2.5\text{-}4]$$

$$I_d = \iiint_{\lambda_{d,i}^L}^{\lambda_{d,i}^U} p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}_d) d\boldsymbol{\lambda}_d \quad [2.5\text{-}5]$$

$$v_d = \prod_{I=1}^{N_d} (\lambda_{d,i}^U - \lambda_{d,i}^L) \quad [2.5\text{-}6]$$

where v_d is the volume of the N_d-dimensional hypercube V_d bounded by the limits $[\lambda_{d,i}^L, \lambda_{d,i}^U]$ of variables $\lambda_{d,i}$.

For the i-th pixel of an image the $D \times 1$ evidence vector $\boldsymbol{\varepsilon}_i = [\, \varepsilon_{i,1} \quad \cdots \quad \varepsilon_{i,D}]^T$ contains the belief that this pixel belongs to each one of the D classes. The d-th element of $\boldsymbol{\varepsilon}_i$:

$$\varepsilon_{d,i} = \frac{P(d|\boldsymbol{y})}{\sum_{t=1}^{D} P(t|\boldsymbol{y})} = \frac{\frac{I_d(y)}{v_d} p_D(d)}{\sum_{t=1}^{D} \frac{I_t(y)}{v_t} p_D(t)} \quad [2.5\text{-}7]$$

describes a metric of probability that this pixel belongs to a region of class d. The evidence vector $\boldsymbol{\varepsilon}$ is such that $\sum_{t=1}^{D} \varepsilon_d = 1$.

Combined together, the evidence vectors of an $R \times C \times M$ spectral image generate a $R \times C \times D$ evidence array $\mathcal{E}$. The elements of $\mathcal{E}$ at row i and column j is the evidence vector of the corresponding pixel. The d-th plane $\mathcal{E}_d$ of array $\mathcal{E}$ contains the evidence for class d. Each plane $\mathcal{E}_d$ can be treated as a grayscale image (its values lie between 0 and 1). Therefore evidence calculation can be thought as a pixel-wise transformation of the original data $\{\boldsymbol{y}\}$ (an $R \times C \times D$ array) into an $R \times C \times D$ evidence array $\mathcal{E}$.

Numerical Calculation of the Evidence Integral

At each pixel, it is required to calculate the $D \times 1$ evidence vector $\boldsymbol{\varepsilon}_i$ based on the spectral measurement $\boldsymbol{y}$. This requires the calculation of D integrals $I_d(\boldsymbol{y})$ (one for each class) that are functions of the spectral measurement $\boldsymbol{y}$ at that pixel. The calculation of I_d is the most computationally expensive part of local image segmentation, however this cost can be reduced by considering the shape of the corresponding observation model $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})$ with respect to the argument $\boldsymbol{\lambda}$. Due to the convexity of $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})$ its value possesses a single peak at the ML solution $\boldsymbol{\lambda}^{ML}$ (see Sections 2.4.3 and 2.4.4) and diminishes away from the peak [Shepp and Vardi 1982]. The magnitude of $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})$ is significant only in a neighborhood around its peak. The extent of this neighborhood can be estimated by modeling the peak as a maultivariate Gaussian function and approximating the variance $\sigma_{d,i}^2$ based on the inverse of the Fisher information matrix (Eq. 2.4.29, Section 2.4.5).

Practically there are two ways to calculate $I_d(\boldsymbol{y})$.

1. If the $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})$ peak lies entirely within the hypercube defined by $p_{\boldsymbol{\Lambda}}(\boldsymbol{\lambda}_d)$:

   $$\lambda_{d,i}^L - 2\sigma_{d,i} < \lambda_{d,i}^{ML} < \lambda_{d,i}^U + 2\sigma_{d,i} \quad , \quad i = 1,2,\ldots,N_d$$

   then the peak is not truncated by the hypercube boundaries and I_d can be approximated analytically using Laplace's method [Kass and Raftery 1995]:

   $$I_d = p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}_d^{ML}) \cdot \sqrt{\frac{(2\pi)^{N_d}}{|\boldsymbol{J}_d(\boldsymbol{\lambda}_d^{ML})|}}$$

where N_d is the number of source at class d, $\boldsymbol{\lambda}_d^{ML}$ is the ML estimate of the $N_d \times 1$ source rate vector $\boldsymbol{\lambda}$ given the measurement $\boldsymbol{y}$, and $|\boldsymbol{J}_d(\boldsymbol{\lambda}_d^{ML})|$ is the determinant of the fisher information matrix for class d calculated at the ML estimate $\boldsymbol{\lambda}_d^{ML}$. The calculation of I_d using the Laplace method is very fast and is based on the calculation of the ML estimate $\boldsymbol{\lambda}_d^{ML}$ via spectral unmixing described in Section 2.4.

2. If the $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})$ peak does not lie entirely within the boundaries of the hypercube V_d, the integral I_d needs to be calculated via numerical integration.

$$I_d \approx V_s(d) \cdot \sum_{s=1}^{N_s} p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}_d(s))$$

where $\boldsymbol{\lambda}_d(s)$ are samples of $\boldsymbol{\lambda}_d$ obtained from a volume $V_s(d)$ that within the hypercube V_d, where $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}_d)$ magnitude is significant. Numerical calculation of the multi-dimensional integral I_d can be implemented using either classical quadrature methods or Monte Carlo Methods [Press et al. 2007]. Classical quadrature methods sample $\boldsymbol{\lambda}_d(s)$ using a grid of $V_s(d)$ but are considered inefficient for solving multi-dimensional integrals due to the large sample number N_s required. Simple Monte Carlo methods (usually the only solution for large dimensionale integrals) sample the volume $V_s(d)$ randomly using an appropriate sampler (e.g. the Sobol sampler [Press et al. 2007]). The brute-force application of the Monte Carlo method is still computationally inefficient. The bounds $[\lambda_{d,i}^L, \lambda_{d,i}^U]$ that describe the user's a-priori knowledge define a hypercube that can be much larger than the region where $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}_d)$ takes significant values, therefore a small fraction of $\boldsymbol{\lambda}_d(s)$ will sample the peak of $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}_d)$, causing significant integration errors. As the sample size N_s increases, I_d calculation precision improves but the computational cost can become prohibitive for large data sets. Advanced Monte Carlo methods [MacKay 2003; Sivia and Skilling 2006; Press et al. 2007] could provide better I_d estimates, but still the computational cost of performing d Monte Carlo integrations per pixel in a large data set is prohibitive.

It turns out that the choice of the uniform prior simplifies the calculation of I_d significantly. Monte Carlo methods are designed to calculate integrals of functions that contain few sharp peaks at unknown locations. Since $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}_d)$ has a single peak at a known location (the ML estimate $\boldsymbol{\lambda}^{ML}$), I_d can be calculated quite efficiently by choosing $V_s(d)$ as the part of V_d where $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda}_d)$ magnitude is large, here it is chosen as the part of the region $\lambda_{d,i}^L - 2\sigma_{d,i} < \lambda_{d,i}^{ML} < \lambda_{d,i}^U + 2\sigma_{d,i}$ that lies inside V_d.

The numerical calculation of I_d via numerical integration becomes more expensive when the signal level is low (below 50 photon counts per pixel) for two main reasons:

- At low signal level, the width of the peak of $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})$ increases, making harder to satisfy the conditions required for implementing the fast Laplace's method. This means that the slower numerical integration is implemented more often.
- Utilizing the Poisson observation model $p_{\boldsymbol{Y}|\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{\lambda})$ is much more computationally expensive compared to using a Gaussian approximation of the Poisson PDF (Section 2.4.4). However, the appropriate approximation depends on the magnitude of the Poisson parameter. For those channels whose estimated photon rate $d_j = \boldsymbol{t}_j^T \cdot \boldsymbol{\lambda}_s$ is

extremely low (<20 photons per pixel) a better approximation to the Poisson observation is [Rich 2009]:

$$\frac{e^{-d_j} \cdot {d_j}^{y_j}}{y_j!} \approx \frac{1}{\sqrt{2\pi d_j}} \cdot \exp(-\frac{(y_j + 0.5 - d_j)^2}{2 \cdot d_j})$$

The Poisson model $p_{Y|\Lambda}(\boldsymbol{y}|\boldsymbol{\lambda})$ should be used without approximation when the estimated photon rate $d_j = \boldsymbol{t}_j^T \cdot \boldsymbol{\lambda}_s < 1$ because in this range all Gaussian approximations of the Poisson induce larger errors.

Evidence Refinement by Morphological Imaging

The calculation of the evidence vector $\boldsymbol{\varepsilon}_i$ is a function of the properties of the D classes and the measurement $\boldsymbol{y}_i$ at this pixel. A classification of the pixel to a particular class based just on this calculation is prone to errors, especially when the signal level is low, or when there exist multiple classes of similar properties. In order to improve image segmentation performance, it is necessary to include information about the spatial context present in the image. While this is mainly implemented by the global image segmentation algorithm described in Section 2.5.3, the performance of the local segmentation algorithm can be improved by applying some limited morphological processing that corrects a large fraction of some easy to fix errors. This is of particular interest for the overall computational efficiency of the pipeline since this correction can reduce significantly the number of pixels requiring global segmentation, which is a very computationally expensive task.

Morphological image processing is based on elementary kernels that process neighborhoods of grayscale images [Soille 2003] to extract or remove image features. Morphological image processing utilizes spatial context embedded in the kernels. The purpose of utilizing morphological image processing here is just to remove some obvious classification errors locally by correcting evidence array calculation that is used for pixel classification.

For each plane of $\mathcal{E}_d$ the morphological processing applied is a sequence of three operators: i) morphological reconstruction $R_g^\delta(\cdot)$, ii) morphological closing $\phi(\cdot)$, and iii) median filtering $\mathcal{M}(\cdot)$:

$$\hat{\mathcal{E}}_d = \mathcal{M}(\phi(R^\delta_{\phi(\mathcal{E}_d)}(\gamma(\mathcal{E}_d))))$$

Morphological reconstruction $R_g^\delta(f)$ of a "marker" image f with respect to a "mask" image g refers to repeated geodesic dilations $\delta_g(f)$ of the "marker" image f with respect to the "mask" image g until stability (more geodesic dilations do not change the outcome). Geodesic dilation $\delta_g(f) = \min(\delta(f), g)$ refers to the pixel-wise minimum of g and the dilation $\delta(f)$. Morphological opening $\gamma(f) = \delta(\varepsilon(f))$ of an image f refers to sequential elution $\varepsilon(\cdot)$ and then dilation $\delta(\cdot)$. Morphological closing $\phi(f) = \varepsilon(\delta(f))$ of an image f refers to sequential dilation $\delta(\cdot)$ and then elution $\varepsilon(\cdot)$. All morphological operators assume the 4-neighborhood system [Soille 2003].

Morphological opening of a grayscale image removes positive artifacts (small regions of high intensity compared to its surroundings). Morphological closing of a grayscale image removes negative artifacts (small regions of low intensity compared to its surroundings). Morphological reconstruction $R_g^\delta(f)$ dilutes the "marker" image in a way that is constrained by the "mask" image. Here the "marker" image was chosen as the opening $f = \gamma(\mathcal{E}_d)$, while the "mask" was chosen as the morphological closing $g = \phi(\mathcal{E}_d)$. The outcome of morphological reconstruction

$R^{\delta}_{\phi(\mathcal{E}_d)}(\gamma(\mathcal{E}_d))$ provides a dilution of $\gamma(\mathcal{E}_d)$ that is always less than $\phi(\mathcal{E}_d)$. This operation achieves to remove both negative and positive artifacts. All morphological operators in this step utilize a disk-shaped structuring element of radius 1 pixel [Soille 2003; Li 2009].

The above described morphological reconstruction has the overall effect that it increases the outline of regions by an offset of approximately 1 pixel. This offset is corrected by first applying the second operator (morphological closing $\phi(\cdot)$) and then applying the third operator (median filtering $\mathcal{M}(\cdot)$). The kernel of the median filter was chosen to be 3×3 pixels, which also results in increasing the size of the objects by approximately 1 pixel. To compensate for the dilation effects of the first and third operator, the structural element of the second operator (morphological closing) was chosen to be a disk-shaped structuring element of radius 2 pixels.

Applying the sequence of morphological operators to each plane $\mathcal{E}_d$ of the evidence array $\mathcal{E}$ provides a grayscale image that is used as the d-th plane $\hat{\mathcal{E}}_d$ of the refined evidence array $\hat{\mathcal{E}}$. Each pixel of the refined array $\hat{\mathcal{E}}$ is then normalized so that the sum of its D elements equals 1, and corresponds to 1 the refined evidence vector $\hat{\boldsymbol{\varepsilon}}_i$.

Pixel Classification

In the last step of local image segmentation, each pixel is assigned to a label (a particular class) based on its refined evidence vector $\hat{\boldsymbol{\varepsilon}}_i$ and a selection law. Let l_i denote the label of pixel i . A simple selection law is the maximum a-posteriori estimate:

$$l_i = argmax\{\hat{\boldsymbol{\varepsilon}}_i\}$$

where pixel i is labeled to belong to the class that corresponds to the largest value in the refined evidence vector $\hat{\varepsilon}_i$.

Another more elaborate way is to divide the D-dimensional space $R^D_* = \{\boldsymbol{x} = [x_1 \quad \cdots \quad x_D]^T \in R^D : 0 \leq x_i \leq 1, \|\boldsymbol{x}\|_1 = 1\}$ into D complementary decision regions $\mathcal{H}_d$ $(d = 1,2,\ldots D)$ and decide the class of pixel I based on which decision region contains the refined evidence vector $\hat{\boldsymbol{\varepsilon}}_i$ of the pixel [Gallager 2009; Wornell 2010]. Decision regions $\mathcal{H}_d$ can be constructed before this classification step by i) generating simulated data for each one of the D classes based on the source rates bounds $\left[\lambda^L_{d,i}, \lambda^U_{d,i}\right]$ provided by the user, ii) calculating the evidence vector $\boldsymbol{\varepsilon}_i$ for the simulated data, iii) partition R^D_* into regions and assign each region to the class that generates the largest number of $\boldsymbol{\varepsilon}_i$ that lie in this region.

The final outcome of the local segmentation task is a $R \times C$ local segmentation array $\boldsymbol{L}_l$, whose elements take integer values in the set $\{1,2,\ldots D\}$ and describe the resulting segmentation of the image. $\boldsymbol{L}_l$ can be equivalently described by a $R \times C \times D$ binary local segmentation array $\boldsymbol{B}_l$, whose elements take values 0 or 1. The pixels of the d-th plane $\boldsymbol{B}_{l,d}$ equal 1 only at the pixels classified to domain d in $\boldsymbol{L}_l$.

2.5.3 Global Image Segmentation by Markov Random Fields

Since local segmentation methods do not incorporate spatial context information, their outcome suffers from several errors:

- Many pixels can be misclassified into a wrong class. While ideally image segmentation partitions the image into regions (each one belonging to a particular class), in practice these regions are corrupted with pixels misclassified, generating a salt and pepper style noise

pattern. This noise makes very hard to estimate correctly the extent and geometric properties (surface, boundary) of the corresponding physical objects (e.g. cells and fibers) and degrades or prevents their quantification.

- Local segmentation does not prevent neighboring pixels from belonging to incompatible classes, that is classes that cannot be in contact.

Global segmentation methods can prevent or reduce significantly these errors by incorporating spatial context information and segmenting a set of pixels simultaneously instead of segmenting individual pixels. Markov random field (MRF) modeling can be utilized to conduct global image segmentation [Li 2009]. Since all published methods for solving MRF are very computationally expensive, and high-throughput microscopy provides very large data sets, it is extremely computationally expensive to conduct global image segmentation in these data sets. This section describes an efficient way to implement MRF-based global image segmentation in high-throughput 3D spectral microscopy data. The basic idea is that MRF-based image segmentation is implemented to refine the outcome of the computationally cheaper (but more error-prone) local segmentation method described in Section 2.5.2. Specifically MRF is applied to segment neighborhoods of pixels located at the interface of the regions provided by the local image segmentation algorithm.

Markov Random Field (MRF) Modeling

As described in Section 2.2.5, a Markov Random Field (MRF) is a set of random variables $\mathcal{F} = [F_1, F_2, \dots, F_m]$ defined on an undirected graph $\mathcal{G}$ (a set of sites $\mathcal{S}$ with respect to a neighborhood system $\mathcal{N}$) that satisfy the positivity and Markov property [Li 2009; Koller and Friedman 2009]. In the image segmentation problem, every site (node) describes a pixel in the image, and corresponds to a random variable F_i whose value f_i can take a value from the discrete-value label set $\mathcal{L} = \{1,2, \dots, D\}$ that corresponds to the D classes. MRF models fuse the observed image $\{\boldsymbol{y}\} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_m]$ and a model of spatial context by describing the joint probability $\mathbb{P}(\{f\}|\{\boldsymbol{y}\})$ of a configuration $\{f\} = [f_1, f_2, \dots, f_m]$ based on a posterior energy function $U(\{f\}|\{\boldsymbol{y}\})$. The joint probability $\mathbb{P}(\{f\}|\{\boldsymbol{y}_i\})$ of a configuration $\{f\}$ given a spectral image $\{\boldsymbol{y}\} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_m]$ is calculated as a Gibbs distribution [Hammersley and Clifford 1971; Li 2009]:

$$\mathbb{P}(\{f\}|\{\boldsymbol{y}_i\}) = Z^{-1}\exp(-U(\{f\}|\{\boldsymbol{y}_i\})) \quad [2.5.3]$$

$$U(\{f\}|\{\boldsymbol{y}_i\}) = U(\{\boldsymbol{y}_i\}|\{f\}) + U(\{f\}) \quad [2.5.4]$$

Solving the MRF provides a configuration $\{\hat{f}\}$ that is optimal based on some criterion. When the MRF models an image segmentation problem, the solution $\{\hat{f}\}$ provides the optimal pixel coloring into classes.

The maximum a posteriori (MAP) configuration is a popular estimate of the most "probable" configuration of the MRF [Li 2009]. Finding the MAP requires solving the optimization problem of finding the configuration that minimizes the MAP energy $U(\{f\}|\{\boldsymbol{y}_i\})$[3]:

$$\{\hat{f}\}_{MAP} = \text{argmin}_{\{f\}}(U(\{f\}|\{\boldsymbol{y}_i\}))$$

[3] The correspondence with statistical mechanics, where low-energy configurations are the most probable is obvious [Dill and Bromberg 2002]

The likelihood energy $U(\{\boldsymbol{y}_i\}|\{f\})$ incorporates the effect of the data (the spectral image $\{\boldsymbol{y}_i\}$). The total likelihood energy of the image is the sum of the likelihood energies of its m pixels:

$$U(\{\boldsymbol{y}_i\}|\{f\}) = \sum_{i=1}^{m} u(\boldsymbol{y}_i|f_i)$$

Where $u(\boldsymbol{y}_i|f_i)$ is the likelihood energy of the i-th pixel, a function of the data $\boldsymbol{y}_i$ and the label (class).

In this study, the likelihood energy of each pixel $u(\boldsymbol{y}_i|f_i)$ is chosen as the minus natural logarithm of the refined class evidence vector $\hat{\varepsilon}_i$ of this pixel (which is a function of the data $\boldsymbol{y}_i$).

$$u(\boldsymbol{y}_i|f_i = d) = -\log(\hat{\varepsilon}_{i,d})$$

$u(\boldsymbol{y}_i|f_i)$ is a function of the discrete value label f_i that takes D values $(1,2,\dots D)$. As the evidence of a class increases (larger $\hat{\varepsilon}_{i,d}$) then the pixel likelihood energy $u(\boldsymbol{y}_i|f_i)$ decrases, therefore this label becomes more probable.

The prior energy $U(\{f\})$ is the way to model the spatial context present in the image. When $U(\{f\}) \ll U(\{\boldsymbol{y}_i\}|\{f\})$ the model includes no spatial context in the model. When $U(\{\boldsymbol{y}_i\}|\{f\}) \ll (\{f\})$ then data do not affect the MAP solution and the solution depends only on its boundary conditions and $U(\{f\})$. The balance between the two is a metric of trust in the available data. In the MRF literature, the prior energy $U(\{f\})$ is expressed via clique potentials [Li 2009], where cliques correspond to local neighborhoods of pixels compatible with the neighborhood system [Koller and Friedman 2009]: For computational feasibility reasons, practical MRF models utilize cliques of order up to two:

$$U(\{f\}) = \sum_{i=1}^{m} U_{c1}(f_i) + \sum_{i=1}^{m} \sum_{i' \in N_i} U_{c2}(f_i, f_{i'})$$

The first term corresponds to the energy contribution of cliques of size 1 (a single node) and the second term to cliques of size 2 (neighboring nodes of the graph). U_{ci} are functions (clique potentials) that provide the clique energy as a function of the labels of the nodes. N_i denotes the neighboring nodes of node i [Koller and Friedman 2009]. The energy of size 1 cliques is not critical for MRF (encodes no spatial context) and can be neglected. However, the expression for $U_{c2}(f_i, f_{i'})$ is very critical as this is the way to describe spatial context.

In this thesis, the prior energy is modeled using a standard multi-level logistic (MLL) model [Li 2009], a generalization of the famous Ising model [MacKay 2003]. The energy of cliques of size 1 is neglected. The energy of the size 2 cliques is modeled as:

$$U_{c2}(f_i = k, f_{i'} = l) = \boldsymbol{\Psi}(k,l) = \Psi_{kl}$$

where Ψ_{kl} is the element at the k-th row and l-th column of the $D \times D$ class-context matrix $\boldsymbol{\Psi}$. Ψ_{kl} contains the energy "penalty" for the case where neighboring pixels are labeled with classes k and l respectively. For simplicity, Ψ_{kl} can be written as

$$\Psi_{kl} = \begin{cases} \alpha\,, & k = l \\ \beta\,, & k \neq l \ \ and \ k - l \ compatible \\ \gamma\,, & k \neq l \ and \ k - l \ incompatible \end{cases}$$

Where α, β, γ are positive constants. Usually $\alpha > \beta$ since if a pixel belongs to class k, it is more possible that neighboring pixels belong to the same class (assuming the pixel size is smaller

than the size of the objects that correspond to the classes). The value of γ should be very large in order to prevent neighboring pixels from being labeled into incompatible classes. The diagonal elements of $\boldsymbol{\Psi}$ equal α. The non-diagonal elements are equal to either β (compatible classes) or γ (incompatible classes). The model parameters α, β are chosen in application-specific basis based on a small number of reference images. The exact value of γ does not matter, as long as it is much smaller compared to α and β.

The total MAP energy for a particular configuration of the MRF can be expressed as:

$$U(\{f\}|\{\boldsymbol{y}_i\}) = \sum_{i=1}^{m} \Phi_i(f_i) + \sum_{i=1}^{m} \sum_{i' \in N_i} \Psi_{kl}(f_i, f_{i'}) \qquad [2.5.5]$$

Where $\Phi_i(f_i) = -\log(\hat{\varepsilon}_{i,f_i})$ are the energy terms that depend just on the label of pixels, and $\Psi_{kl}(f_i, f_{i'})$ are the energy terms that depend on the configurations of pairs of neighboring pixels. This expression is written in the form of the logarithm of the joint probability of a factor graph that contains factors of order one (first term) and two (second term) and therefore its MAP can be found using the max-sum belief propagation algorithm[4] [MacKay 2003].

Max-sum Belief Propagation Algorithm

The max-sum belief propagation (BP) is an algorithm used to calculate the optimal configuration of factor graphs that contains factors of order up to 2 [MacKay 2003]:

$$U\{f\} = \sum_{i=1}^{m} \varphi_i(f_i) + \sum_{i=1}^{m} \sum_{j \in N_i} \psi_{ij}(f_i, f_j)$$

where the joint probability $P(\{f\})$ of any graph configuration $\{f\}$ equals $P(\{f\}) \propto \exp(-U\{f\})$.

The max-sum BP algorithm is based on calculating the beliefs $B_{i \to j}$ between the nodes of the graph. $B_{i \to j}$ is a $D \times 1$ vector (where D is the size of the labeling set, here the number of classes) that propagates from node i to node j. the term $B_{i \to j}(f_j)$ corresponds to the particular elemebt of vector $B_{i \to j}$ corresponding to the label f_j. $B_{i \to j}$ qualitatively describes the probability assigned by an observer in node i about the possible states in node j [MacKay 2003]. The marginal probability for the configuration of node i $P^*(f_i)$ can be expressed as a function of the believes from the neighbors N_i of node I towards i.

$$L_i = -\log(P^*(f_i)) = \varphi_i(f_i) + \sum_{j \in N_i} B_{j \to i}$$

BP was initially developed for non-loopy graphs. In this case $B_{i \to j}$ can be calculated analytically [MacKay 2003]. When BP is applied in loopy graphs, such as graphs that describe pixels in an image, believes need to be calculated using an iterative procedure until they converge [MacKay 2003; Koller and Friedman 2009]:

Min-sum algorithm

1. Initialize believes: $B_{i \to j}^{(0)}$
2. Iterate until convergence
    a. Update believes:

[4] Since the max-sum algorithm is designed to find the maximum of a potential energy, while the MAP pf the expression 2.5.3.2 corresponds to its minimum, the max-sum algorithm is applied to solve the minus of the expression 2.5.3.2.

$$B_{i\to j}^{(t+1)}(f_j) = \min_{f_i}\left\{\varphi_i(f_i) + \psi_{ij}(f_i, f_j) + \sum_{k\in(N_i\setminus j)} B_{k\to i}^{(t)}(f_i)\right\}$$

b. Keep track of optimal solution

$$\beta_{i\to j}^{(t+1)}(f_j) = \operatorname{argmin}_{f_i}\{B_{i\to j}^{(t+1)}(f_j)\}$$

c. Check convergence

3. Pick a random node i and solve (find its optimal configuration $\hat{f}_i$):

$$\hat{f}_i = \operatorname{argmin}_{f_i}\{L_i(f_i)\} = \operatorname{argmin}_{f_i}\left\{\varphi_i(f_i) + \sum_{j\in N_i} B_{j\to i}^{(T)}\right\}$$

4. Trace-back
   a. Iterate for all neighbors of solved nodes $j \in N_i$
      i. Identify the optimal configuration of node j

$$\hat{f}_J = \beta_{j\to i}^{(T)}(\hat{f}_I)$$

where the superscripts (t) correspond to the t-th iteration of the believes calculation, and the superscript (T) refers to the final result of this iteration. It needs to be commented that this is a synchronous computation scheme because each iteration calculates the believes $B_{i\to j}^{(t+1)}(f_j)$ over the whole graph based on the believes $B_{k\to i}^{(t)}(f_i)$ of the previous iteration calculated also over the whole graph.

Fast Numerical Implementation of the Belief Propagation Algorithm

Numerical calculation of the MAP solution in MRF is a very computationally expensive task. The main computation cost in the max-sum BP algorithm utilized in this thesis, originates from the calculations of the believes between nodes. Theory suggests that ordinary (synchronous: all believes are calculated simultaneously for each iteration) BP implementation requires at most as many iterations as the "dimension" of the graph $\mathcal{G}$ [Koller and Friedman 2009]. In practice it is suggested that a much smaller number of iterations is required for convergence [Tappen and Freeman 2003]. The same paper also suggests that accelerated (asynchronous) BP decreases computation time 11 times without significant reduction in estimation accuracy, making it as fast as the ordinary graph cut algorithm. Nevertheless, the calculation of the MAP solution in realistic high throughput imaging data sets is still very expensive. For example, finding the MAP in a 2000×1500 pixel spectral image that contains four labels (Chapters 3) requires several hours of CPU time in a quad-core Core i7 PC.

In this thesis, the numerical implementation of MRF is accelerated in three ways:

1. Instead of the standard synchronous calculation scheme, an "east-west-bottom-top" asynchronous scheme is implemented for calculating the believes between nodes, described in [Tappen and Freeman 2003]. For simplicity of discussion, let's consider a rectangular grid consisting of R rows and C columns. Asynchronous implementation starts by processing the rows (which can be implemented by parallel computing in order to get additional performance improvement). For each row, update the "east" believes in a progressive way: first update the belief from the node of the 1st column to the node of the 2nd column, then use this result to update the belief from the node of the 2nd column to the node of the 3rd column, and keep on until the update of the belief from the node of C-1th column to the node of the Cth column. Then, for each row (apply parallel processing) update the "west"

believes starting from the belief of the node Cth column to the node of the C-1th column, until finally the belief from the node of the 2nd column to the node of the 1st column. Then, in a similar approach calculate for each column (apply parallel processing) the "south" believes, and finally the "north" believes. This concludes a full iteration of the asynchronous BP algorithm. The 4 parts of the asynchronous BP are repeated until convergence. This approach seems to accelerate the convergence of believe calculation because asynchronous updates favor faster information travel within the graph.

2. The initialization of the belief $B_{i\to j}^{(0)}$ is not random, but it is chosen to equal the refined evidence vector $\hat{\boldsymbol{\varepsilon}}_j$ of the image pixel that corresponds to node j, since this provides a relatively good initial estimate about the probability if the possible configuration f_j (classes) of that pixel.
3. Perform global segmentation by MRF only in those parts of the image where local segmentation performance is under question. Image analysis data provided below show that local image segmentation can identify quite well the bulk of regions that correspond to different classes. There is no reason to include the core of these solid regions in the MRF model, since they will increase the computation cost without a gain in segmentation accuracy since it is almost certain MRF will provide identical segmentation as the local segmentation algorithm. Instead, the parts of the image that are included in the graph $\mathcal{G}$ of the MRF model are:
   - Pixels around the interface between neighboring regions classified in different classes. These pixels G are calculated using a morphological operation on the binary array $\boldsymbol{B}_l$ provided by local image segmentation:

$$G = \bigcup_{d=1}^{D} \bigcup_{d'=1}^{D} \delta_r(\boldsymbol{B}_{l,d}) \cap \boldsymbol{B}_{l,d'}$$

     Where $\cap$, $\cup$ are the intersection and union operators, $\delta(\cdot)$ is the dilation operator using as structural element a disk of radius r [Soille 2003], and $\boldsymbol{B}_{l,d}$ is the d-th plane of the binary local segmentation array $\boldsymbol{B}_l$. By selecting the radius r of the dilation operator it is possible to control the thickness of the image around the region interface that is included in the MRF model.
   - Pixels that were assigned to a particular class even though the evidence of this class was relatively low.

   The resulting graph $\mathcal{G}$ that corresponds to the pixels that satisfy these criteria may not be connected. Instead usually it consists of several disconnected regions, each of which corresponds to a smaller graph. For computational reasons it is not advantageous to process graphs of very small or very large size. It is advantageous to combine several disconnected graphs into an equivalent graph (a "forest") of appropriate size consisting of several disconnected graphs ("trees") and solve them using the same MRF. For the particular MATLAB implementation of asynchronous BP, the optimal size of the forest that provided the best performance (2.1 msec/pixel) is approximately 2E4 – 2E5 nodes in a system of 4 classes.

   The "known" nodes $\mathcal{K}$ that are not included in the MRF model and have known labeling provided by the local segmentation algorithm. The known nodes act as boundary conditions

that affect the global segmentation. This needs to be incorporated into the MAP energy of the forest in a way that will

The nodes of the MRF forest $\mathcal{G}$ are partitioned into two core nodes $\mathcal{C}$ (not in contact with the "known" nodes $\mathcal{K}$), and boundary nodes $\mathcal{B}$ (in contact with at least one "known" node so that the intersection $N_i \cap \mathcal{K}$ is non-empty). The total MAP energy for a particular configuration of the forest that incorporates the effect of the boundary nodes is:

$$U(\{f\}|\{\boldsymbol{y}_i\}) = \sum_{i \in \mathcal{C}} \Phi_i(f_i) + \sum_{ij \in \mathcal{B}} \left(\Phi_i(f_i) + \sum_{i' \in (N_i \cap \mathcal{K})} \Psi_{kl}(f_i, f_{i'}) \right) + \sum_{i \in \mathcal{G}} \sum_{i' \in (N_i \setminus \mathcal{K})} \Psi_{kl}(f_i, f_{i'})$$

The first two terms are a function only of the label f_i of a single pixel. The first term corresponds to pixels that belong to the core nodes $\mathcal{C}$. The second term corresponds to boundary pixels $\mathcal{B}$. It suggests that in these pixels Φ_i needs to be modified in order to incorporate the effect of the boundary cells. The third term is a function of pairwise labels $f_i, f_{i'}$of two neighboring pixels that belong to the forest $\mathcal{G}$ (neighbors of forest pixels that belong to the known nodes are not included anymore). This expression is also written in the form of logarithm of the joint probability of a factor graph that contains factors of order one (first two terms) and two (third term) and therefore its MAP can be found using belief propagation [MacKay 2003]. The only difference compared to the standard MRF practice that utilizes the whole image is that the first order factors of the boundary nodes $\mathcal{B}$ need to contain the term $\sum_{i' \in (N_i \cap \mathcal{K})} \Psi_{kl}(f_i, f_{i'})$ in addition to the term $\Phi_i(f_i)$.

2.5.4 Segmentation Performance

This section provides simulation results that quantify the performance of the image segmentation algorithms described in Section 2.5. The key step for image segmentation is local segmentation, because it provides a coarse partition of the image into regions dominated by particular classes. The downstream global image segmentation algorithm aims in refining the outcome of local segmentation. Furthermore, when local image segmentation is successful in identifying regions dominated by particular classes, the number of pixels required to be refined by global image segmentation decreases, and image segmentation is much faster.

Performance Metrics

Each pixel in the image is assumed to belong to one particular class. If pixel i belongs to class d, the detected signal $\boldsymbol{y}$ at this pixel is a random vector due to the stochastic nature of the emission rates $\boldsymbol{\lambda}_{0,d}$ of the N_d sources, and the stochastic nature of photon arrival at the detector (particularly at low signal levels).

Based on the measurement $\boldsymbol{y}$ at some image pixel, the probability that this pixel belongs to domain $d = 1,2,\ldots,D$ is described the evidence vector $\boldsymbol{\varepsilon}_i$ described by Equations 2.5-1 to 2.5-7 of Section 2.5.2. Local image segmentation assigns pixel i to a class based on a decision rule (e.g. the MAP rule: $d = \text{argmax}_d\, \varepsilon_{i,d}$). The accuracy of this decision depends on three factors:

- How "similar" are classes: the degree of similarity between two classes depends on how similar are the spectral signatures and emission rate vectors of their emission sources
- The uncertainty in the knowledge of the emission rates of the sources. Class evidence calculations (Equations 2.5-3 to 2.5-6) depend on the limits $[\lambda_{d,i}^L, \lambda_{d,i}^U]$ defined by the user,

which express the approximate knowledge on the range of values of the rates of the sources $\lambda_{d,i}$ of each class. These limits are approximations of the actual (unknown) pdf of $\lambda_{d,i}$.

- Pixel signal cross-talk. This describes the signal detected at pixel i contains some contributions of signal from surrounding pixels. When this contribution is significant, significant decision errors can be induced.

Local image segmentation is a classification problem. Here, the accuracy of local image segmentation is quantified by the following two metrics:

- The classification error matrix $\mathcal{E}$ is a $D \times D$ matrix, whose (i,j) element $\mathcal{E}_{i,j}$ contains the probability that a pixel belonging to class i will be classified the local segmentation algorithm as belonging to class j.
- The sensitivity SE_d and specificity SP_d of identifying pixels of class d are defined as

$$SE_d = \frac{TP_d}{TF_d + FN_d} = \frac{\mathcal{E}_{d,d}}{\sum_{t=1}^{D} \mathcal{E}_{d,t}} \quad , \quad d = 1,2,\dots D$$

$$SP_d = \frac{TN_d}{FP_d + TN_d} = \frac{\sum_{i \neq d}\{\sum_{j \neq d} \mathcal{E}_{i,j} \cdot p_D(i)\}}{\sum_{i \neq d}\{\sum_{j \neq d} \mathcal{E}_{i,j} \cdot p_D(i) + \mathcal{E}_{i,d}\}} \quad , \quad d = 1,2,\dots D$$

Where $p_D(d)$ is the a-priori probability of class d, TP_d is the "true positive" probability for class d (a pixel belonging to class i will be classified as class i), FN_d is the "false negative" probability of d (a pixel belonging to class i will be classified as class $j \neq i$), FP_d is the "false positive" probability of d (a pixel belonging to class $j \neq i$ will be classified as class i), and TN_d is the "true negative" probability of d (a pixel belonging to any class $j \neq i$ will be classified as any class $j \neq i$). The mean sensitivity and specificity $\overline{SE}$ and $\overline{SP}$ can be calculated as:

$$\overline{SE} = \sum_{d=1}^{D} SE_d \cdot p_D(d)$$

$$\overline{SP} = \sum_{d=1}^{D} SP_d \cdot p_D(d)$$

A locus $(SE_d(x), 1 - SP_d(x))$ where x is some parameter that affects the calculation of the evidence vector $\boldsymbol{\varepsilon}_i$, defines a receiver operating characteristic (ROC) curve for domain [Wornell 2010]. SE_d describes the probability that image segmentation assigns a pixel to class d when the pixel indeed belongs to class d. $1 - SP_d(x)$ describes the probability that image segmentation assigns a pixel to class d when the pixel does not belong to class d.

Case Study

It is not possible to derive analytical expressions for SE_d and SP_d that are valid in any imaging case. The performance of the local classifier and its sensitivity to key parameters needs to be evaluated for each imaging case separately. In the following sub-sections, the performance of local image segmentation is evaluated the case of the cell-seeded scaffolds data-set discussed in Section 2.6.2 and Chapter 3. The system classes, sources and source rates bounds (actual and assumed assumed) are summarized in Table 2.5-1. The class map is shown in Fig. 2.6.3.The system consists of four classes. The class "void" contains only instrument noise and describes empty space. Class "nucleus" and class "cytoplasm" contain two sources each and have a common source (CMTMR). Here, the class "scaffold" contains a single source (collagen fluorescence) whose spectral signature is similar to the signature of Hoechst 33342 present in the "nucleus" class (Figure 2.4.3).

The emission rates of the sources of each class are chosen so that the classes "nucleus", "cytoplasm" and "scaffold" have approximately equal mean emission that is SNR times larger

than the emission rate of the background. The parameter V is used to increase the range of the assumed emission rate of the sources of the "cytoplasm" class used in evidence calculations.

Class	Emission sources	Actual emission rate bounds $[\lambda_{d,i}^L, \lambda_{d,i}^U]$	Assumed emission rate bounds $[\lambda_{d,i}^L, \lambda_{d,i}^U]$
void	Instrument noise	0.2	0.2
nucleus	CMTMR	[SNR×0.33, SNR×1.7]	[SNR×0.33, SNR×1.7]
	Hoechst 33342	[SNR×0.33, SNR×1.7]	[SNR×0.33, SNR×1.7]
cytoplasm	CMTMR	[SNR×0.33, SNR×1.7]	[SNR×0.33, SNR×1.7×V]
	Alexa fluor 488	[SNR×0.33, SNR×1.7]	[SNR×0.33, SNR×1.7×V]
scaffold	Collagen	[SNR×0.04, SNR×4]	[SNR×0.04, SNR×4]

Table 2.5-1: Parameters of simulations for the evaluation of local image segmentation performance presented in Section 2.5.4. Emission rates are expressed in photons/pixel. "Actual emission rate bounds" are used to generate the simulated data. "Assumed emission rate bounds" are used in the evidence calculations described in Section 2.5.2.

The Effect of Class "Similarity"

Correct classification of pixels in the appropriate class depends on the degree of similarity of different classes. Different classes can be distinguished based on the spectral signatures and the emission rates of their sources. The larger the difference between the spectral signatures or the emission rates of the sources of different classes, classification should be more accurate (larger sensitivity SE_d and specificity SP_d).

Figure 2.5.1 shows how the performance of the local image segmentation algorithm varies as a function of the SNR parameter (see Table 2.5-1). When SNR is low, the measured data $\boldsymbol{y}$ corresponding to the "nucleus", "cytoplasm", and "scaffold" classes become very similar to the emission of the "void" class. Results show that as SNR decreases, then the sensitivity SE_d of "nucleus", "cytoplasm", and "scaffold" classes and the specificity SP_d of "void" class decrease. The classification error matrices also show that this is because pixels of "nucleus", "cytoplasm", and "scaffold" classes are classified as "void". The performance of the local segmentation algorithm provides at least 80% sensitivity and specificity when SNR>40. Given that the mean pixel noise of the instrument utilized is approximately 0.2 photons/pixel, SNR>40 corresponds to signal of at least 8 photons/pixel, suggesting that the algorithm can perform well even in very low signal levels.

Similar analysis can quantify how ROC curves vary as the mean level of emission at different classes caries. The results shown in Figure 2.5.1 correspond to classes that have equal mean emission. It is expected that as the difference between the mean emission of classes increases, the performance of the classifier will also increase.

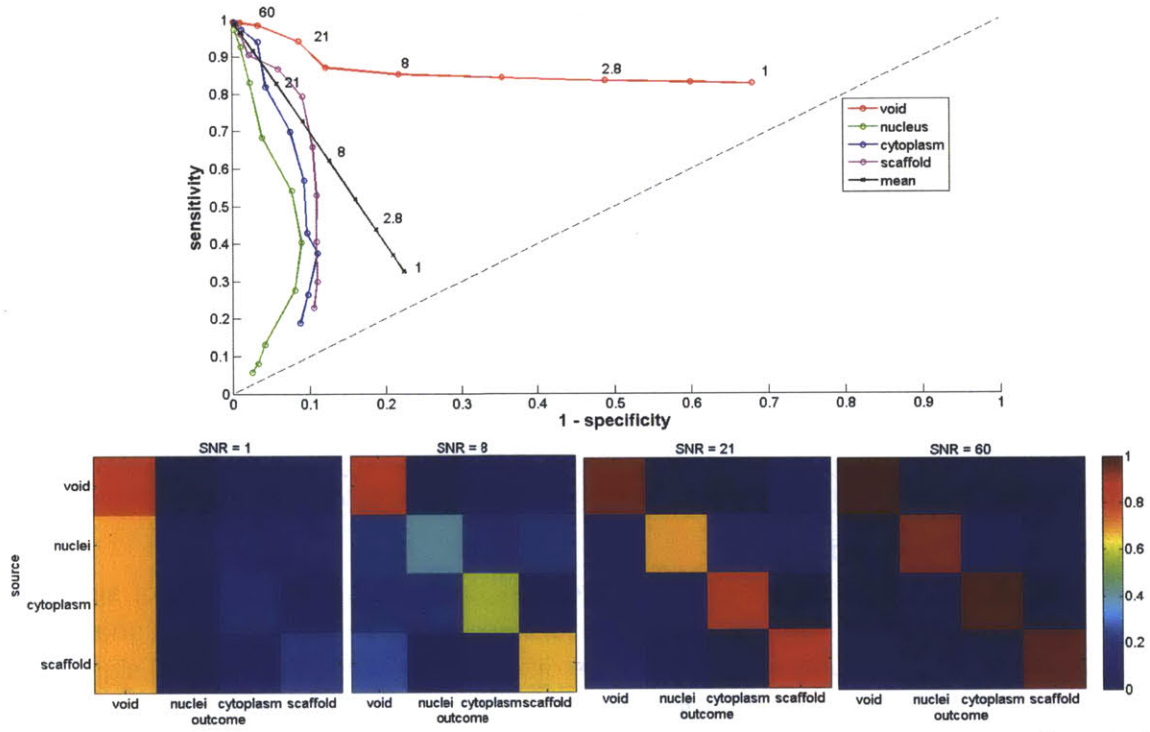


Figure 2.5.1: Performance of the local image segmentation algorithm as a function of the signal to noise ratio (SNR) when there is accurate knowledge of the source emission rates and there is no pixel signal cross-talk. **Top**: ROC curves for the four classes as a function of SNR. **Bottom**: Classification error matrix **E** for four various values of SNR. Results are based on 10000 random data-points per case.

The Effect of the Uncertainty of Class Source Emission

Bayesian classification and model selection schemes are known to be sensitive to their hyper-parameters that describe prior knowledge on the system parameters [Kass and Rafterry 1995]. In the local segmentation approach described in Section 2.5.2, prior knowledge is described by the emission rate bounds $[\lambda_{d,i}^{L}, \lambda_{d,i}^{U}]$. The results of Figure 2.5.1 were obtained assuming perfect knowledge of these bounds. Is it of interest to describe how the performance of the local image classifier varies as imperfect knowledge of these bound increases. In the model described in Table 2.5-1, this uncertainty is modeled using the parameter V. When $V = 1$ then there is perfect knowledge of the bounds. When $V < 1$ or $V > 1$ then the upper limit of the emission rates for the "cytoplasm" class is underestimated or over-estimated respectively.

Figure 2.5.2 shows how the performance of the local image segmentation algorithm varies as a function of the parameter V). When V deviates away from 1, the specificity of the "cytoplasm" class and the sensitivity of the "nucleus" class decrease. At the same time the classification performance of "void" and "scaffold" pixels remains unchanged. Results suggest that as V deviates away from 1 more "cytoplasm" pixels get misclassified as "nucleus" pixels, probably because "nuclei" and "cytoplasm" classes have relatively similar spectral signatures and mean emission rates (both contain the CMTMR source). Results show that very large errors in the knowledge of the bounds of the sources of the "cytoplasm" (underestimate 4 times or overestimate 10 times) induce classification errors (decrease specificity of "cytoplasm" by 30%). Reasonable errors (underestimate or overestimate 2 times) cause reasonable classification errors (decrease specificity of "cytoplasm" by <10%).

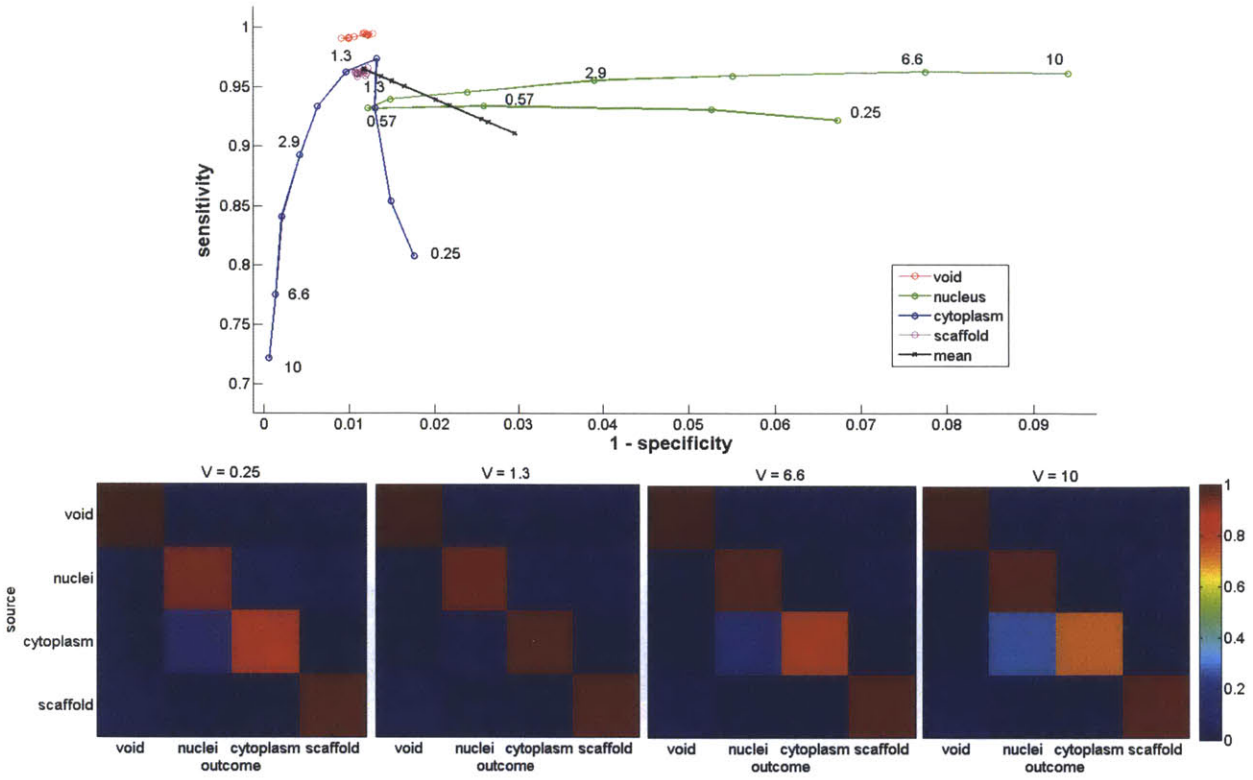


Figure 2.5.2: Performance of the local image segmentation algorithm as a function of uncertainty in theemission rates of "cytoplasm" class when there is no pixel signal cross-talk, and SNR=60. **Top**: ROC curves for the four classes as a function of parameter V. **Bottom**: Classification error matrix E for four various values of V. Results are based on 10000 random data-points per case.

The Effect of Class Signal Mixture

The results presented in Figure 2.5.1 and Figure 2.5.2 are based on simulations where the simulated data $\boldsymbol{y}$ for each pixel contain signal from a single class. In practice, due to the geometry of light focusing, the signal detected at pixel i contains contributions from sources located at pixels in the proximity of pixel i. This means that measured data $\boldsymbol{y}$ contains contributions of various classes (pixel cross-talk signal), which can deteriorate the performance of the local segmentation algorithm.

Figure 2.5.3 shows how the performance of the local image segmentation algorithm varies as a function of the fraction of the signal that is cross-talk from neighboring "cytoplasm" pixels. As the fraction of cross-talk signal increases, the sensitivity of the "void" and "scaffold" classes and the specificity of the "cytoplasm" class decrease. The response of the ROC for the "nucleus" class is more complex, small fraction of cross-talk "cytoplasm" signal decrease "nucleus" class specificity, larger fractions decrease "nucleus" class sensitivity. Results show that the "cytoplasm" cross-talk signal causes void pixels (whose emission is much weaker that "cytoplasm" pixels) to be misclassified as "cytoplasm" pixels. Results also show that the "cytoplasm" cross-talk signal causes "scaffold" pixels to be misclassified as "nucleus pixels", probably because mixing the spectral signatures of "scaffold" sources (blue-green) and "cytoplasm" sources (green-orange) forms spectral signatures similar to "nucleus" sources (blue and orange). Such simulations suggest that local image segmentation algorithms may not classify accurately pixels located close to the boundaries of class regions. However, as long as

the local image classifier achieves to classify accurately the bulk of the region pixels, then the pixels located around region boundaries will be classified using the global image segmentation algorithm.

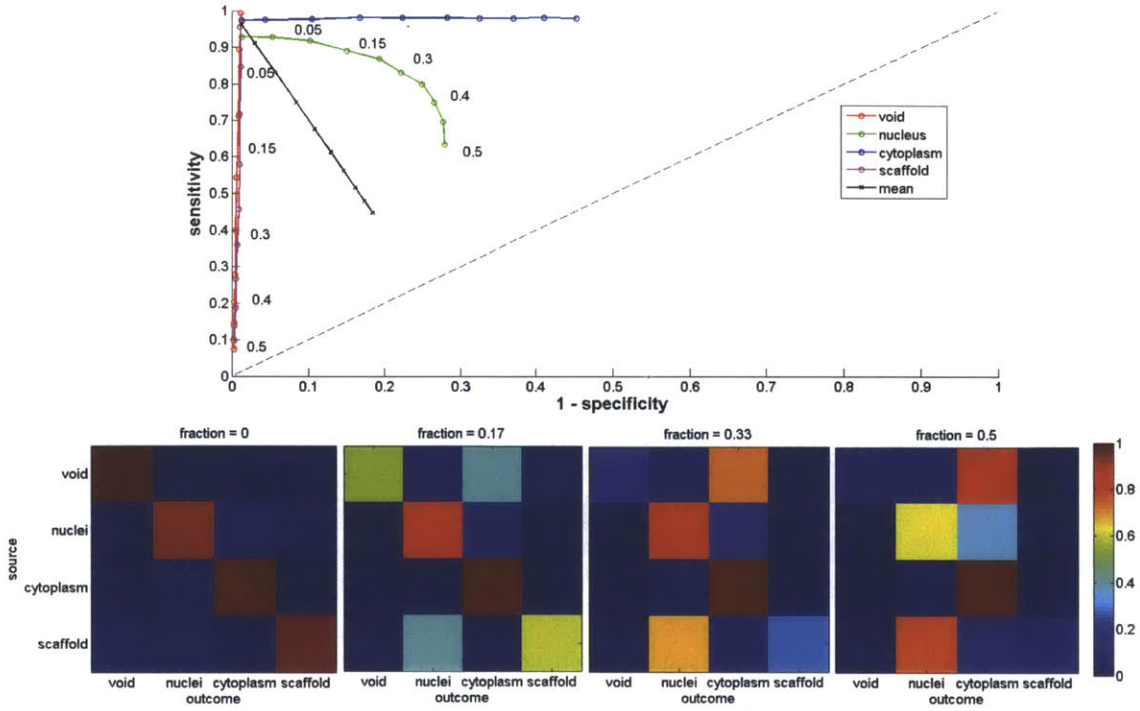


Figure 2.5.3: Performance of the local image segmentation algorithm as a function of pixel signal cross-talk from the "cytoplasm" class when there is accurate knowledge of the source emission rates and SNR=60. **Top**: ROC curves for the four classes as a function of the fraction of signal that corresponds to "cytoplasm". **Bottom**: Classification error matrix E for four various values of the fraction of signal that corresponds to "cytoplasm". Results are based on 10000 random data-points per case.

2.6 Challenging Applications of Spectral Multi-photon Imaging

Section 2.6 provides applications of the image processing pipeline described in this Chapter in the analysis of experimental datasets acquired by the spectral multi-photon microscope described in section 2.3. Each one of the three samples presents a different set of image processing challenges. The last two applications (Sections 2.6.2 and 2.6.3) are based on the imaging studies related to peripheral nerve regeneration described in Chapters 3 and 4.

2.6.1 *Imaging Endogenous Molecules in P. Aeruginusa*

P. Aeruginusa is a bacteria strain that forms colonies in patients that suffer from cystic fibrosis. Bacteria colonies are much more resistant to antibiotics and treatments compared to single isolated bacteria. Several critical functions of *P. Aeruginusa* (including cell-cell signaling, electron harvesting and biofilm formation) are modulated by phenazines, a family of endogenous small molecules [Price-Whelan et al. 2006; Mavrodi et al. 2006; Dietrich et al. 2008]. Phenazines, including PYO and PCA, are fluorescent when they are reduced, which can be exploited for their quantification [Sullivan et al. 2011]. Since existing methods for studying the redox state of small molecules are tedious and provide no spatial resolution and low temporal resolution, it is of interest to exploit fluorescence imaging to study the redox state of these molecules with high spatiotemporal resolution *in vivo*.

Samples of wild type and mutant *P. Aeruginusa* cells were prepared by Dr. N. Sullivan (Newman Lab, Department of Biology, MIT). 1 ml cell aliquots from overnight 100ml cultures were pelleted. Bacteria were suspended in PBS and 100 µl samples were transferred inside glass wells, sealed by a #1.5 coverslip. Cell samples were imaged every 15 min sampling time for 2h after sealing [Sullivan et al. 2011].

The *P. Aeruginusa* imaging data set consists of 2D images of *P. Aeruginusa* bacteria in solution at various time points after transferring bacteria from O_2–rich to O_2–limited conditions. The major challenge in this dataset is the need to estimate the concentration of multiple endogenous weak fluorophores whose emission spectra lie within a 200nm window of the EM spectrum. The dataset contains 2 classes (cells, medium). The "class map"[5] of the system is shown in Figure 2.6.1. The "cell" class describes pixels that correspond to bacteria and contains between 1 (just NADH) and 4 emission sources (NADH, PVD, PCA, PYO) in the wild type strain that expresses both phenazines (PYO, PCA) and the siderophore PVD. The "solution" class describes pixels that correspond to the medium that surround bacteria, and the emission sources of the "cell" class apart from NADH and PVD that are not secreted. This fluorophore set is hard to unmix, as described by the data shown in Figure 2.4.16 and Figure 2.4.17.

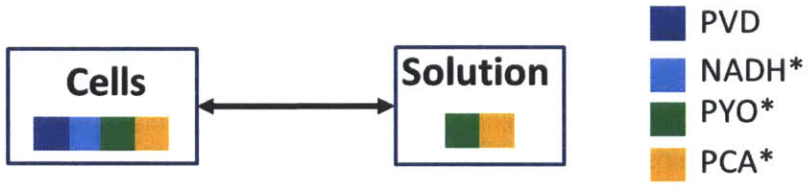


Figure 2.6.1: Class map for P. Aeruginusa imaging for the wild type strain that expresses both PCA and PYO phenazines. Asterisk (*) refers to a molecule that is fluorescent only when in reduced state.

Figure 2.6.2 shows an application of the developed methodology in quantifying the temporal response of reduced phenazines, reduced NADH and PVD after *P. Aeruginusa* are transported from an oxygen-rich to an oxygen-limited environment. This application provided the first estimation of reduced phenazine concentration *in vivo*, and the first measurement of the dynamics of the phenazine-PVD-NADH system [Sullivan et al. 2011].

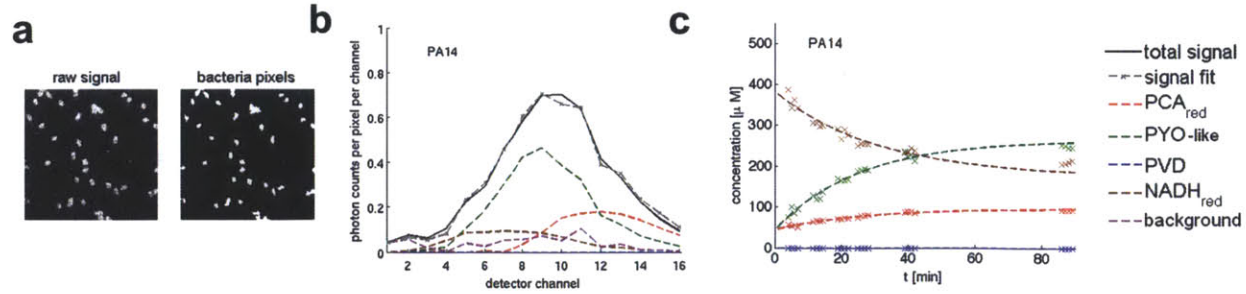


Figure 2.6.2: Quantifying the dynamics of reduced phenazine concentraton in P. Aeruginusa after a sudden move from an oxygen-rich to an oxygen-limited environment by spectral multiphoton microscopy. **a**: Left: representative images of bacteria (all 16-channels have been added to form a grayscale image (raw). The signal level is on the order of 5 photons per pixel. Right: Pixels classified as "cells". **b:** spectral unmixing of the fluorescence emission for the sum of the signal in pixels belonging to the "cell" class. **C:** time response of the concentrations of reduced

[5] The "class map" is a schematic that shows the classes present in an image (shown as boxes), the emission sources present in each class (colors inside the class box) and the valid neighborhood relationships between classes (shown as arrows).

phenazines PYO* and PCA*, the siderophore PVD and reduced NADH within 90 minutes after oxygen depletion. Data are obtained from [Sullivan et al. 2011].

2.6.2 *Imaging Human Dermal Fibroblasts inside Porous Collagen Scaffolds*

Image informatics is an imaging-based method for studying complex signal transduction pathways inside cells [Perlman et al. 2004; Nir et al. 2010]. Information about the pathway of interest is inferred by studying how the cell response to particular stimuli changes when one or multiple parts of the signal transduction pathway are perturbed, see Chapter 3. The response of each experiment is described by a series of imaging-based features that quantifies each cells present in the dataset.

This dataset is part of an image informatics study of the SMAD-mediated TGFβ-induced differentiation of fibroblasts inside porous collagen scaffolds. Lentiviral shRNA was used to knock down particular genes of the TGFβ-SMAD pathway in primary human dermal fibroblasts (HDF). Transfected cells were seeded inside porous collagen-GAG (CG) scaffolds similar to the ones used clinically for skin regeneration [Yannas et al, 1982, 1989]. Cells were incubated at 37°C, 5%CO_2 for five days in the presence of TGFβ isoforms, and then fixed, immune-stained and imaged. More information about this experiment can be obtained in Chapter 3.

The data set consists of four classes: cell nucleus (nuc), cell cytoplasm (cyto), scaffold (scaf) and void (the empty space between cells and scaffold struts). The class map of the system is shown in Figure 2.6.3. The spectral unxmixing performance of each class is expected to be very good, since each class contains just 2 sources that are quite well separated.

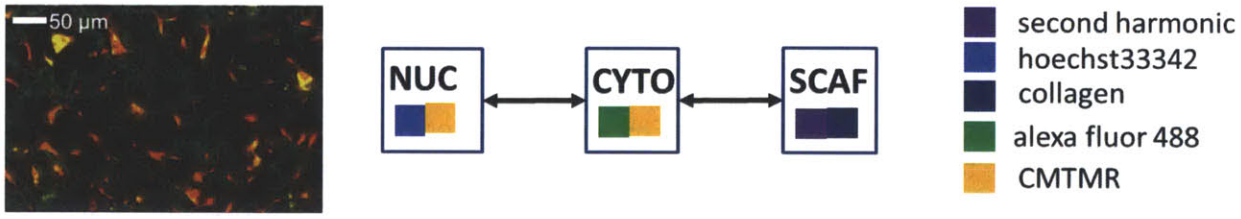


Figure 2.6.3: Class map for imaging cell-seeded scaffolds. **Left**: Spectral image of fibroblasts seeded inside the scaffold after spectral unmixing. Red: cells, green: scaffold.

Results on the application of the developed pipeline to process the images of the cell-seeded scaffolds of the image informatics study are provided in Chapter 3.

2.6.3 *Imaging Ex Vivo Samples of Injured Peripheral Nerves*

Multi-photon imaging is a promising imaging modality for *in vivo* imaging of small animals, due to its key features (large penetration depth, low photo-toxicity). In this dataset, the unique properties of multi-photon excitation are exploited for acquiring information-rich images of *ex vivo* rat peripheral nerves (PN) samples obtained from a peripheral nerve regeneration animal model [Soller et al. 2012]. The objective of this study is to provide tissue and cellular level description of how biomaterials of different properties affect the kinetics and final outcome of wound healing in severely injured PN. More information and results about this experiment is available in Chapter 4.

Scaffold fabrication, animal experiments, and nerve tissue harvesting, fixation and sectioning was conducted by Dr E. Soller (Yannas Lab, Department of Mechanical Engineering, MIT). After transecting the sciatic nerve in rats and separating the resulting stumps by 15 mm (a significant wound that cannot regenerate by itself), the two stumps are entubulated by a collagen tube and

let heal for 1, 2, or 9 weeks. The harvested nerves are sections, immunolabeled for various antigens of interest and imaged. In this study, the ability of multiphoton excitation to excite simultaneously multiple fluorophores and to distinguish D-banded versus non-banded collagen based on the emission of second harmonic scattering, are exploited in order to obtain information-rich images of the nerve regenerates with minimal sample usage.

Based on nerve anatomy, and the staining protocol applied to the sample and the localization of the corresponding antigens (Chapter 4), the class map of the system is shown in Figure 2.6.4. It consists of ten classes "cell nucleus" (nuc), "perineurium" (prn), "blood vessels" (bvs), "myofibroblasts" (mfb), "fibroblasts" (fbc), "schwann cells" (scc), "axons" (axn), "endoneurial matrix" (end), "banded collagen" (col) and "collagen scaffold" (scf). Each class contains between 1 to 3 sources. Therefore, even though there are 7 emission sources in the system, 3 of which have very similar spectral signature (dapi, collagen autofluorescence, alexa fluor 350), spectral unmixing in each class is expected to be accurate because the emissions of each class have quite distinct spectral signatures.

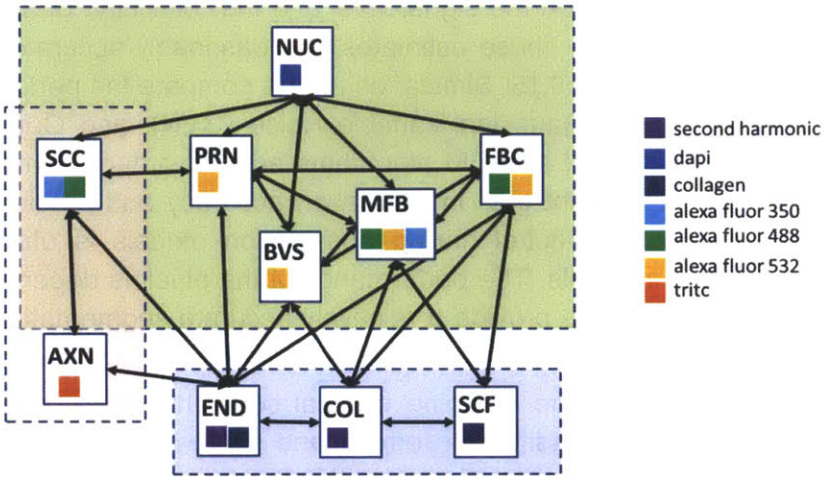


Figure 2.6.4: Class map for the *ex vivo* nerve imaging dataset.

Results on the application of the developed pipeline to process the images of the *ex vivo* nerve imaging dataset are provided in Chapter 4.

2.7 Summary and Future Work

This chapter provides an image processing pipeline for large data-sets of spectral microscopy images of complex biological samples. The developed image processing pipeline is a key enabling technology for extensive application of spectral fluorescence microscopy in biological research. The wide-spread use of fluorescence microscopy in biological research, lead to the development of many image processing software tools, some proprietary and available commercially (e.g. Imaris), others open-sourced and available at no cost (e.g. BioImageXD, CellProfiler, FARSIGHT, Fiji, ImageJ) [Eliceiri et al. 2012; Hamilton 2012; Kankaanpää et al. 2012; Schindelin et al. 2012]. These software tools include a wide range of features: microscope control and data acquisition, image visualization, image processing (segmentation, classification), image quantification, particle tracking, and statistical analysis. Nevertheless none of these tools has been designed in a way that exploits the high information content present in spectral fluorescence microscopy. Only a few of them provide basic spectral unmixing calculations [Hamilton 2012]. The generic pipeline described in this Chapter can act as the first

image processing step whose outputs can be further analyzed by the available image processing tools developed to process 3D fluorescence images. Although all application described in this thesis were acquired by multi-photon microscopy, the analysis and algorithms described in this thesis can be applied also to spectral confocal (single-photon excitation) microscopy.

The pipeline consists of two main parts: a local image segmentation scheme and a global image segmentation scheme. By far the most important part of the pipeline is the local image segmentation, which treats pixel-wide image segmentation as a model selection problem. The local segmentation scheme requires D spectral unmixing and D class evidence computations per pixel (D is the number of pixel classes present in the image), making essential the efficient and accurate implementation of spectral unmixing and class evidence calculations. This chapter provides analytic estimates of the estimation error of spectral unmixing as a function of the set of sources present in a pixel, their emission rates and modeling/calibration errors. The validity of these expressions depends strongly on the signal level and the "difficulty" of spectral unmixing quantified by the variance factors G_i. These estimates are reasonably accurate for signal level larger than 30 photons/pixel and $G_i > 0.15$. Simulation results compare the performance of three spectral unmixing algorithms (EM, Gaussian Variable Noise (GVN), and Gaussian Common Noise (GCN)) and suggest that GVN and EM algorithms are 1-2 orders of magnitude slower than the basic GCN without significant gain in estimation accuracy and precision even at low signal images. The convexity of spectral mixing observation models is utilized in efficient calculations of class evidence integrals. The performance of the pipeline depends on the ability of the local segmentation algorithm to provide a relatively accurate segmentation of the image, specifically identify coarsely the regions that correspond to different objects.

Global image segmentation refines the outcome of local segmentation to provide regions of smooth boundaries, remove pixel classification errors, and prevent incompatible classes from coming to proximity. Global image segmentation algorithms are very computationally expensive, prohibiting their application in large data sets. Here, a fast global image segmentation approach was implemented by including only parts of the image (boundaries of regions provided by the local image segmentation, pixels where incompatible classes meet) in a Markov Random Field model, and then solving the model by an efficient belief propagation numerical scheme. The computation cost (per pixel) of global segmentation is approximately 2 orders of magnitude higher than the computation cost of spectral unmixing, and 1 order of magnitude higher than the cost of local segmentation.

The developed image processing pipeline has three key features:

- enables high-content fluorescent imaging that is not feasible using standard fluorescence microscopy instrumentation and processing. The imaging applications presented in Section 2.6 and Chapters 3, 4 are beyond the published state-of-the-art in terms of information content (total number of classes, total number of sources) per image acquisition (Section 2.2.3). However, the analysis of the image processing pipeline presented in this Chapter suggests that high-content high-accuracy imaging can be achieved only when the staining protocol and the imaging acquisition parameters are chosen appropriately to ensure that i) neighboring classes are significantly "different", ii) the sources present at each class have significantly different spectral signatures but similar range of emission rates.

- Can be applied to process large data sets obtained by high-throughput imaging due to the utilization of algorithms that are computationally efficient and can be run in parallel processing mode.
- Can be utilized by users that are not expert in optics or signal processing. Specifically, the inputs required by the user are parameters that either depend on the nature of the sample and the staining protocol (number of classes, sources per class), or can be easily estimated after imaging (coarse bounds on source emission rates).

The developed pipeline can be extended and improved in several ways

- The code of the image processing pipeline was implemented in the MATLAB scientific computing environment. Although MATLAB is an efficient environment for rapid prototyping of software tools, its numerical performance is limited and it is not open-sourced. A much more efficient version of the code would be implemented in a more efficient programing language such as C or Java. Such an implementation would increase its outreach to the biological/medical research community either as a standalone application or as a plugin to ImageJ, favoring wide use of spectral fluorescence microscopy.
- Several image computation tasks can be done in parallel. This enables their implementation in way that can make use of the massive parallel processing power of graphics processing units (GPU) cards available commercially. Implementations based on multi-core GPUs could improve the speed of all computation-intensive tasks (spectral unmixing, evidence calculations, belief propagation) by at least one order of magnitude.
- The image content of fluorescence imaging can be further increased by combining spectral microscopy with fluorescence life-time imaging (FLIM). High-sensitivity spectral-FLIM imaging can be implemented using MA-PMTs whose outputs are processed by an array of constant-fraction discriminator and counter circuits [Becker 2005]. A large fraction of the analysis and algorithms described in this Chapter can be applied to analyze Spectral-FLIM fluorescence microscopy images. Spectral-FLIM fluorescence imaging could enable accurate 3D imaging of very challenging fluorophore sets (such as the phenazine set described in Section 2.6.1) that are hard to image using spectral imaging alone, and could improve significantly the accuracy and precision of the estimation of the emission rates of the sources present in fluorescence images.
- The implementation of spectral fluorescence imaging in deep tissue imaging *in vivo* may require modification of the image processing algorithms, or implementation of optical tricks to handle the deteriorating effects of emission light scattering in the spectral response.

Accelerating the required computations of the proposed image processing pipeline aims towards implementation of the pipeline in real time. In this case, the developed pipeline can be embedded inside the software that controls microscope acquisition, leading to the generation of "smart" microscopes. Such instruments could enable high-throughput high-content imaging of large complex biological samples, for example tissue cytometry [Kim et al. 2007b, Ragan et al. 2012], with minimal user input.

2.8 Acknowledgements

The author would like to thank Georgios Papachristoudis (PhD candidate, EECS, MIT) for his valuable and kind contribution of the first implementation of the max-sum belief propagation algorithm in MATLAB.

2.9 Literature Cited

Bai J, Xiang S, and Pan C. A Graph-Based Classification Method for Hyperspectral Images. *IEEE Trans on Geoscience and Remote Sensing* **51**(2): 803-17, 2013. (doi: 10.1109/TGRS.2012.2205002).

Bates M, Huang B, Dempsey GT, Zhuang X. Multicolor super-resolution imaging with photo-switchable fluorescent probes. *Science* **317**(5845): 1749-53, 2007.

Becker W. *Advanced Time-Correlated Single Photon Counting Techniques*, Springer, 2005.

Berg RH, Evaluation of spectral imaging for plant cell analysis, *J. Microscopy* **214**(2): 174-81, 2004.

Berry MW, Browne M, Langville AN, Pauca VP, and Plemmons RJ. Algorithms and applications for approximate nonnegative matrix factorization. *Comput. Stat. Dat. Anal.* **52**(1): 155-173, 2007.

Besag J. Spatial Interaction and the Statistical Analysis of Lattice Systems, *J. R. Stat. Soc. Ser. B* **36**:192-236, 1974.

Besag J. On the Statistical Analysis of Dirty Pictures, *J. R. Stat. Soc. Ser. B* **48**: 259-302, 1986.

Bestvater F, Spiess E, Stobrawa G, Hacker M, Feurer T, Porwol T, Berchner-Pfannschmidt U, Wotzlaw C, and Acker H, Two-photon fluorescence absorption and emission spectra of dyes relevant for cell imaging, *J. Microsc.* **208**(2): 108-15, 2002.

Betzig E, Patterson GH, Sougrat R, Lindwasser OW, Olenych S, Bonifacino JS, Davidson MW, Lippincott-Schwartz J, and Hess HF. Imaging intracellular fluorescent proteins at nanometer resolution. *Science.* **313**(5793): 1642-5, 2006.

Bilmes JA. A Gentle Tutorial of the EM algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models, *Technical Report* TR-97-021, International Computer Science Institute, 1998.

Bird DK, Eliceiri KW, Fan C-H, and White JG. Simultaneous Two-Photon Spectral and Lifetime Fluorescence Microscopy. *Applied Optics* **43**(27): 5173-82, 2004.

Bjornsson CS, Lin G, Al-Kofahi Y, Narayanaswamy A, Smith KL, Shain W, and Roysam B. Associative image analysis: a method for automated quantification of 3D multi-parameter images of brain tissue. *J Neurosci Methods* **170**(1): 165-78, 2008. (doi: 10.1016/ j.jneumeth.2007.12.024).

Boardman JW, and Kruse F.A. Automated spectral analysis: A geologic example using AVIRIS data, north Grapevine Mountains, Nevada: in Proceedings, *10th Thematic Conf. on Geologic Remote Sensing*, Environmental Research Institute of Michigan, Ann Arbor, MI, p. I-407-18, 1994.

Born M, and Wolf E, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, Seventh Edition, Cambridge University Press, 1999.

Boyd S, and Vandenberghe L. *Convex Optimization*, Cambridge University Press, 2004.

Boykov Y, Veksler O., Zabih R., Fast approximate energy minimization via graph cuts, *IEEE Trans. Pattern Analysis and Machine Intelligence* **23**(11): 1222 - 39, 2001. (doi: 10.1109/34.969114).

Bro R, and de Jong S. A fast non-negativity-constrained least squares algorithm. *J. Chemometrics* **11**(5): 393 – 401, 1997.

Buehler C, Kim KH, Greuter U, Schlumpf N, and So PT. Single-photon counting multicolor multiphoton fluorescence microscope. J Fluoresc. 15(1): 41-51, 2005.

Burton K, Jeong J,Wachsmann-Hogiu S, and Farkas DL. Spectral Optical Imaging in Biology and Medicine, in *Biomedical Optical Imaging*, edited by Fujimoto JG, and Farkas DL, Oxford University Press, 2009.

Campagnola PJ, and Loew LM. Second-harmonic imaging microscopy for visualizing biomolecular arrays in cells, tissues and organisms. *Nat Biotechnol.* **21**(11): 1356-60, 2003.

Candy JV. *Bayesian Signal Processing. Classical, Modern and Particle Filtering Methods*, Wiley, 2009.

Canny J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**(6): 679-98, 1986. (doi: 10.1109/TPAMI.1986.4767851).

Carpenter AE. Image-based chemical screening. *Nat Chem Biol.* **3**(8):461-5, 2007.

Carpenter AE, and Sabatini DM. Systematic genome-wide screens of gene function. *Nat Rev Genet.* **5**(1): 11-22, 2004.

Caselles V, Kimmel R, and Sapiro G. Geodesic Active Contours. *Int. J. Comp. Vis.* **22**(1): 61-79, 1997. (doi:10.1023/A:1007979877043).

Chamberlain J, *The Principles of Interferometric Spectroscopy*, Wiley, New York, 1978.

Chang CI. An Information-Theoretic Approach to Spectral Variability, Similarity, and Discrimination for Hyperspectral Image Analysis. *IEEE Trans. Info Theory* **46**(5): 1927-32, 2000.

Chen JL, Villa KL, Cha JW, So PT, Kubota Y, and Nedivi E. Clustered dynamics of inhibitory synapses and dendritic spines in the adult neocortex. *Neuron* **74**(2): 361-73, 2012.

Chen X, Nadiarynkh O, Plotnikov S, Campagnola PJ. Second harmonic generation microscopy for quantitative analysis of collagen fibrillar structure. *Nat Protoc.***7**(4): 654-69, 2012.

Coelho LP, Shariff A, and Murphy RF. Nuclear segmentation in microscope cell images: A hand-segmented dataset and comparison of algorithms, *ISBI 2009 IEEE Int. Symp. Biomed. Imag*: 518-21, 2009.

Cohen FS, and Cooper DB, Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Noncausal Markovian Random Fields, *IEEE Trans. Pattern Analysis and Machine Intelligence* **9**(2): 195-219, 1987. (doi: 10.1109/TPAMI.1987.4767895).

Cremers D, Tischhäuser F, Weickert J, and Schnörr C. Diffusion snakes: introducing statistical shape knowledge into the Mumford-Shah functional. *Int. J. Comp. Vis.* **50**(3): 295-313, 2002. (doi:10.1023/A:1020826424915).

Crivat G, and Taraska JW. Imaging proteins inside cells with fluorescent tags. *Trends Biotechnol.* **30**(1): 8-16, 2012.

Davis LM, and Shen G., Extension of multidimensional microscopy to ultra-sensitive applications with maximum-likelihood analysis, *Proc. SPIE* 6443: 64430N, 2007.

Dempster AP, Laird NM, and Rubin DB. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. Royal Statist. Soc. B.* **39**(1): 1-38, 1977.

Denk W, Strickler JH, and Webb WW. Two-photon laser scanning fluorescence microscopy. Science. **248**(4951): 73-6, 1990.

Dietrich LE, Teal TK, Price-Whelan A, and Newman DK. Redox-active antibiotics control gene expression and community behavior in divergent bacteria. *Science* **321**(5893):1203-6, 2008.

Dill KA, and Bromberg S. *Molecular Driving Forces: Statistical Thermodynamics in Chemistry & Biology*. Garland Science, 2002.

Dima AA, Elliott JT, Filliben JJ, Halter M, Peskin A, Bernal J, Kociolek M, Brady MC, Tang HC, and Plant AL. Comparison of segmentation algorithms for fluorescence microscopy images of cells. *Cytometry A.* **79**(7): 545-59, 2011.

Dubes RC, Jain AK, Nadabar SG, and Chen CC. MRF model-based algorithms for image segmentation, *Proc. 10th International Conference on Pattern Recognition* **1**: 808-14, 1990. (doi: 10.1109/ICPR.1990.118221).

Eches O, Dobigeon N, and Tourneret J-Y. Markov random fields for joint unmixing and segmentation of hyperspectral images. Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing: 1-4, 2010. (doi: 10.1109/WHISPERS.2010.5594841).

Eliceiri KW, Berthold MR, Goldberg IG, Ibáñez L, Manjunath BS, Martone ME, Murphy RF, Peng H, Plant AL, Roysam B, Stuurman N, Swedlow JR, Tomancak P, and Carpenter AE. Biological imaging software tools. *Nat Methods.* **9**(7): 697-710, 2012. (doi: 10.1038/nmeth.2084).

Elliott H, Derin H, Cristi R, and Geman D. Application of the Gibbs distribution to image segmentation, *IEEE International Conference on Acoustics, Speech, and Signal Processing* **9**(1): 678-81, 1984. (doi: 10.1109/ICASSP.1984.1172637).

Felzenszwalb P, and Huttenlocher D. Efficient Belief Propagation for Early Vision, *Int'l J. Computer Vision* **70**(1): 41-54, 2006.

Fish DA, Brinicombe AM, Pike ER, and Walker JG. Blind deconvolution by means of the Richardson–Lucy algorithm, *J. Opt. Soc. Am. A* **12**(1): 58-65, 1995.

Gallager RG. *Stochastic Processes: Concept s for Applications*, Class Notes, Massachusetts Institute of Technology, 2009

Gannaway J, and Sheppard CJR. Second harmonic imaging in the scanning optical microscope. *Opt. and Quant. Elec.* **10**: 435-9, 1978.

Gao X, Cui Y, Levenson RM, Chung LW, and Nie S. In vivo cancer targeting and imaging with semiconductor quantum dots. *Nat Biotechnol.* **22**(8): 969-76, 2004.

Garini Y, Macville M, du Manoir S, Buckwaldy RA, Lavi M, Katzir N, Wine D, Bar-Am I, Schröck

E, Cabib D, and Ried T. Spectral Karyotyping, *Bioimaging* **4**: 65-72, 1996.

Garini Y, Young IT, NcNamara G, Spectral Imaging: Principles and Application, *Cytom. Part. A* **69A**: 735-47, 2006.

Geman S, and Geman D, Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, *IEEE Trans. Pattern Analysis and Machine Intelligence* **6**(6): 721-41, 1984. (doi: 10.1109/TPAMI.1984.4767596).

Giepmans BN, Adams SR, Ellisman MH, and Tsien RY. The fluorescent toolbox for assessing protein location and function. *Science* **312** (5771): 217-24, 2006.

Glory E, and Murphy RF. Automated subcellular location determination and high-throughput microscopy. *Dev Cell* **12**(1): 7-16, 2007.

Goeppert-Mayer M. Uber Elementarakte mit zwei Quantensprungen. *Annals of Physics* (Leipzig) 5: 273–294, 1931.

Gu M. *Advanced Optical Imaging Theory*, Springer 2000.

Hagwood C, Bernal J, Halter M, and Elliott J, Evaluation of Segmentation Algorithms on Cell Populations Using CDF Curves, *IEEE Trans. on Medical Imaging* **31**(2): 380-90, 2012.

Hakamata T. (Editiorial Chief) et. al. *Photomultiplier Tubes: Basics and Applications*, Third Edition, Hamamatsu Photonics KK, 2006.

Hamilton NA. Open source tools for fluorescent imaging. *Methods Enzymol.***504**: 393-417, 2012. (doi: 10.1016/B978-0-12-391857-4.00020-3).

Hammersley JM, and Clifford P. Markov field on finite graphs and lattices, unpublished, 1971.

Haraguchi T, Shimi T, Koujin T, Hashiguchi N, and Hiraoka Y, Spectral imaging fluorescence microscopy, *Genes to cells* **7**: 881-7, 2002.

Hashimoto H, Isobe K, Suda A, Kannari F, Kawano H, Mizuno H, Miyawaki A, and Midorikawa K. Measurement of two-photon excitation spectra of fluorescent proteins with nonlinear Fourier-transform spectroscopy. *Applied Optics* **49**(17): 3323-9, 2010.

Hell SW. Far-field optical nanoscopy. *Science* **316**(5828): 1153-8, 2007.

Hell SW, and Wichmann J. Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy. *Optics Letters* **19** (11): 780-782, 1994.

Helmchen F, and Denk W. Deep tissue two-photon microscopy. *Nat Methods* **2**(12): 932-40, 2005.

Hiraoka Y, Shimi T, and Haraguchi T. Multispectral imaging fluorescence microscopy for living cells, *Cell struct and funct.* **27**: 367-74, 2002.

Hyvärinen A, Karhunen J, and Oja E. *Independent Component Analysis*, Wiley, New York, 2001.

Im KB, Kang MS, Kim J, Bestvater F, Seghiri Z, Wachsmuth M, and Grailhe R. Two-photon spectral imaging with high temporal and spectral resolution, *Optics Express* **18**(26): 26905-14, 2010.

Jackson Q, and Landgrebe DA, Adaptive Bayesian Contextual Classification Based on Markov Random Fields, *IEEE Trans. Geoscience and Remote Sensing* **40**(11): 2454 - 63, 2002.

Jain V, Seung HS, and Turaga SC. Machines that learn to segment images: a crucial technology for connectomics. *Curr Opin Neurobiol.* **20**(5): 653-66, 2010.

James J, *Spectrograph design fundamentals*, Cambridge university press, Cambridge, 2007.

Juskaitis RJ, Measuring the real point spread function of high numerical aperture microscope objective lenses, in *Handbook of Confocal Microscopy* (J. Pawley editor), Third Edition, Springer, 2006.

Kankaanpää P, Paavolainen L, Tiitta S, Karjalainen M, Päivärinne J, Nieminen J, Marjomäki V, Heino J, and White DJ. BioImageXD: an open, general-purpose and high-throughput image-processing platform. *Nat Methods.* **9**(7): 683-9, 2012. (doi: 10.1038/nmeth.2047).

Kass M, Witkin A, and Terzopoulos D, Snakes: Active Contour Models. *Int. J. Comp. Vis.* **1**(4): 321-31, 1988.

Kass RE, and Rafterry AE. Bayes Factors, *J. Am. Stat. Ass.* **90**(430): 773-95, 1995.

Kaufman L. Implementing and Accelerating the EM Algorithm for Positron Emission Tomography. *IEEE Trans. on Medical Imaging* **6**(1): 37-51, 1987. (doi: 10.1109/TMI.1987.4307796)

Kaufman L. Maximum Likelihood, Least Squares, and Penalized Least Squares for PET, *IEEE Trans. Med. Imag.* **12**(2): 200-14, 1993.

Keenan MR, Timlin JA, van Benthem MH, and Haaland DM, Algorithms for Constrained Linear Unmixing with Application to the Hyperspectral Analysis of Fluorophore Mixtures, *Proc. SPIE* **4816**: 193 - 202, 2002.

Kerfoot IB, and Bresler Y. Theoretical analysis of multispectral image segmentation criteria. *IEEE Trans. on Image Processing* **8**(6): 798-820, 1999. (doi: 10.1109/83.766858).

Kerschensteiner M, Schwab ME, Lichtman JW, and Misgeld T. In vivo imaging of axonal degeneration and regeneration in the injured spinal cord. *Nat Med.* **11**(5): 572-7, 2005.

Keshava N. A survey of spectal unmixing algorithms, *Lincoln Lab journal* **14**(1): 55-78, 2003.

Kim KH, Development of high-speed two-photon microscopy for biological and medical applications, PhD Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 2005.

Kim KH, Buehler C, Bahlmann K, Ragan T, Lee WC, Nedivi E, Heffer EL, Fantini S, So PT. Multifocal multiphoton microscopy based on multianode photomultiplier tubes. *Opt Express* **15**(18): 11658-78, 2007.

Kim KH, Ragan T, Previte MJ, Bahlmann K, Harley BA, Wiktor-Brown DM, Stitt MS, Hendricks CA, Almeida KH, Engelward BP, and So PT. Three-dimensional tissue cytometer based on high-speed multiphoton microscopy. *Cytometry A.* 71(12): 991-1002, 2007b.

Kim D, and So PT. High-throughput three-dimensional lithographic microfabrication. *Opt Lett.* **35**(10): 1602-4, 2010.

Kitami T, Logan DJ, Negri J, Hasaka T, Tolliday NJ, Carpenter AE, Spiegelman BM, and Mootha VK. A chemical screen probing the relationship between mitochondrial content and cell size. *PLoS One* **7**(3):e33755, 2012.

Koller D, and Friedman N. *Probabilistic graphical models: principles and techniques*, MIT Press, 2009.

Kolmogorov V. Convergent Tree-Reweighted Message Passing for Energy Minimization, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **28**(10): 1568-83, 2006. (doi: 10.1109/TPAMI.2006.200).

Kolmogorov V, and Rother C. Comparison of energy minimization algorithms for highly connected graphs, *Proc. 9th European Conf. Comp.Vision*: 1-15, 2006.

Konishi S, and Kitagawa G. *Information Criteria and Statistical Modeling*, Springer, 2008.

Lakowicz JR. *Principles of Fluorescence Spectroscopy*, Third Edition, Springer, 2006.

Landsford R, Bearman G, and Fraser SE, Resolution of multiple green fluorescent protein color variants and dyes using two-photon microscopy and imaging spectroscopy, *J. Biom. Optics* **6**(3): 311-8, 2001.

Lawson C, and Hanson R. *Solving Least Squares Problems*, Prentice Hall, New Jersey, 1974.

Lee DD, and Seung HS. Learning the parts of objects by non-negative matrix factorization. *Nature* **401**(6755): 788-91, 1999.

Li SZ. *Markov Random Fields Modeling in Image Analysis*, Third Edition, Springer, 2009.

Lippincott-Schwartz J, and Patterson GH. Photoactivatable fluorescent proteins for diffraction-limited and super-resolution imaging. *Trends Cell Biol.* **19**(11): 555-65, 2009.

Lippincott-Schwartz J. Emerging in vivo analyses of cell function using fluorescence imaging. *Annu Rev Biochem.* **80**: 327-32, 2011.

Livet J, Weissman TA, Kang H, Draft RW, Lu J, Bennis RA, Sanes JR, and Lichtman JW. Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. *Nature*. **450**(7166): 56-62, 2007.

Lucy LB. An iterative technique for the rectification of observed distributions. *Astronom. J.* **79**(6): 745-54, 1974.

Luo Y, and Duraiswami R. Efficient Parallel Nonnegative Least Squares on Multicore Architectures. *SIAM J. Sci. Comput.* **33**(5): 2848–63, 2011.

MacKay D. *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.

Malik Z, Cabib D, Buckwald RA, Talmi A, Garini Y, and Lipson SG. Fourier transform multipixel spectroscopy for quantitative cytology, *J. Microsc.* **182**(2): 133-40, 1996.

Marroquin J, Mitter S, and Poggio T. Probabilistic Solution of Ill-Posed Problems in Computational Vision, *J. Am. Statist. Assoc.* **82** (397): 76-89, 1987.

Marroquin JL., Santana EA., and Botello S. Hidden Markov measure field models for image segmentation, *IEEE Trans. Pattern Analysis and Machine Intelligence* **25**(11): 1380-7, 2003.

Masters B, and So PTC. *Handbook of Biomedical Nonlinear Optical Microscopy.* Oxford University Press, 2008.

Mavrodi DV, Blankenfeldt W, and Thomashow LS. Phenazine compounds in fluorescent Pseudomonas spp. biosynthesis and regulation. *Annu Rev Phytopathol.* **44**:417-45, 2006.

Mittag A, Lenz D, Gerstner AO, Sack U, Steinbrecher M, Koksch M, Raffael A, Bocsi J, and Tárnok A. Polychromatic (eight-color) slide-based cytometry for the phenotyping of

leukocyte, NK, and NKT subsets. *Cytometry A.* **65**(2): 103-15, 2005.

Mondal PP, Vicidomini G, and Diaspro A. Image reconstruction for multiphoton fluorescence microscopy, *Appl. Phys. Lett.* **92**(10): 103902, 2008. (doi: 10.1063/1.2888177).

Mondal PP, Vicidomini G, and Diaspro A. Markov random field aided Bayesian approach for image reconstruction in confocal microscopy, *J. Appl. Phys.* **102**(4): 044701, 2007.

Neher R, and Neher E. Optimizing imaging parameters for the separation of multiple labels in a fluorescent image, *J. Microsc.* **213**(1): 46-62, 2004.

Ng HP, Ong SH, Foong KWC, Goh PS, and Nowinski WL. Medical Image Segmentation Using K-Means Clustering and Improved Watershed Algorithm. *IEEE Southwest Symposium on Image Analysis and Interpretation*: 61-65, 2006. (doi: 10.1109/SSIAI.2006.1633722).

Nir O, Bakal C, Perrimon N, and Berger B. Inference of RhoGAP/GTPase regulation using single-cell morphological data from a combinatorial RNAi screen, *Genome Res.* **20**(3): 372-80, 2010.

Nocedal J, and Wright S. *Numerical Optimization*, Springer, 2006.

Oppenheim AV, Willsky AS, and Hamid S. *Signals and Systems*, Second edition, Prentice Hall, 1996.

Orvedahl A, Sumpter R Jr, Xiao G, Ng A, Zou Z, Tang Y, Narimatsu M, Gilpin C, Sun Q, Roth M, Forst CV, Wrana JL, Zhang YE, Luby-Phelps K, Xavier RJ, Xie Y, and Levine B. Image-based genome-wide siRNA screen identifies selective autophagy factors. *Nature* **480**(7375): 113-7, 2011.

Otsu N. Threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**: 62-6, 1979. (doi: 10.1109/TSMC.1979.4310076).

Palmer C, and Loewen E. *Diffraction grating handbook*, Sixth Edition, Newport Corporation, 2005.

Pawley J. *Handbook of Biological Confocal Microscopy*, Third Edition, Springer, 2006.

Pearl J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Pepperkok R, and Ellenberg J. High-throughput fluorescence microscopy for systems biology. *Nat Rev Mol Cell Biol.* **7**(9):690-6, 2006.

Perlman ZE, Slack MD, Feng Y, Mitchison TJ, Wu LF, and Altschuler SJ, Multidimensional drug profiling by automated microscopy. *Science* **306**(5699): 1194-8, 2004.

Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes. The art of scientific computing.* Third Edition, Cambridge University Press, 2007.

Price-Whelan A, Dietrich LE, and Newman DK. Rethinking 'secondary' metabolism: physiological roles for phenazine antibiotics. *Nat Chem Biol.* **2**(2): 71-8, 2006.

Radosevich AJ, Bouchard MB, Burgess SA, Chen BR, and Hillman EM. Hyperspectral in vivo two-photon microscopy of intrinsic contrast. *Opt Lett.* **33**(18): 2164-6, 2008.

Ragan T, Kadiri LR, Venkataraju KU, Bahlmann K, Sutin J, Taranda J, Arganda-Carreras I, Kim Y, Seung HS, Osten P. Serial two-photon tomography for automated ex vivo mouse brain imaging. Nat Methods. **9**(3): 255-8, 2012. (doi: 10.1038/nmeth.1854).

Rand WM. Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Ass* **66**(336): 846-50, 1971.

Rao CR, Linear Statistical Inference and its Applications, Second edition, Wiley-Interscience, 2001.

Rich WJ, *Examining the accuracy of the normal approximation to the Poisson random variable*, Masters Theses and Doctoral Dissertations. Eastern Michigan University, 2009.

Richards JA, and Jia X. *Remote Sensing Digital Image Analysis, An Introduction*. Fourth Edition. Springer, 2006.

Richardson WH. Bayesian-Based Iterative Method of Image Restoration. *J. Opt. Soc. Am.* **62**(1): 55-9, 1972.

Rietdorf JR, and Stelzer EHK. Special Optical Elements, in *Handbook of Biological Confocal Microscopy*, Third edition, Pawley JB editor, Springer, 2006.

Rust MJ, Bates M, and Zhuang X. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nat Methods* **3**(10): 793-5, 2006.

Saleh BE, and Teich MC, *Fundamentals of Photonics*, Second edition, Wiley-interscience, 2007.

Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez JY, White DJ, Hartenstein V, Eliceiri K, Tomancak P, and Cardona A. Fiji: an open-source platform for biological-image analysis. *Nat Methods.* **9**(7): 676-82, 2012. (doi: 10.1038/nmeth.2019).

Schwarz JW, and Staenz K, Adaptive Threshold for Spectral Matching of Hyperspectral Data, *Can. J. Rem. Sens.* **27**(3): 216-24, 2001.

Shapiro LG, and Stockman GC. *Computer Vision*, Prentice-Hall, New Jersey, 2001.

Shepp LA, and Vardi Y. Maximum Likelihood Reconstruction for Emission Tomography, *IEEE Trans. Med. Imag.* **1**(2): 113-122, 1982.

Shewchuk JR. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*, Technical Report, Carnagie Mellon University, 1994.

Sinclair MB, Haaland DM, Timlin JA, and Jones HDT, Hyperspectral confocal microscope, *Applied Optics* **45**(24): 6283-91, 2006.

Sivia D, and Skilling J. *Data Analysis: A Bayesian Tutorial*, Second edition, Oxford University Press, 2006.

So PTC. Two-photon Fluorescence Light Microscopy, Encyclopedia of life sciences, 2002.

So PT, Dong CY, Masters BR, Berland KM. Two-photon excitation fluorescence microscopy. *Annu Rev Biomed Eng.* **2**: 399-429, 2000.

So PT, and Kim D. Depth resolved wide field illumination for biomedical imaging and fabrication. *Conf Proc IEEE Eng Med Biol Soc. 2009*: 3234-5, 2009.

Soille P, *Morphological Image Analysis: Principles and Applications*, Second Edition, Springer, 2003.

Strang G. *Linear algebra and its applications*, Third edition, Harcourth Brace Jovanovich, 1988.

Strauss DJ. Clustering on Coloured Lattices, *J. Appl.Prob.* **14**(1): 135-43, 1977.

Sullivan NL, Tzeranis DS, Wang Y, So PT, and Newman D. Quantifying the dynamics of

bacterial secondary metabolites by spectral multiphoton microscopy. *ACS Chem Biol.* **6**(9): 893-9, 2011.

Szeliski R, Zabih R, Scharstein D, Veksler O, Kolmogorov V, Agarwala A, Tappen M, and Rother C. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(6): 1068-80, 2008. (doi: 10.1109/TPAMI.2007.70844).

Tappen MF, and Freeman WT. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters, *Proc. 9th IEEE International Conference on Computer Vision* **2**: 900-6, 2003.

Tsui BM, Zhao X, Frey EC, and Gullberg GT. Comparison between ML-EM and WLS-CG Algorithms for SPECT Image Reconstruction. *IEEE Trans. Nucl. Sci.* **38**(6): 1766-72, 1991. (doi: 10.1109/23.124174).

Tsupryk A, Tovkach I, Gavrilov D, Kosobokova O, Gudkov G, Tyshko G, Gorbovitski B, and Gorfinkel V. Ultra sensitive sensor with enhanced dynamic range for high speed detection of multi-color fluorescence radiation. *Biosens Bioelectron.* **23**(10): 1512-8, 2008.

Tsurui H, Nishimura H, Hattori S, Hirose S, Okumura K, and Shirai T. Seven-color fluorescence imaging of tissue samples based on fourier spectroscopy and singular value decomposition, *J. Histochem. Cytochem.* **48**(5): 653-62, 2000.

van Benthem MH, Keenan MR, and Haaland DM, Application of equality constraints on variables during alternating least squares procedures, *J. Chemometrics* **16**(12): 613-22, 2002.

van Benthem MH, and Keenan MR. Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. *Journal of Chemometrics* **18**(10): 441-50, 2004.

van Benthem MH, and Keenan MR. Fast combinatorial algorithm for the solution of linearly constrained least squares problems. US Patent # 7,451,173, Issued Nov. 11, 2008.

Varshney PK, and Arora MK. *Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data*, Springer, 2004.

Vermaas WF, Timlin JA, Jones HD, Sinclair MB, Nieman LT, Hamad SW, Melgaard DK, and Haaland DM. In vivo hyperspectral confocal fluorescence imaging to determine pigment localization and distribution in cyanobacterial cells. *Proc Natl Acad Sci U S A* **105**(10): 4050-5, 2008.

Wachman ES, Niu W, Farkas DL, AOTF microscope for imaging with increased speed and spectral versality, *Biophys. J.* **73**(3): 1215-22, 1997.

Wainwright MJ, Jaakkola TS, and Willsky AS. Tree-based reparameterization framework for analysis of sum-product and related algorithms, *IEEE Trans. Information Theory* **49**(5): 1120-46, 2003. (doi: 10.1109/TIT.2003.810642).

Wornell GW. *Lecture Notes for Course 6.437 (Inference and Information)*. Massachusetts Institute of Technology, 2010.

Xu P, and Prince JL. Gradient vector flow: a new external force for snakes. *Proc. IEEE Conf. on Comp. Vis. Patt. Recog*: 66-71, 1997.

Xu C, and Webb W. Measurement of two-photon excitation cross sections of molecular

fluorophores with data from 690 to 1050 nm, *J. Opt. Soc. Am. B.* **13**: 481-91, 1996.

Xu C, Zipfel W, Shear JB, Williams RM, Webb WW. Multiphoton fluorescence excitation: new spectral windows for biological nonlinear microscopy. *Proc Natl Acad Sci U S A.* **93**(20): 10763-8, 1996.

Weiss Y, and Freeman WT. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs, *IEEE Trans. Information Theory* **47**(2): 736-44, 2001.

Yannas IV, Burke JF, Orgill DP, and Skrabut EM. Wound tissue can utilize a polymeric template to synthesize a functional extension of skin, *Science* **215**(4529): 174-6, 1982.

Yannas IV, Lee E, Orgill DP, Skrabut EM, and Murphy GF. Synthesis and characterization of a model extracellular matrix that induces partial regeneration of adult mammalian skin, *Proc. Natl. Acad. Sci. USA* **86**(3): 933-7, 1989.

Yedidia JS, Freeman WT, and Weiss Y. Generalized Belief Propagation, *Advances in Neural Information Processing Systems (NIPS)* **13**: 689-95, 2000.

Zhang J, Campbell RE, Ting AY, and Tsien RY. Creating new fluorescent probes for cell biology. *Nat Rev Mol Cell Biol.* **3**(12): 906-18, 2002.

Zijdenbos AP, Dawant BM, Margolin RA, and Palmer AC. Morphometric analysis of white matter lesions in MR images: method and validation. *IEEE Trans. on Medical Imaging* **13**(4): 716-24, 1994. (doi: 10.1109/42.363096).

Zimmermann T. Spectral Imaging and Linear Unmixing in Light Microscopy, *Adv. Biochem Engin./Biotechnol.* **95**: 245-65, 2005.

Zipfel WR, Williams RM, Webb WW. Nonlinear magic: multiphoton microscopy in the biosciences. *Nat Biotechnol.* **21**(11):1369-77, 2003.

Chapter 3: An Image Informatics Study of TGFβ Signaling via the SMAD Pathway

3.1 Chapter Overview

This chapter describes an imaging-based method for studying signal transduction pathways in cells that interact with 3D matrices. The methodology belongs to the "image informatics" framework: inference about the role of individual components of a signaling pathway is based on how particular perturbations to the pathway affect cell response to particular stimuli, where the system response is quantified based on imaging assays. It combines RNA interference, the spectral multi-photon imaging techniques described in Chapter 2, and statistical analysis. The methodology is applied in a pilot study of the TGFβ signaling via the SMAD pathway in fibroblasts inside porous collagen scaffolds (CG), in order to infer how different components of the pathway modulate the different effects of TGFβ1 and TGFβ3 isoforms.

3.1.1 Motivation

Cells sense and respond to their environment via signal transduction pathways, cascades of proteins and small molecules that transmit information (in the form of molecule concentration and modification) inside the cell [Gomperts et al. 2009; Scott and Pawson 2009; Gough and Yaffe 2011; Palsson 2011]. The vast majority of existing knowledge about signal transduction pathways is based on established biochemical and genetic methods. Current efforts to model and understand signal transduction using complex biological systems requires novel high-throughput approaches, including genomic and proteomic methods [Yaffe 2008; Zhang and Neubert 2009; Zhang et al. 2010], and imaging-based methods such as image informatics [Perlman et al. 2004; Bakal et al. 2007; Wollman and Stuurman 2007].

Image informatics methodologies combine genetic manipulation of cells, large scale imaging, image processing and statistics to study complex signal transduction pathways in cells. So far, image informatics has been applied in studies of cells cultured on standard culture dishes. Unfortunately, the 2D surface of a culture dish presents to cells an environment that is very different compared to the microenvironments felt inside tissues. Since the insoluble microenvironment of cells affects critically their phenotypes [Griffith and Swartz 2006], it is important to develop imaging-based studies that can study cells inside more physiologically relevant 3D matrix models.

This study extends the image informatics framework into studies of signal transduction pathways in cells that interact with a 3D matrix. The role of individual pathway components is inferred based on how knocking down particular components of the cell-matrix system affects the system response to particular stimuli. A novel feature compared to previous image informatics approaches is that the output of the cell-matrix system is quantified not just based on morphometric features of the cells, but also on morphometric features that describe cell-matrix adhesion. The methodology is applied in a pilot study on the role of SMAD pathway components in mediating TGFβ signaling, a pathway known to affect many cell phenotypes, including myofibroblast differentiation, a critical phenotype in wound healing (Chapter 4).

3.1.2 Chapter Description

Section 3.2 provides basic background on the image informatics approach, on the SMAD pathway and on the role of TGFβ in wound healing. Section 3.3 describes the various steps of the image informatics methodology, the experimental implementation in the SMAD ilot study, the application-specific image processing steps, and the immunoblotting assay used to validate the results of the imaging analysis. Section 3.4 provides the results of the study, Section 3.5 discusses the findings, and Section 3.6 suggests directions for future research.

3.1.3 Contributions to this study

This study was a collaborative effort of the author of this thesis (Dimitrios Tzeranis) with two visiting students (Chengpin Shen, Jin Guo) at the So lab (MIT). Chengpin Shen developed the lentiviral shRNA vector expression system and contributed to experimental design and several experimental protocols (shRNA transfection, immune-blotting and immune-fluorescence). Jin Guo finalized and verified the shRNA transfection procedure, finalized WB and IF protocols, verified the shRNA protein knock-down efficiency, and acquired the data. Melissa contributed in acquisition of Western Blot data, and preparation of samples for imaging. The author (DT) contributed in experimental design, protocol troubleshooting, developed the image processing pipeline, and performed statistical analysis in both IF and WB data.

3.2 Background

3.2.1 Image Informatics

Description of Image Informatics

Image informatics refers to imaging-based approaches that infer information about the components of a cell signaling pathway. The pathway of interest is modeled as a system, whose response to external stimuli (inputs) can be quantified based on imaging-based assays (outputs) called features, Figure 3.2.1. The output response of the system is expected to change significantly when a critical part of the system is perturbed. Therefore, the roles of individual pathway components can be inferred by quantifying how the system output varies over different treatment conditions.

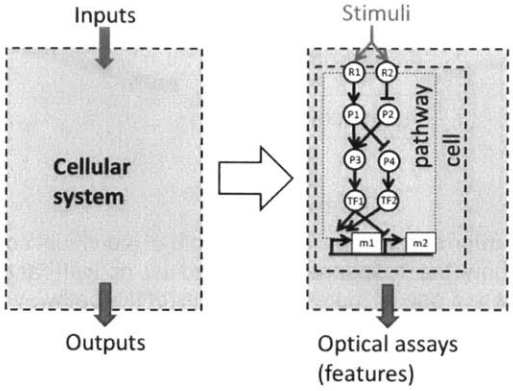


Figure 3.2.1: Image informatics considers the signal transduction pathway of interest as a system, which is excited by stimuli (inputs) and quantified via optical assays (features).

Each experimental condition is called a treatment condition (TC), and corresponds to a combination of a particular input applied to a particular perturbed system, see Figure 3.2.2. The role of pathway components is inferred by statistical analysis of the outputs of all TC. Signal transduction pathways are usually large (in terms of component number) and complex (in terms of component topology, variation among different cells) [Gomperts et al. 2009]. Therefore in order to pick the most informative TC and analyze TC results it is necessary to incorporate a-priori knowledge of the basic biology of the pathway.

Key Features of Image Informatics

Image informatics methodologies are based on large-scale high-resolution imaging of a large number of treatment conditions (TC). Each TC is described based on images of cells (here cells are located inside a matrix), where each individual cell is quantified by a vector of imaging-based assays called "features". Each TC is described therefore by the feature matrix that contains the feature vectors of its cells. Since the methodology relies on imaging assays, it can be applied to systems where the pathway of interest affects markers that can be detected by imaging. From a systems theory perspective this describes the need for a system (signaling pathway) that is "controllable" by the stimuli and "observable" by the measured outputs (feature vector) [Chen 1998].

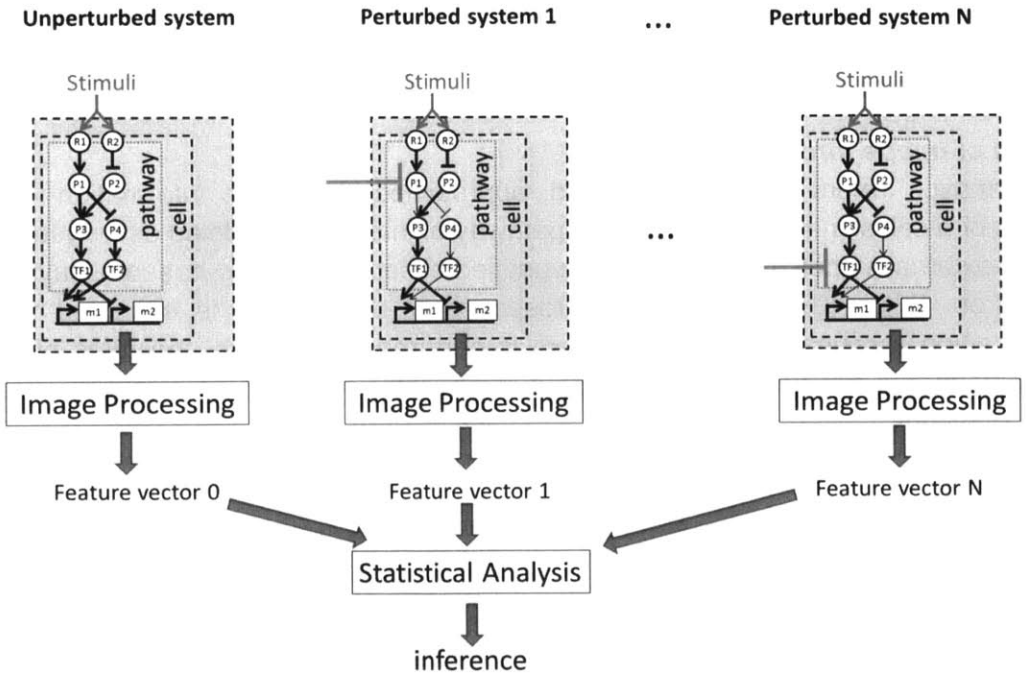


Figure 3.2.2: Overview of the image informatics approach. The role of individual components of a signal transduction pathway is inferred by quantifying how the response (evaluated by optical-based assays called features) of the system to particular stimuli changes when one or more components of the pathway are perturbed.

Image informatics infers information about an intracellular molecular system and does not provide direct biochemical/genetic evidence. Therefore it is usually necessary to verify its conclusions through standard biochemical or genetic experiments. In some cases, statistical analysis of image informatics data includes known positive/negative results in order to estimate the sensitivity/selectivity of the approach [Perlman et al. 2004; Nir et al. 2010]. Even when this

validation is not possible, a key contribution of image informatics is to accelerate discovery in complex biological systems through guiding the focus and design of the more laborious, and expensive biochemical and genetic experiments needed to obtain solid proof of molecular interactions and biological significance.

State-of-the-Art in Image Informatics

The pioneering paper in the image informatics field profiled the dose-dependent effects of 100 different drugs in human cell morphology, grouped drugs of similar mechanisms, and predict targets for drugs of uncertain mechanism [Perlman et al. 2004]. Another key study classified cell morphologies in a RNAi screening of 249 genes predicted to affect cell morphology, and identified signaling networks that regulate cell protrusion, adhesion and tension [Bakal et al. 2007]. RNAi screening and single-cell classification was used to infer potential RhoGAP-GTPase interactions in fruit fly cells [Nir et al. 2010]. High-dimensional quantitative phenotyping found that half of 4718 nonessential gene deletion yeast mutants affected cell morphology [Green and Yamada 2007]. A whole genome RNA interference screen combined with automated image analysis identified genes that affect cell spreading in Drosophila [Madri et al. 1988]. A novel segmentation algorithm applied to more than 55,000 cell images infected by either hepatitis C or dengue virus identified genes involved in virus entry and replication [Grinnell 2003]. A FRET-based screening of JNK regulators by 1565 dsRNAs detected groups of individual dsRNAs with common patterns of genetic interaction [Bakal et al. 2008].

Limitations of Current Image Informatics

The major limitation of current image informatics implementations is that they are based on images of cells cultured on flat culture disks. Although cell culture on culture plates is the standard practice in biological research, it lacks a biologically-relevant extracellular matrix (Section 5.2.1). It is known that many important physiological or pathologic cell phenotypes depend critically on the matrix around cells [Madri et al. 1988, Grinnell 2003, Stegemann et al. 2005; Griffith and Swartz 2006; Allen and Jones 2011]. The absence of a physiologically relevant matrix prevents accurate studies of many important phenomena. Current image informatics implementations can be considered as a first step towards more biologically relevant models that could emulate the microenvironment that cells sense in tissues [Green and Yamada 2007], where cells interact either with other cells or with the extracellular matrix. The next generation of image-informatics implementations should therefore include an appropriate matrix, treat the cell-matrix system as a whole, and utilize imaging-based assays to quantify both cell morphology and cell-matrix interactions.

The vast majority of current image informatics studies quantify TC based on morphometric assays of cells using a single fluorescent stain of the cell cytoplasm. Since cell-matrix systems are significantly more complex, it seems necessary to describe them by a larger variety of assays. In this case it may not be adequate to rely on cell morphometric features derived from a single fluorescent stain localized in the cell cytosol [Madri et al. 1988, Grinnell 2003, Stegemann et al. 2005]. There is a need to describe such complex cell-matrix systems by multiple fluorescent markers. At the same time it is necessary not to increase the number of imaging experiments (and the associated costs). These two contradicting objectives can be satisfied by utilizing high-throughput high-content 3D-imaging modalities, such as spectral multi-photon imaging described in Chapter 2 and utilized in this study.

3.2.2 Image Processing

Image processing pipelines for image informatics are long and complex. They consist of a generic pipeline, followed by an application-specific pipeline.

The generic pipeline depends on the instrument being used. Most studies utilize wide-field fluorescence microscopy in commercially available automated imaging platforms (e.g. cellomics) that provide software that implements the generic pipeline. This thesis utilizes a custom-built spectral multi-photon microscope. The generic image processing pipeline for spectral microscopy described in Chapter 2 provides: i) a segmentation of the 3D image into regions belonging to particular "classes" (kinds of objects), and ii) the ML estimate of the rates of the emission sources present in the image.

The application-specific image processing pipeline processes the outputs of the generic pipeline in order to identify and quantify objects of interest. In the image informatics implementation discussed here, the objects of interest are cells (described by the classes "nucleus" and "cytoplasm"), and the surrounding matrix (described by the class "scaffold"). Processing cell objects is quite tricky because cells may touch each other, making necessary to segregate cell clusters. The same problem occurs in cells interacting with a matrix. Therefore, here the application-specific part of the image processing pipeline focuses on identifying and quantifying single cells in cell clusters.

Processing Images that Contain Cell Clusters

Imaging-based methods often face the challenge of processing images that contain cell clusters. Part of the image processing pipeline focuses on identifying accurately the boundaries of cell nuclei and cytoplasm in such cell clusters. Published pipelines usually consist of five parts:

1. **Pre-processing**: Prepares data for segmentation using operations such as denoising, voxel size correction (3D imaging) and intensity correction. Noise artifacts can be attenuated using low-pass filtering [Lin et al. 2003; Wählby et al. 2004; Adiga et al. 2006]. Intensity correction compensates for non-uniform illumination or for intensity attenuation in 3D imaging [Wählby et al. 2004].
2. **Segmentation**: In the vast majority of published studies, images consist of one or two emission sources, whose emission is detected in two separate channels. One channel usually detects a nucleic acid stain that labels cell nuclei. The second channel detects a stain that is localized in cell cytoplasms. In the segmentation step, each channel is divided into "foreground" (the class of interest, e.g. nuclei, cytoplasm) and "background" using segmentation algorithms developed for single-channel grayscale images (discussed in detail in Section 2.2.4). When the image contains non-overlapping cell cytoplasm regions (e.g. low-density cell cultures where cells are well separated and do not make large cell-cell contacts) the segmentation task is relatively easy and simple segmentation algorithms can be sufficient [Dima et al. 2011; Hangwood et el 2012]. Some kinds of segmentation errors can be corrected using morphological processing such as filling and opening [Lin et al. 2003; Cheng and Rajapakse 2009]. More advanced and accurate methods include active contours [Zimmer et al. 2002; Xiong et al. 2006; Cheng and Rajapakse 2009], watersheding based on channel intensity gradient [Wählby et al. 2004], thresholding [Lin et al. 2003; Adiga et al. 2006; Xiong et al. 2006], Markov Random Fields [Al-Kofahi et al. 2010], and dynamic

programming [Rosado-Toro and Rodriguez 2012]. In all published papers the signal level in the class of interest is significantly larger compared to the background signal. Usually segmentation of cell nuclei is easier than segmentation of cell cytoplasm because nuclei have sharper edges and standard shapes.

3. **Seeding**: When the image contains regions of overlapping objects (e.g. nuclei or cell cytoplasms in contact), it is critical to first identify the number of the objects, and provide an initial seed for each object. In several studies the user manually provides the seeds of the overlapping regions [Zimmer et al. 2002; Jones et al. 2005]. In high-throughput automated studies it is preferred to identify these seeds computationally. Nuclei objects are easier to identify. Nuclei objects can be used as seeds for cell cytoplasm [Xiong et al. 2006]. This is adequate in 2D imaging but may not be enough in 3D imaging. Most published pipelines use as seeds the local minima of the internal distance transform [Soille 2003; Lin et al. 2003; Wählby et al. 2004; Adiga et al. 2006; Cheng and Rajapakse 2009]. This approach usually provides more seeds than the actual number of objects particularly in noisy images. Some noisy seeds can be eliminated by h-min morphological filtering [Cheng and Rajapakse 2009]. Alternative methods for seeding include applying Laplacian of Gaussian filtering followed by local maximum clustering [Al-Kofahi et al. 2010], and identifying the local minima of the gradient-weighted inner distance transform [Lin et al. 2003].
4. **Splitting (region growing)**: This step divides a cytoplasm region that corresponds to a cell cluster into individual cell objects. Splitting is based on the seeds provided by the seeding step and information provided by the image. A simple but inaccurate approach is Voronoi segmentation, where each pixel is assigned to the nearest seeding point [Soille 2003]. The standard approach for splitting regions in contact is to apply the watershed algorithm in the inner distance transform [Wählby et al. 2004; Adiga et al. 2006]. Improved cell splitting is achieved by incorporating edge information by applying the watershed algorithm to a gradient-weighted distance transform [Lin et al. 2003], or by applying Voronoi segmentation to a modified distance metric that incorporates gradient information [Jones et al. 2005] (the method implemented in the CellProfiler image processing platform) [Lamprecht et al. 2007; Vokes and Carpenter 2008]. Other methods include identifying cell contacts using gradient-difference information [Surut and Phukpattaranont 2010], Markov random field modeling [Chen et al. 2006; Al-Kofahi et al. 2009] (included in the FARSIGHT image processing platform [Bjornsson et al. 2008]), applying dynamic programming to identify the minimum cumulative path [Rosado-Toro and Rodriguez 2012], applying the watershed algorithm in the outer distance transform [Cheng and Rajapakse 2009] (appropriate for circular regions of similar dimensions), and active contour methods that can handle touching regions [Xiong et al. 2006; Coulot et al. 2006]. In all cases, successful splitting requires accurate seeding. There are few exceptions where segmentation and splitting are implemented in a single-step, for example using Markov Random Fields [Chen et al. 2006].
5. **Post-processing**: This step corrects over-segmentation caused by incorrect seeding [Lin et al. 2003; Coelho et al. 2009].

In this study, parts 1-2 are implemented by the generic image processing pipeline described in Chapter 2, and parts 3-5 are implemented by an application-specific image processing pipeline described in Section 3.3.5.

3.2.3 Mathematical Methods in Image Informatics

Image informatics methodologies rely on analyzing large data sets obtained by high-throughput imaging experiments. They can borrow several tools developed in computer science and systems biology for processing large data sets in order to extract statistically-significant correlations [Tan and Linding 2009]. This section summarizes some of these methods.

Dimension Reduction

Each cell object is described by a large number of features (a feature vector). Each treatment condition (TC) is described by a feature matrix that contains the feature vectors of a population of cells. Due to the large dimension of feature vectors utilized in image informatics, it is necessary to reduce the dimension of the feature vector in order to remove redundancy, increase calculation accuracy and decrease computational cost. Dimensionality reduction methods include projection-based, and feature selection techniques. Projection-based methods, particularly principal component analysis (PCA), have been applied extensively, including several image informatics studies [Nir et al. 2010; Lee et al. 2012]. The main disadvantage of PCA is that the new base set does not correspond to features of biological context. Feature selection techniques reduce dimension by selecting a subset of the initial features in order to minimize redundancy or maximize the information content without data transformation [Hall 2000].

Statistical Analysis of Image Informatics Data

There is no standard way to process the large amount of imaging data provided by image informatics experiments. This thesis applies a simple statistical analysis based on clustering similar to the one implemented in a related study [Nir et al. 2010]. Clustering segregates TC into groups of "similar" reduced feature arrays. Clustering methods include the widely-adopted k-means and expectation maximization (EM) methods [Jain et al. 1999]. The centroid-based k-means clustering is visually intuitive and guaranteed to converge, however it lacks the theoretical statistical foundation provided by distribution-based methods such as the expectation maximization (EM) clustering method. EM clustering, when applied to data sets modeled by the Gaussian mixture model (GMM), provides the maximum a posteriori (MAP) estimates of the GMM parameters [Dempster et al. 1977].

3.2.4 SMAD-Mediated TGFβ-Induced Myofibroblast Differentiation

The developed image informatics methodology is demonstrated in a pilot study of the SMAD signal transduction pathway in primary fibroblasts inside porous collagen scaffolds. The SMAD pathway is the major pathway that transduces the effects of TGFβ cytokines [Massagué 2000], important regulators of many pathological processes including wound healing [Yannas 2001]. Of particular interest is the effect of TGFβ1 and TGFβ3 isoforms on myofibroblast differentiation during wound healing.

Myofibroblasts and Wound Healing

Myofibroblasts (MFB) are transformed cells that express significant contractile apparatus [Deodhar and Rana 1997; Desmouliere et al. 2005]. MFB appear in several pathological conditions including wound healing and fibrosis [Desmouliere et al. 2005; Hinz 2007, 2009]. Several kinds of cells in various organs (e.g. fibroblasts in skin, stellate cells in liver [Masuzaki

et al. 2012]) can differentiate into MFB by appropriate stimuli [Hinz et al. 2007]. During spontaneous wound healing, capsules of contractile cells rich in MFB appear at the wound edges [Murphy et al. 1990; Troxel 1994; Chamberlain 1998b]. The presence of MFB in wound healing is correlated with wound contraction and scar formation [Yannas 2001]. MFB apply significant contractile forces and synthesize a collagen-rich capsule. Both effects are believed to have significantly adverse effects on the wound healing outcome, and in particular on the ability of the organ to regenerate spontaneously (for more details, see Chapter 4).

The SMAD Signal Transduction Pathway

The SMAD pathway is the major signal transduction pathway that cells use to transduce TGFβ stimuli [Massagué 1998; Massagué and Gomis 2006]. The pathway operates in six steps (Figure 3.2.3). 1: The TGFβ molecule binds to a TGFβ type II receptor (TGFBR2) dimer on the cell membrane. 2: The TGFBR2 complex recruits a TGFβ type I receptor (TGFBR1) dimer (homodimer or heterodimer). 3: TGFBR2 activates TGFBR1 by phosphorylating residues at its cytosolic domain. 4: The cytosolic domains of activated TGFBR1 recruit and phosphorylate receptor-SMADs (rSMAD: SMAD1, 2, 3, 5, 8). 5: Activated rSMADs bind the coSMAD Smad4. 6: The rSMAD-coSMAD complex translocates into the nucleus and regulates gene transcription by acting as a transcription factor [Goumans et al. 2002, 2003, 2009; Feng and Derynck 2005; Massagué and Gomis 2006].

Cytokines		TGFβ1, TGFβ2, TGFβ3
Receptors	**Type I**	ALK1, ALK5
	Type II	TGFBR2
SMAD proteins	**rSMAD**	SMAD1, SMAD2, SMAD3, SMAD5, SMAD8
	coSMAD	SMAD4

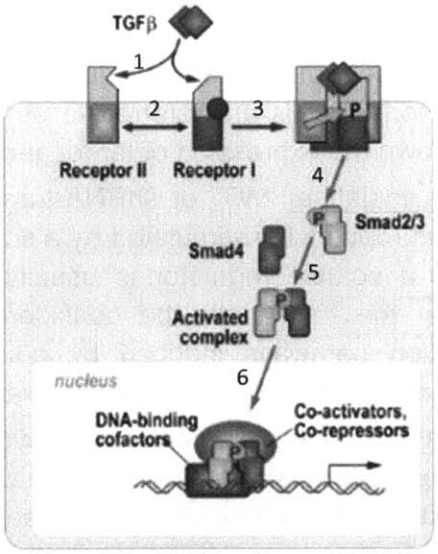


Figure 3.2.3: Overview of the TGFβ pathway. Left: List of the core proteins of the TGFβ pathway. Right: Steps of cell stimulation by TGFβ isoforms via the SMAD pathway (edited from [Derynck and Zhang 2003]).

TGFβ isoforms in Wound Healing

The three isoforms of the transforming growth factor beta (TGFβ1, 2 and 3) are members of the TGFβ superfamily [Derynck and Miyazono 2007]. Although the three isoforms are very similar in primary structure [Massagué 1990], they are associated with antagonistic effects in wound healing. In animal models of skin wound healing, TGFβ1 is associated with wound repair via wound contraction and scar formation, TGFβ2 appears to synergize with TGFβ1, and TGFβ3 is associated with improved regeneration. TGFβ1 is a prerequisite for MFB differentiation [Grinnell 1994, Klass et al. 2009], promotes chemotactic recruitment of fibroblasts in skin wounds [Grinnell 1994]. Neutralizing antibodies against TGFβ1 and TGFβ2 reduce scar formation in rat

dermal wounds [Shah et al. 1995]. On the other hand, TGFβ3 promotes epidermal cell migration, halts dermal cell migration, and down-regulates MFB differentiation [Bandyopadhyay et al. 2006; Waddington et al. 2010]. Transient overexpression of TGFβ3 in rat lungs initiates pro-fibrotic effects similar but less severe than TGFβ1, by less drastic inhibition of matrix degradation and gene regulation of SMAD pathway components (TGFβ3 down-regulates TGFβ1, up-regulates SMADs, but does not affect ALK1 and ALK5) [Ask et al. 2008].

3.3 Methods

3.3.1 Overview of the Proposed Image Informatics Methodology

This section describes an image informatics methodology for studying signal transduction pathways in cells interacting with a 3D matrix *in vitro*. The signaling pathway is considered as a system that is stimulated by external inputs (soluble molecules or insoluble matrix), see Figure 3.2.1. The response of the system is quantified by spectral multi-photon 3D imaging, complimented by biochemical assays. The role of individual components of the signal transduction pathway is inferred by quantifying how the system response to particular stimuli changes when one or more components of the pathway are knocked down (Figure 3.2.2). The key components of the methodology (Figure 3.3.1) are:

System perturbation: remove (or significantly down-regulate) a component of the pathway under study. Here, stable knock-down of a protein component of the pathway is achieved by using lentivirus-based transfection to induce stable expression of shRNA that targets the mRNA of this protein. Normal cells are referred to as "wild type" (WT). Cells stably expressing shRNA that knocks down the expression of target gene X are referred to as "X^-" cells.

System stimulation: "WT" or shRNA-transfected X^- cells are seeded inside an appropriate 3D matrix and (optionally) stimulated by a soluble regulator (e.g. cytokine, growth factor, hormone). Although a soluble regulator is usually considered as the stimulant of signal transduction pathways, the matrix can be considered as another regulator of the pathway, as signal transduction pathways induced by adhesion receptors cross-talk with signaling pathways stimulated by receptors for soluble regulators [Gomperts et al. 2009].

Output quantification: The cell-matrix system is incubated for an appropriate period. Then the cell-matrix system is imaged in order to quantified its response by imaging-based assays. All imaging data were acquired using the spectral multi-photon microscope (So lab, Department of Biological Engineering, MIT) described in Section 2.3. Alternatively, some cell-matrix samples can be lysed and analyzed by immunoblotting.

Image Processing: Imaging data are processed first by the generic image processing pipeline described in Chapter 2, and then by an application-specific pipeline that identifies and quantifies cell objects. Image processing quantifies each TC by a feature array that includes the feature vectors for a number of cells.

Statistical Processing: Low-level statistical analysis quantifies the degree of "difference" (a metric of "distance") between the feature arrays of different TC. High-level statistical analysis combines these distance results with a-priori knowledge on the biology of the signal transduction pathway in order to infer conclusions about the role of pathway components.

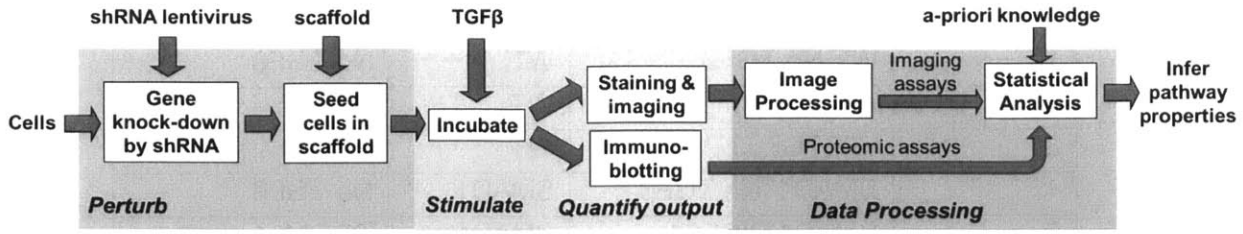


Figure 3.3.1: Schematic of the proposed image informatics methodology applied in the study of TGFβ isoform signaling via the SMAD pathway.

Advances Compared to the State-of-the-Art

- The developed methodology focuses on quantifying a signal transduction pathway in cells that interact with a 3D matrix (here: a porous collagen biomaterial). In contrast, existing image informatics studies focus on cells cultured on flat plastic surfaces that lack a physiologically relevant ECM.
- The signal transduction pathway is considered to be a part of an integrated cell-matrix system. The state of the system is described using imaging-based features that describe both cells (morphology, αSMA expression), and cell-matrix interactions (extent, topology). In contrast, existing image informatics studies quantify only cell morphology.
- Due to the complexity and the 3D nature of a cell-matrix system, imaging features are calculated based on spectral multi-photon microscopy images (Section 2.3). Multi-photon microscopy provides 3D imaging capability. Spectral microscopy enables to quantify both cell morphology features and cell-matrix adhesion features from the same image.
- The response of the cell-matrix system is quantified mostly by imaging-based assays. Imaging results are supplemented by a proteomic assay (immune-blotting).

3.3.2 Experimental Design

The proposed image informatics framework is demonstrated in a study of TGFβ signaling via the SMAD signal transduction pathway in human dermal fibroblasts seeded inside porous collagen scaffolds. The study focuses on identifying the role of various components of the SMAD pathway in the differential transduction of TGFβ1 and TGFβ3 stimuli.

The system of interest is stimulated by one of the two TGFβ isoform solutions. Perturbation of the system is implemented by knocking down components of the SMAD pathway via RNA interference (RNAi). Each treatment condition (TC) corresponds to a particular TGFβ stimulation and a particular system perturbation. After 5 days of incubation, the response of each cell-seeded scaffold is quantified by either spectral multi-photon imaging or by immunoblotting. Both assays focus on the expression of contractile cell phenotypes, particularly the expression of alpha smooth muscle actin (αSMA), an established marker of myofibroblasts (MFB).

The SMAD-mediated response of fibroblasts to TGFβ is a system well-suited for an image informatics study. TGFβ isoforms regulate MFB differentiation and can affect significantly both the shape of the cells (contractile cells are more elongated and contain stress fibers) and the way cells interact with the surrounding cells and matrix. Therefore, genetic perturbations against SMAD pathway components are expected to affect the system response that is detected by imaging (cell morphology, and cell-matrix adhesion topology).

TC #	Treatment Condition	Cell Type	Stimulation
1	WT No TGFβ	WT	No TGFβ
2	WT TGFβ1	WT	TGFβ1
3	WT TGFβ3	WT	TGFβ3
4	SMAD1$^-$ No TGFβ	SMAD1$^-$	No TGFβ
5	SMAD1$^-$ TGFβ1	SMAD1$^-$	TGFβ1
6	SMAD1$^-$ TGFβ3	SMAD1$^-$	TGFβ3
7	SMAD2$^-$ No TGFβ	SMAD2$^-$	No TGFβ
8	SMAD2$^-$ TGFβ1	SMAD2$^-$	TGFβ1
9	SMAD2$^-$ TGFβ3	SMAD2$^-$	TGFβ3
10	SMAD3$^-$ No TGFβ	SMAD3$^-$	No TGFβ
11	SMAD3$^-$ TGFβ1	SMAD3$^-$	TGFβ1
12	SMAD3$^-$ TGFβ3	SMAD3$^-$	TGFβ3
13	SMAD5$^-$ No TGFβ	SMAD5$^-$	No TGFβ
14	SMAD5$^-$ TGFβ1	SMAD5$^-$	TGFβ1
15	SMAD5$^-$ TGFβ3	SMAD5$^-$	TGFβ3
16	ALK1$^-$ No TGFβ	ALK1$^-$	No TGFβ
17	ALK1$^-$TGFβ1	ALK1$^-$	TGFβ1
18	ALK1$^-$ TGFβ3	ALK1$^-$	TGFβ3
19	ALK5$^-$ No TGFβ	ALK5$^-$	No TGFβ
20	ALK5$^-$ TGFβ1	ALK5$^-$	TGFβ1
21	ALK5$^-$ TGFβ3	ALK5$^-$	TGFβ3

Table 3.3-1: Overview of the treatment conditions (TC) for the pilot TGFβ SMAD signaling image informatics study.

3.3.3 Cell culture, RNA Interference and Stimulation

The cell-matrix system consists of primary human dermal fibroblasts (CC-2511 Lonza, Basel, Switzerland) seeded inside porous collagen GAG scaffolds. Fibroblasts were cultured at 37°C, 5% CO_2 in complete culture medium (high-glucose DME medium, 10% fetal bovine serum, 1% penicillin/streptomycin). The cell-matrix system was perturbed by knocking down one member of the SMAD pathway via shRNA. The members of the SMAD pathway that were knocked down in this study consist of two TGFBRI receptors (ALK1,5) and four SMAD proteins (SMAD1,2,3,5).

Stable knock-down was achieved by using lentivirus to stably transfect the "wild type" primary human dermal fibroblasts with the shRNA construct. Lentivirus containing shRNA plasmids were produced in HEK293T (Figure 3.3.2 left). Stable transfection of fibroblasts with shRNA lentivirus was done in the presence of 8 µg/ml hexadimethrine bromide (Sigma). Virus-treated cells were selected by 2 µg/ml puromycin (Sigma) treatment over 3 days. A puromycin killing curve concluded that 2 µg/ml is the minimal concentration of puromycin where non-transfected cells cannot survive. The appropriate multiplicity of infection (MOI; how many virus particles per cell are necessary for successful cell transfection) was determined to be 25~30 TU per fibroblast (1 TU = 10 physical viral particles) via puromycin screening with 2 µg/ml puromycin [Guo 2013]. In virus-transfected cells, the shRNA DNA incorporated in the cell DNA reduces significantly the expression of the target protein by preventing the translation of the transcribed mRNA using the

intrinsic RNA interference mechanism present in cells (Figure 3.3.2 right). The efficiency of all shRNA-mediated protein knock-down was estimated around 70-80% via immune-blotting [Guo 2013].

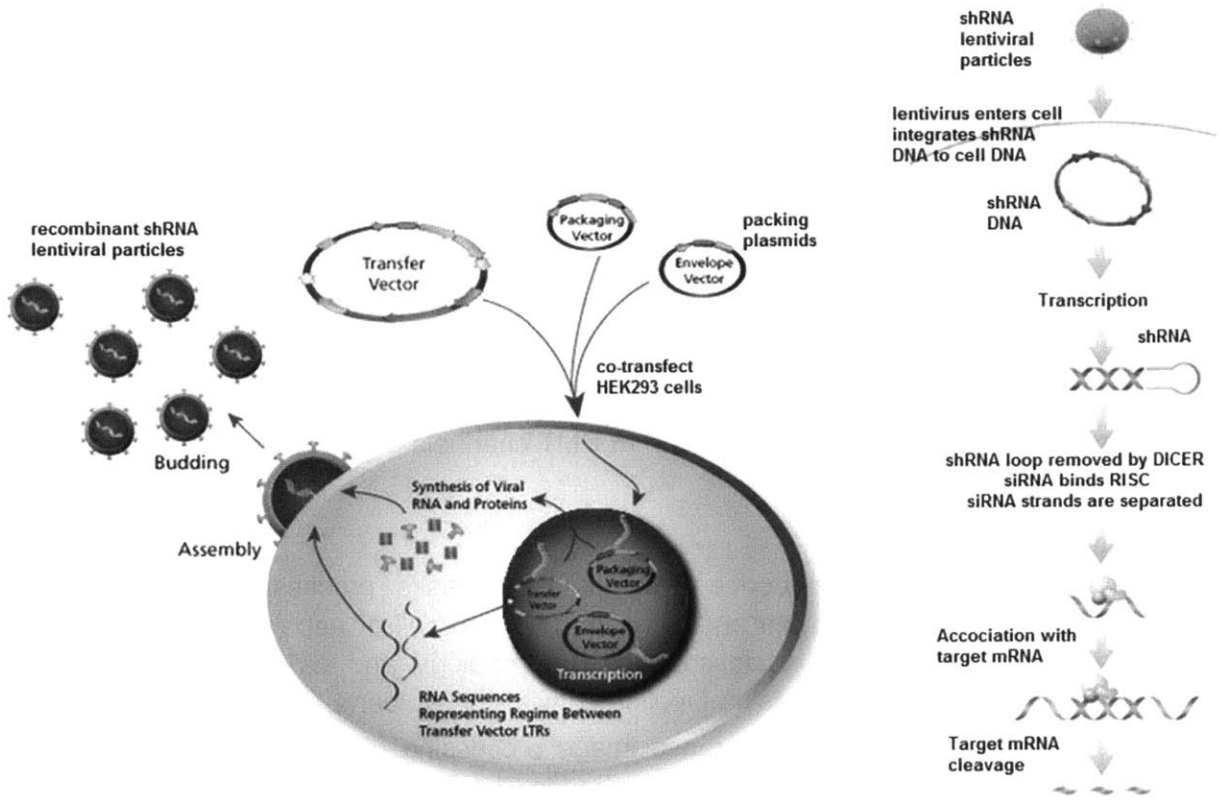


Figure 3.3.2: Knocking down the expression of one SMAD pathway component (target protein) by RNA interference implemented by stable shRNA expression in primary cells. Left: Recombinant expression of lentiviral particles that contain the shRNA DNA for the target protein (adapted from Sigma Aldrich). Right: Mechanism utilized by shRNA lentiviral particles to knock down the expression of the target protein via RNAi (adapted from www.scbt.com/).

Collagen-GAG scaffolds (0.5% solid fraction, approximately 96 μm mean pore diameter, cross-linked by DHT for 48h at 120˚C under 50 mTorr vacuum) were fabricated as described in Appendix E. Upon reaching 80% confluency, cells (wild type or transfected with shRNA) were trypsinized and seeded inside the scaffold by placing a scaffold sample in contact with a 20μl drop of 2000 cells/μl suspension on an agarose-coated well. The cell-seeded scaffold was incubated 1h at 37˚C, 5% CO_2 before adding more culture medium in order to let cells attach to the scaffold. After overnight incubation, the medium was replaced by fresh culture medium, which could be supplemented with no TGFβ isoform, or 5 ng/ml TGFβ1 or 5 ng/ml TGFβ3 (Peprotech, Rocky Hill, NJ) in order to stimulate the SMAD pathway. Cell-seeded scaffolds were incubated 5 days, during which the culture medium was replaced once on the third day.

3.3.4 Staining and Imaging

Cells inside cell-seeded scaffolds were stained by incubating with 25μM cell-tracker Orange CMTMR (C2927, Invitrogen) in DMEM for 45 min at 37˚C and 5% CO_2, incubating 45 min more in DMEM, and washing by ice-cold PBS twice. Samples were then fixed for 15 min with 4% paraformaldehyde, permeabilized with 0.25% Triton X100 in PBS for 15 min, blocked by 5%

BSA in 0.25% PBST for 1.5 h at room temperature, treated with mouse anti-αSMA primary antibody (A2547 Sigma) diluted 1:400 in antibody diluent (1% BSA in 0.25% PBST) at 4˚C overnight, washed by PBS three times for 10 min at room temperature, treated with 1:200 alexa fluor 488 conjugated anti-mouse secondary antibody (715-545-150, Jackson Immunoresearch, West Grove, PA) in antibody diluent for 1.5 h at room temperature, counterstained with 20 nM Hoechst 33342 (H21492, Invitrogen) in PBS at 37 ˚C for 30 min, and washed three more times with PBS for 10 min at room temperature.

3D images of stained cell-seeded scaffolds were acquired using the spectral multi-photon microscope described in Section 2.3. Specimen were excited by 16 mW 775nm femtosecond pulsed laser through a Zeiss 40× C-apochromat 1.2 NA objective. 3D imaging over a region 500×500 μm wide and 30μm thick was achieved by acquiring multiple images using a motorized sample stage (Prior Scientific, Cambridge, UK) and a piezoelectric objective actuator (P-721.00, Physik Instrumente, Karlsruhe, Germany). Each 256×256 pixel (40 μs pixel sampling, 110×110 μm field of view) contains 16 channels per pixel, where each channel detects photons within a 13 nm wide range of the spectrum, so that the 16 PMT channels detected photons in the [380, 590] nm range.

3.3.5 Image Processing

Image processing consists of a sequence of computational tasks (processing pipeline, Figure 3.3.3) that analyzes raw imaging data and provides for each cell of each treatment condition (TC) a N-sized feature vector that describes its morphology and its interaction with the surrounding matrix. The image processing pipeline consists of two parts.

The first part consists of the general-purpose image processing pipeline (image registration, spectral unmixing and image segmentation) described in Chapter 2. Images contain regions that belong to four classes. Three of them contain a distinct set of 2 emission sources: the "cytoplasm" class (CMTMR, Alexa fluor 488), the "nucleus" class (CMTMR, Hoechst 33342) and the "scaffold" class (fluorescence and second harmonic emission by collagen). The fourth class is "void" (empty space between cells and scaffold) whose emission is modeled as instrument noise. The set of emission sources present in the cells is similar to the alexa350-alexa488-CMTMR system described in Section 2.4.7 which was shown to be easy to spectrally unmix accurately. The two sources present in the matrix are also easy to spectrally unmix because >90% of SH is concentrated in channel 1, where collagen fluorescence emission is practically zero. Since Hoechst33342 and collagen fluorescence have similar spectral signatures, straightforward application of spectral unmixing would result in significant errors.

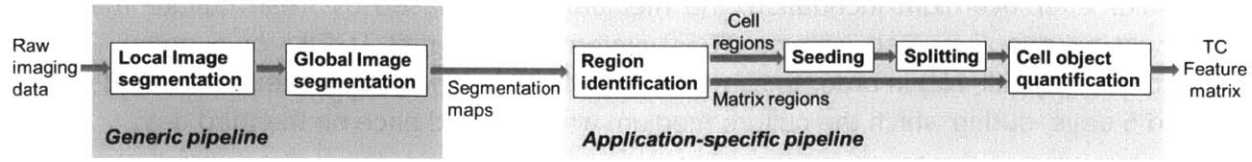


Figure 3.3.3: Schematic of image processing steps for calculating the feature matrixc that describes each treatment condition.

The second part of the image processing pipeline consists of three application-specific processing tasks: region identification, cell identification, and cell quantification.

Region identification organizes neighboring pixels of identical class ("nucleus", "cytoplasm", "matrix", "void") into regions. Due to the nature of the classes present, nuclei regions are surrounded by cytoplasm regions, which are surrounded by either matrix or void regions.

Cell identification identifies cell objects, each consisting of one nucleus and one cytoplasm region. In many cases cells are in contact with other cells, so a single cytoplasm region may enclose multiple nuclei. In this case the cytoplasm region is segregated into multiple cell objects using the "seeding" and "splitting" tasks described in Section 3.2.2. Here, seeding is implemented by the h-min morphological filtering algorithm described in [Cheng and Rajapakse 2009]. This seeding approach usually provides excess seeds due to the low SNR at low signal level. Seeds corresponding to nuclei are usually easy to identify because it is relatively easy to identify cell objects. However, in 3D imaging it is possible that the number of cells present in a plane is more than the number of nuclei present in the same imaging plane (some nuclei may be located in different planes). The user needs to manually pick the correct seeds in the cytoplasm based on intensity contours. Splitting of cell cytoplasm regions is implemented using a Voronoi segmentation scheme that utilizes a modified distance metric that incorporates gradient information [Jones et al. 2005]. The user can exclude cell objects whose cytoplasm segmentation clearly appears wrong. Only cell objects that contain a nucleus are including in the following quantification step.

Cell object quantification: The third step of the processing pipeline calculates a series of N assays (features) that describe each cell object. The N-dimensional feature vector of each cell consists of features that describe the morphology of the cytoplasm, the morphology of the nucleus, and the morphology of cell-matrix proximity. The features that describe the morphology of the nucleus utilize the emissions based on the nucleic and cytoplasmic stains, and the expression of contractile phenotypes based on the immunofluorescence signal for αSMA. Matrix features describe the extent and spatial distribution of cell-matrix contact sites (based on proximity of pixels classified as cytoplasm and matrix). A detailed description of features is provided in Table 3.3-2. The feature vectors of all cells correspond to the feature matrix of the treatment condition.

Cytoplasm Morphology		Nucleus Morphology		Cell Microenvironment	
Feature	**Meaning**	**Feature**	**Meaning**	**Feature**	**Meaning**
CA	Area [μm^2]	NA	Area [μm^2]	CMPA	Cell-Matrix Proximity Area [um2]
CP	Perimeter [μm]	NP	Perimeter [μm]	CMPAF	Cell-Matrix Proximity Area Fraction
CSF	Shape factor	NSF	Shape factor	CMPAIA	Cell-Matrix Proximity Alexa488 Int. Average
CE	Eccentricity	NE	Eccentricity	CMPAIS	Cell-Matrix Proximity Alexa488 Int. Std Dev
CPNPR	Cell Perimeter to Nucl. Perimeter Ratio	NTIA	Total intensity average [PC]	CMCCO	Cell-Matrix Proximity Cell Centroids Offset
CTIA	Total Intensity Average [PC]	NTIS	Total intensity std. dev. [PC]	CVPA	Cell-Void Proximity Area [um2]
CTIS	Total Intensity std dev [PC]	NHIA	Hoechst Intensity average [PC]	CVPAF	Cell-Void Proximity Area Fraction
CAIA	Alexa488 Intensity Average [PC]	NHIS	Hoechst Intensity std. dev. [PC]	CVPAIA	Cell-Void Proximity Alexa488 Int. Average
CAIS	Alexa488 Intensity std dev [PC]	NCIA	CMTMR Intensity average [PC]	CVPAIS	Cell-Void Proximity Alexa488 Int. Std. Dev.
CCIA	CMTMR Intensity average [PC]	NCIS	CMTMR Intensity std. dev. [PC]	CVCCO	Cell-Void Proximity Cell Centroids Offset
CCIS	CMTMR Intensity std. dev. [PC]	NGCO	Grayscale centroid offset [μm]	CCPA	Cell-Cell Proximity Area [um2]
CNIAR	Cytoplasm to Nucleus Int. Average Ratio	NCO	Nucleus-cell centroids offset [μm]	CCPAF	Cell-Cell Proximity Area Fraction
CNCAR	Cytoplasm to Nucleus CMTMR Average Ratio			CCPAIA	Cell-Cell Proximity Alexa488 Int. Average
CGCO	Grayscale Centroid Offset [um]			CCPAIS	Cell-Cell Proximity Alexa488 Int. Std. Dev.
CCO	Cell Centroids Offset [um]			CCCCO	Cell-Cell Proximity Cell Centroids Offset
				CMPF	Cell-Matrix Interface / Perimeter Fraction
				CVPF	Cell-Void Interface / Perimeter Fraction
				CCPF	Cell-Cell Interface / Perimeter Fraction

Table 3.3-2: Imaging-based features utilized in quantifying individual cells. 15 features describe the cell cytoplasm (morphology, αSMA expression). 12 features describe the cell nucleus (morphology). 18 features describe the cell immediate environment (cell-matrix, cell-cell, cell-void).

3.3.6 Statistical Analysis

Following [Nir et al. 2010], statistical analysis takes place in two steps (Figure 3.3.4). Low-level statistical analysis compares different treatment conditions (TC) by evaluating the degree of "similarity" between their feature arrays. High-level statistical analysis combines the results of the low-level statistical analysis with a-priori knowledge of the biological system to infer relationships related to specific biological questions. This study focuses on the low-level statistical analysis part, and specifically in quantifying how each shRNA knock-down affects the differences between cell responds to the three treatments (No TGFβ, TGFβ1, TGFβ3).

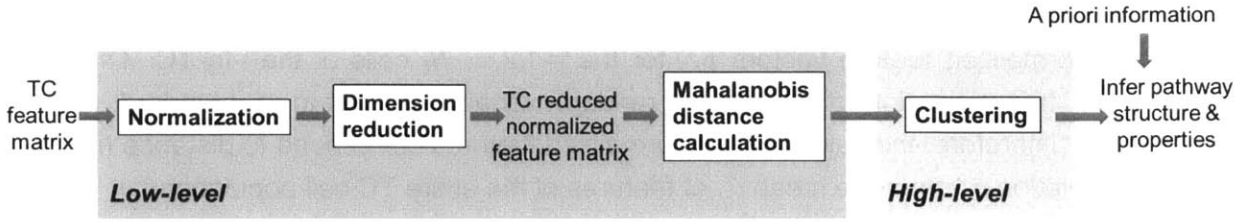


Figure 3.3.4 Schematic of the steps utilized in statistical analysis.

Low level statistical analysis

The output of each treatment condition (TC) describes how a particular cell-scaffold system responds to a particualr stimulus. The outcome of the i-th TC is described by the feature matrix

$$\boldsymbol{F}_i = [\boldsymbol{f}_{i,1} \quad \cdots \quad \boldsymbol{f}_{i,N_i}] = [\boldsymbol{v}_{i,1} \quad \cdots \quad \boldsymbol{v}_{i,N_F}]^T$$

$\boldsymbol{F}_i$ is a $N_F \times N_i$ matrix whose columns $\boldsymbol{f}_{i,j}$ are the feature vectors of the N_i cells of the TC, calculated based on the procedure described in Section 3.3.5. The N_i elements of the j-th row $\boldsymbol{v}_{i,j}$ is a sample of the random variable V_{ij} that describes the j-th feature of i-th TC. $\boldsymbol{v}_{i,j}$ can be used to estimate the statistic distribution of $V_{i,j}$. The larger the number of cells N_i quantified per TC, the more accurate is the estimation of the statistical distribution of $V_{i,j}$.

When the responses of two TC differ significantly, then many of the statistical distributions $V_{i,j}$ will also differ. The objective of low-level statistical analysis is to quantify the degree of "difference" between the responses of different TC, or equivalently, quantify how different are the statistical distributions that describe $V_{i,j}$. This can be challenging due to the large number N_T of treatment conditions, the different nature of the various features included in the analysis, the large dimension N_F of the feature vector, and the large number N_i of cells present in each TC.

The low-level statistical analysis utilized in this study follows closely the approach described in [Nir et al. 2010] and consists of three steps:

Feature normalization:

The response of each TC is quantified based on a set of imaging-based features (Table 3.3-2). Only cells that do not contain outliers are considered in the following analysis. For each TC and for each feature, outliers are defined as data outside the range $[F_L - 1.5d_F, F_U + 1.5d_F]$ where F_L, F_U are the 25th and 75th percentiles, and $d_F = F_U - F_L$ is the interquantile range [Härdle and Simar 2003].

The features shown in Table 3.3-2 describe a wide range of attributes, do not have common units and their numerical values may differ by orders of magnitude. In order for each feature to

contribute equally to the "distance" metric it is necessary to non-dimensionalize all features. Here, the j-th feature of the i-th TC is non-dimensionalized as:

$$Z_{i,j} = \frac{V_{i,j} - \hat{\mu}_j}{\hat{\sigma}_j}$$

$\hat{\mu}_j$ and $\hat{\sigma}_j$ are the estimates of the mean and standard deviation of the j-th feature over the union of cell populations of all treatment conditions (union of the j-th row $\boldsymbol{v}_{i,j}$ of all $\boldsymbol{F}_i$). The resulting normalized $N_F \times N_i$ feature matrix:

$$\tilde{\boldsymbol{F}}_i = \left[\tilde{\boldsymbol{f}}_{i,1} \quad \dots \quad \tilde{\boldsymbol{f}}_{i,N_i}\right] = \left[\boldsymbol{z}_{i,1} \quad \dots \quad \boldsymbol{z}_{i,N_F}\right]^T$$

Contains the normalized feature vectors $\tilde{\boldsymbol{f}}_{i,j}$ for the j=1,2,..., N_i cells of the i-th TC. This step resembles the transformation of a normal random variable $V_{i,j}$ into a standard normal distribution $Z_{i,j}$. Therefore, the values of the normalized features correspond to distance (in units of standard deviation $\hat{\sigma}_j$) from the mean $\hat{\mu}_j$ of features of the entire TC cell population.

Dimension reduction:

The image processing pipeline quantifies each cell using a feature vector of large dimension N_F. The reason to include a large number of features is that it is not known a-priori which features contain useful information and can distinguish the outcome of different TC. It is possible that many features do not contain useful information and just vary randomly. It is also possible that the values of many features are highly correlated, creating redundancy. The presence of such redundant features is not known a-priori. Therefore, in order to decrease computational cost, reduce complexity, increase accuracy and perhaps gain insight on critical features of the system, it is advised to reduce the dimension of the feature vector. Although there are multiple mathematic methods for vector dimension reduction (see Section 3.2.1), this work the projection method of Principal Component Analysis (PCA), a computationally efficient method that has been applied extensively in the signal processing literature. PCA decomposes the augmented normalized feature matrix $\tilde{\boldsymbol{F}}$ (the union of the $\tilde{\boldsymbol{F}}_i$ arrays of al TC) into a new coordinate system described by the $N_F \times N_F$ orthogonal matrix $\boldsymbol{P}$:

$$\tilde{\boldsymbol{F}} = \left[\tilde{\boldsymbol{F}}_i \quad \dots \quad \tilde{\boldsymbol{F}}_{N_T}\right] = \tilde{\boldsymbol{P}} \cdot \tilde{\boldsymbol{Q}}$$

The columns of matrix $\tilde{\boldsymbol{P}}$ contain the axes of the new coordinate system by order of decreasing variance [Hastie et al. 2009]. $\tilde{\boldsymbol{Q}}$ contains the projections of $\tilde{\boldsymbol{F}}$ in the PCA axes $\tilde{\boldsymbol{P}}$. Dimension reduction is achieved by keeping a few N_R "significant" rows of $\boldsymbol{P}$:

$$\tilde{\boldsymbol{F}} \approx \boldsymbol{P} \cdot \boldsymbol{Q}$$

Where $\boldsymbol{P}$, $\boldsymbol{Q}$ contain the first N_R rows of $\tilde{\boldsymbol{P}}$, and the first N_R rows of $\tilde{\boldsymbol{Q}}$ respectively. Unfortunately, the axes of the reduced-dimension system (columns of $\boldsymbol{P}$) are linear combinations of the normalized features $Z_{i,j}$ and therefore usually they have no physical meaning. This is the major disadvantage of PCA.

Eventually, for each TC dimension reduction converts its normalized feature matrix $\tilde{\boldsymbol{F}}_i$ into a reduced normalized $N_R \times N_i$ feature matrix $\boldsymbol{Q}_i$, which describes the outcome of the i-th TC.

$$\boldsymbol{Q}_i = \left[\boldsymbol{q}_{i,1} \quad \dots \quad \boldsymbol{q}_{i,N_i}\right] = \left[\boldsymbol{n}_{i,1} \quad \dots \quad \boldsymbol{n}_{i,N_R}\right]^T$$

Mahalanobis Distance calculation:

The j-th row $\boldsymbol{n}_{i,j}$ of $\boldsymbol{Q}_i$ contains a sample for the j-th random variable $N_{i,j}$ that describes the

value of the j-th reduced feature in the i-th TC. The "similarity" between different TC response is quantified based on how different are the statistical distributions for $N_{i,j}$ for different TC.

It is assumed that for the i-th TC, the $N_R \times 1$ random vector $\boldsymbol{N}_i = [N_{i,1} \quad \cdots \quad N_{i,N_R}]^T$ that describes the values of its reduced normalized features can be modeled as a multivariate normal distribution $N(\boldsymbol{\mu}_i, \boldsymbol{C}_i)$ of mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{C}_i$ that can be estimated as:

$$\widehat{\boldsymbol{\mu}}_i = [\hat{\mu}_{i,1} \quad \cdots \quad \hat{\mu}_{i,N_R}]^T$$

$$\widehat{\boldsymbol{C}}_i = \frac{1}{N_i}\boldsymbol{Q}_i \cdot \boldsymbol{Q}_i^T$$

where $\hat{\mu}_{i,k}$ is the mean of the k-th row of the $\boldsymbol{Q}_i$ matrix.

The "fitness" of the measured output of the i-th TC (described by the feature matrix $\boldsymbol{Q}_i$) for the j-th TC can be quantified by the Mahalanobis distance of the feature matrix $\boldsymbol{Q}_i = [\boldsymbol{q}_{i,1} \quad \cdots \quad \boldsymbol{q}_{i,N_i}]$ from the multivariate normal distribution $N(\boldsymbol{\mu}_j, \boldsymbol{C}_j)$ [Hastie et al. 2009]:

$$m_{i,j} = \frac{1}{N_i}\sum_{k=1}^{N_i}\{\sqrt{(\boldsymbol{q}_{i,k} - \widehat{\boldsymbol{\mu}}_j)^T \widehat{\boldsymbol{C}}_j^{-1}(\boldsymbol{q}_{i,k} - \widehat{\boldsymbol{\mu}}_j)}\}$$

In multivariate normal distributions, the Mahalanobis distance is the corresponding entity to the non-dimensional distance $z = \left|\frac{x-\mu}{\sigma}\right|$ used in the standard normal distribution to describe the "distance" of a measurement x from the mean μ given variance σ^2. Therefore $m_{i,j}$ describes the average distance of the cell measurements of the i-th TC from the multivariate normal distribution that models the j-th TC. In general $m_{i,j} \neq m_{j,i}$, so $m_{i,j}$ by itself cannot be used as a "distance" metric. Instead, the degree of "difference" between the reduced normalized features of cells from two different TC is quantified by the following Mahalanobis mean distance metric:

$$D_{i,j} = \frac{m_{i,j} + m_{j,i}}{2 \cdot N_R}$$

where N_R is the dimension of the reduced normalized feature space.

Low-level analysis provides for each TC the reduced normalized feature matrix $\boldsymbol{Q}_i$ and the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{C}_i)$ that fits best the data in $\boldsymbol{Q}_i$. The difference between the response of two TC's are described by the distance metric $D_{i,j}$.

High level statistical analysis

This pilot study does not provide a detailed high-level analysis or a mathematical model that attempt to interpret the distance $D_{i,j}$ between different TC. For the c-th kind of cell (WT or X^-) low-level statistical analysis provides a 3×3 cell response matrix $\mathcal{D}_c$ that contains the distances $D_{i,j}$ between the 3 TC that describe the cell response to the stimuli (No TGFβ, TGFβ1, TGFβ3). Some qualitative intuition about the effects of each shRNA knock down can be obtained by studying how the off-diagonal elements of $\mathcal{D}_c$ (the elements corresponding to $D_{i,j}$ between different stimuli) change compared to the response of WT cells.

Additional intuition is provided by considering the response of different kinds of cells to identical stimulus. This is implemented by clustering the mean Mahalanobis distance $D_{i,j}$ between the corresponding TC utilizing the shortest distance algorithm [Hastie et al. 2009].

3.3.7 Immuno-blotting

After 5 days of incubation with TGFβ1, TGFβ3, or no TGFβ, cell-seeded scaffolds were washed twice with ice-cold PBS with gentle shaking, and lysed by adding 25 µl ice-cold lysis buffer (50mM Tris HCl pH 7, 150mM NaCl, 5mM EDTA, 0.5% NP40, 10 µl/ml protease inhibitors, 2mM $NaVO_3$, 10mM NaF) per scaffold and incubating 1.5 hour at 4°C while rotating. Protein concentration in the lysates was measured by BCA assay (Protocol J.2). Lysates of equal protein amount in reducing Laemmli buffer were boiled at 95°C for 5 min, loaded and run in a Tris-HCl gel (Bio-Rad), transferred to PVDF membranes (Millipore), blocked with 3% BSA in TBST at room temperature for 2h, treated with 1:500 anti-αSMA antibody (A2547, Sigma) or 1:10000 anti-αtubulin antibody (ab7291, abcam) in antibody diluent (1% BSA in TBST) overnight at 4°C, washed three times in TBST, and incubated with HRP-conjugated secondary antibody (1:1E5 for αSMA, 1:4E for αtubulin, 31340, Thermo Scientific) in antibody diluent for 1 hour at room temperature. Membranes were washed four times in TBST, treated with 500µl chemiluminescence substrate (34095, Thermo Scientific) and imaged in a FluorChem 8900 station (Alpha Innotech). Blot images were quantified by densitometry using the appropriate plugin of ImageJ software (National Institute of Health, Bethesda , MD).

3.3.8 Research Question

Both TGFβ1 and TGFβ3 isoforms are involved in several important pathological processes including inflammation, wound healing and fibrosis. Although there is a significant body of literature about TGFβ1, the effects of TGFβ3 and in particular how it stimulates cells different than TGFβ1 are not well understood. Few studies compare how these isoforms induce different effects to cells. Although the TGFBR-SMAD cascade is known to be the major pathway that regulates MFB differentiation [Gu et al. 2007; Usuki et al. 2012], up to date it is not known how this pathway transduces stimuli by different TGFβ isoforms (particularly TGFβ1 and TGFβ3) and whether this results in different MFB differentiation. It is not known whether different components of the SMAD pathway regulate differential TGFβ1 versus TGFβ3 stimuli response, and whether this is related to MFB differentiation. This study was chosen due to the suggested critical role of TGFβ isoforms in the expression of contractile phenotypes during wound healing, a phenotype that is suggested to affect significantly the outcome of wound healing in injured organs [Singer and Clark 1999; Yannas 2001; Douglas 2010].

3.4 Results

3.4.1 The need to Include the Surrounding Matrix in Cell Signaling Studies

Figure 3.4.1 shows representative images of contractile cells in 2D *in vitro* cultures, 3D *in vitro* cultures, and *ex vivo* animal samples. Figure 3.4.1a shows differentiated myofibroblasts cultured on glass surfaces, and Figure 3.4.1b shows differentiated myofibroblasts cultured inside porous collagen-GAG scaffolds. In both cases myofibroblasts were derived by differentiating fibroblasts by a 5-day treatment with 5 ng/ml TGFβ1. Figure 3.4.1c shows contractile cells interacting with porous collagen scaffolds in *ex vivo* samples of transected rat sciatic nerves grafted with collagen scaffold (type D, see Chapter 4).

The images of Figure 3.4.1 show a dramatic difference between the morphology of contractile cells in 2D and 3D micro-environments. Myofibroblasts grown in 2D are extremely flat, thin and large (dimensions of cell axes are more than 50 µm), and their nuclei are located above the

main body of the cell, resembling in shape fried eggs. Large stress fibers containing αSMA span their length ending in focal adhesions [Hinz 2006, 2007]. Myofibroblasts cultured inside a 3D matrix environments are spindle-shaped, their long axis is approximate 20-50 μm long, and their nuclei are located in the center of the cell surrounded by actin fibers. The morphology of cells grown inside porous scaffolds *in vitro* appear very similar to the morphology of cells interacting with similar scaffolds *in vivo*. Since the insoluble environment of cells is a major modulator of their phenotypes (see also Section 5.2.1) it is expected that cells will behave differently in the 2D and 3D cell cultures [Grinnell 2003; Griffith and Swartz 2006; Green and Yamada 2007], making the porous collagen scaffold a much more biologically relevant system to study signal transduction in cells.

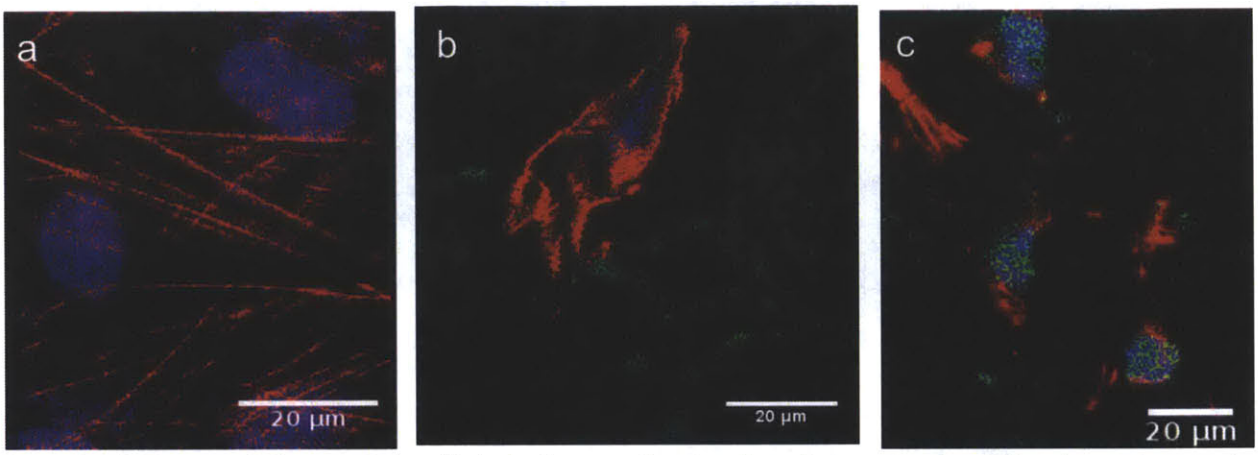


Figure 3.4.1: Representative spectral multi-photon images of contractile cells in various insoluble microenvironments. a: contractile cells cultured on glass surfaces *in vitro*, b: contractile cells cultured inside collagen scaffolds *in vitro,* c: *ex vivo image* of contractile cells inside a collagen scaffold during nerve regeneration (see Chapter 4). Blue: DAPI (cell nuclei), Green: collagen scaffold, Red: phalloidin (actin).

3.4.2 Image Processing

Figure 3.4.2 shows representative images of results at various steps of the image processing pipeline that converts raw spectral multi-photon microscopy data into cell objects surrounded by matrix. Figure 3.4.2a shows an intensity image obtained by summing the photon counts of the 16 channels of the spectral image. Cells can be distinguished from the surrounding scaffold struts because total cell emission is larger than total scaffold emission. Nevertheless, intensity by itself cannot be utilized to accurately distinguish cells from the matrix due to the inhomogeneity of emission throughout the cell bodies and the presence of significant noise. Instead, the various regions in the image are resolved based on the emission properties of their sources as described in Chapter 2. Figure 3.4.2b shows a HSV pseudo-color maximum intensity projection image of spectral unmixing results. Here spectral unmixing calculations consider for each pixel all possible sources present in the image. The emission of the scaffold is dominated by collagen auto-fluorescence (green). The emission of cells contains mostly CMTMR (yellow) and alexa488 (red) that is conjugated to the secondary antibody against the primary anti-αSMA antibody. Few nucleus pixels are dominated by the Hoechst33342 nucleic acid stain (blue). Spectral unmixing (for each domain separately) is the basis of the image segmentation scheme described in Chapter 2. Figure 3.4.2c shows the corresponding image segmentation map

provided by global image segmentation. Green corresponds to "scaffold" pixels, red to "cytoplasm" pixels, blue to "nucleus" pixels and black to "void" pixels.

In general cell-scaffold imaging reveals that fibroblasts and myofibroblasts interact extensively with their surrounding matrix. Although a few clusters of cells of moderate size are present (for example the one located in the bottom left part of Figure 3.4.2c), neither in WT cells nor in any one of the shRNA-treated cells was observed a tendency to cells to form cell niches.

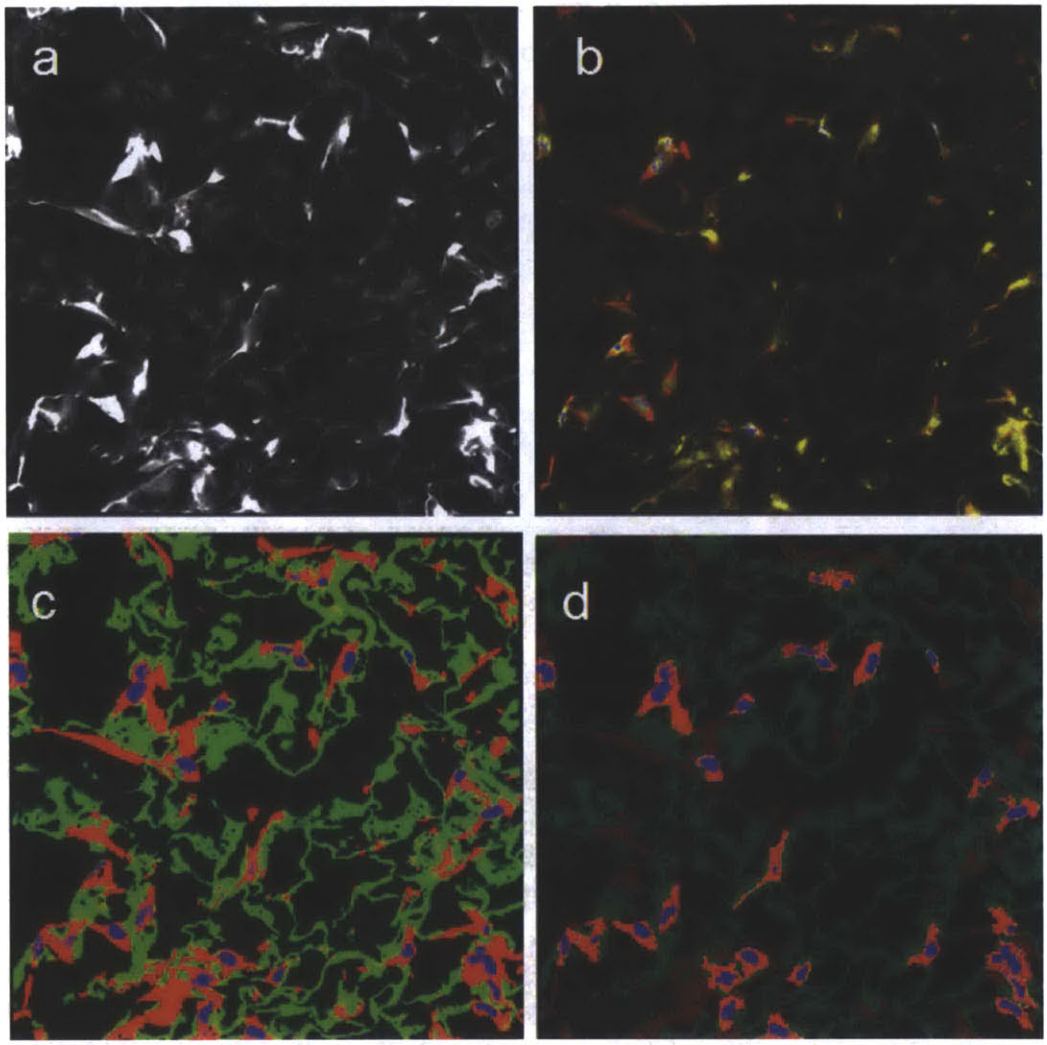


Figure 3.4.2: Representative images of the various steps in the image processing pipeline of the image informatics study. a: intensity data (sum of the signal from all 16 channels). b: Maximum intensity projection of spectral unmixing outcome. Red: Alexa fluor 488 conjugated secondary antibody against αSMA primary antibody, yellow: CMTMR cell tracker, blue: Hoechst 33342 dye, green: collagen auto-fluorescence. c: outcome of the global image segmentation algorithm. Blue: nuclei, Green: matrix, Red: cytoplasm. d: Selected cell objects to be quantified. The field of view of each image is approximately 400×400 μm.

The segmentation map shown in Figure 3.4.2c is the outcome of the generic image processing pipeline described in Chapter 2, and the input to the application-specific image processing pipeline described in Section 3.3.5. Figure 3.4.2c highlights some of the challenges faced when processing spectral images of cells inside 3D matrix environments. For example, some cytoplasm regions contain multiple nucleus regions (corresponding to cells that are in contact),

while others contain no nucleus region (nucleus is located in another plane). Figure 3.4.2d shows the outcome of the application-specific image processing pipeline. While many steps of the application-specific image processing are automatic (identification of nuclei regions and nuclei in contact, seeding in cytoplasm regions in the presence and absence of nuclei, and cell splitting) the pipeline still requires user input at two points. First, the user needs to correct the outcome of cell seeding, since the significant presence of noise induces some errors. Second, the user needs to pick appropriate cell objects where the outcome of cell splitting seems meaningful. The result is that usually a small percent of the cells (less than 30%) present in each field satisfies the necessary conditions and is selected for further processing. The remaining cells are not selected primary due to three main reasons: i) The cell cytoplasm lacks a nucleus (nucleus is located in another plane), ii) The cell is part of a big cell cluster and the outcome of cell seeding/splitting calculations does not look convincing, iii) other imaging artifacts. In Figure 3.4.2d, the selected cells are highlighted in bright red and blue. Each one of these cell objects is then quantified by the feature vector described in Table 3.3-2.

Figure 3.4.3 shows representative results of class evidence maps for the six TC corresponding to ALK1$^-$ cells and SMAD3$^-$ cells. Figure 3.4.4 shows the selected identified cell objects that correspond to the images shown in Figure 3.4.3. These figures show that the type of cell and the type of stimulus affects the observed response. For example TGFβ3 increases cell proliferation in ALK1$^-$ cells (which is quantified by the fraction of cell perimeter in contact with other cells) much more than SMAD3$^-$. TGFβ1 decreases cell size in SMAD3$^-$ cells more than ALK1$^-$ cells.

It should be noted that even in the case where an imaging field contains a large number of cells (e.g. the TGFβ3 treatment of ALK1$^-$ cells) the cells do not form clusters and keep their extensive cell-matrix adhesion pattern.

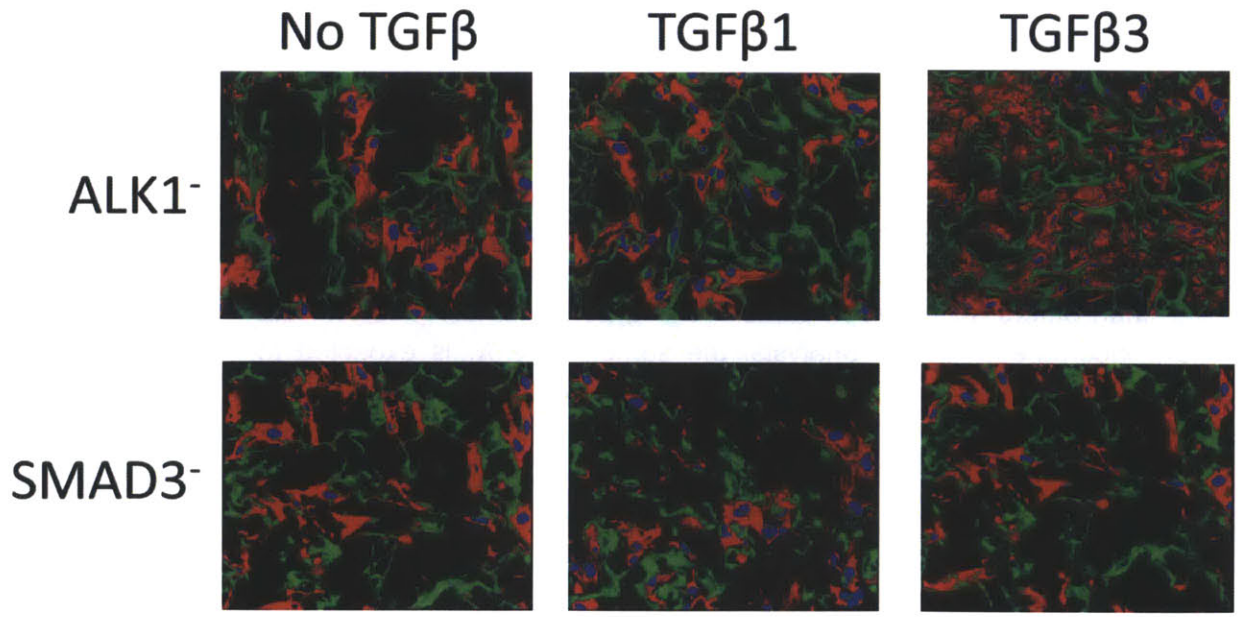


Figure 3.4.3: Representative image segmentation maps of six treatment conditions. Top row: SMAD3$^-$ cells. Bottom row: ALK1$^-$ cells. Left column: No TGFβ stimulus. Middle column: TGFβ1 stimulus. Right column: TGFβ3 stimulus. Red: cell cytoplasm. Green: collagen scaffold. Blue: cell nuclei.

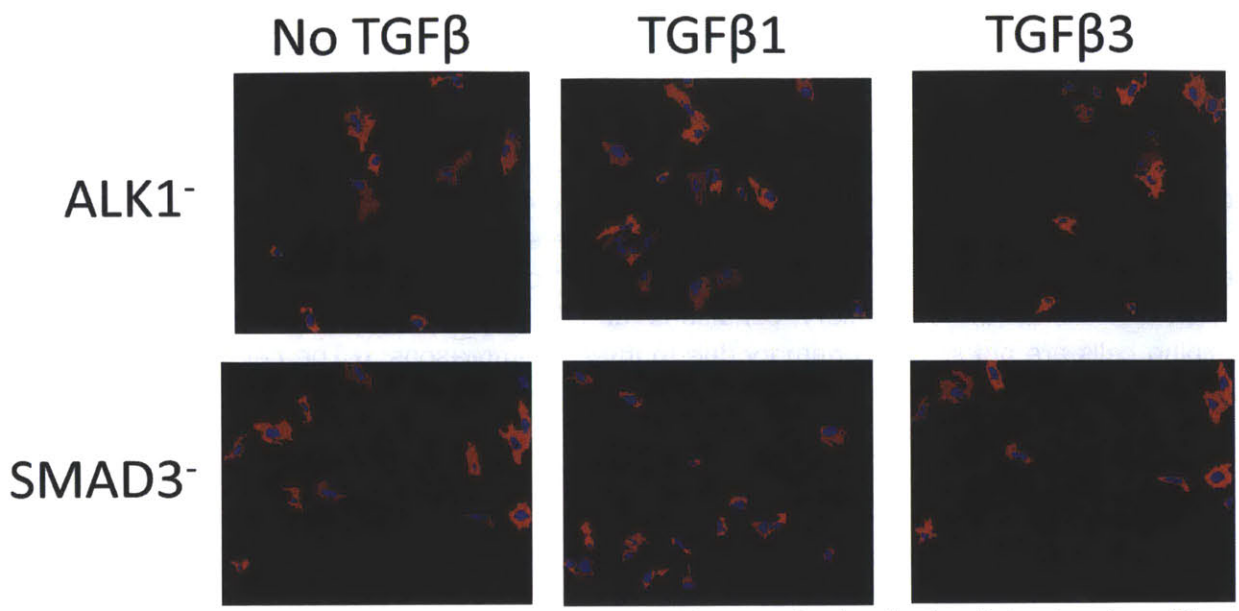


Figure 3.4.4: Representative images of cell objects selected for quantification for the six treatment conditions of Figure 3.4.3. Top row: SMAD3$^-$ cells. Bottom row: ALK1$^-$ cells. Left column: No TGFβ stimulus. Middle column: TGFβ1 stimulus. Right column: TGFβ3 stimulus. Red: cell cytoplasm. Green: collagen scaffold. Blue: cell nuclei.

Figures Figure 3.4.3 and Figure 3.4.4 also show that different treatments affect the morphology of the matrix surrounding cells. It has been experimentally demonstrate that the cells utilized in this study (fibroblasts) can apply enough forces to buckle the struts of the scaffold [Freyman et al. 2001; Harley et al. 2007]. The results of the contractile forces applied by cells are evident macroscopically since the diameter of the scaffold decreases significantly within 5 days, particularly to TC known to induce significant MFB differentiation (e.g. wild type cells treated by TGFβ1).

3.4.3 Data Analysis

Raw Data

Figure 3.4.5 shows the sample size N_i (number of cells analyzed) for each TC for the preliminary results presented here. Unfortunately, the sample size of some TC is inappropriately small because samples contained less cells, because some samples were much harder to analyze than others, or because some imaging fields contained fewer cells. As more imaging data are incorporated in the analysis, the sample size N_i is expected to increase. All data presented in this thesis are obtained from 2 to 4 samples.

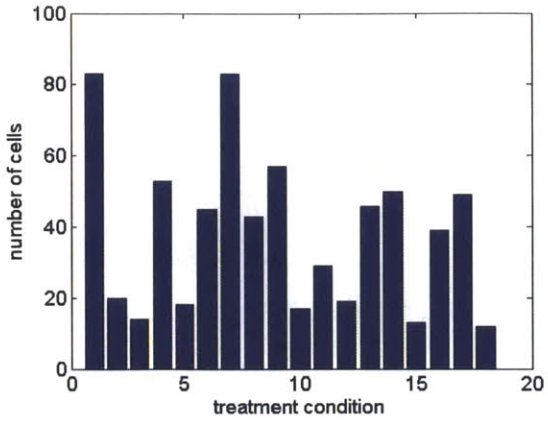


Figure 3.4.5: Sample size (number of cells analyzed) per treatment condition.

Figure 3.4.6, Figure 3.4.7, and Figure 3.4.8 shows the statistical distribution of the $N_F = 45$ features utilized in this study. The features are organized as features that quantify the cell cytoplasm (Figure 3.4.6), features that describe cell nuclei (Figure 3.4.7) and features that describe the micro-environment of cells (Figure 3.4.8). Each figure provides box plots that show the distribution of each feature in the 18 TC, and a histogram that shows the distribution of the feature over the entire cell population of all TC. Each box plot shows the 25th and 75th percentiles F_L , F_U (rectangular box), the median value (horizontal line), therange of data not considered outliers (whiskers), and outliers (shown as +), defined as data outside the range $[F_L - 1.5d_F, F_U + 1.5d_F]$ where $d_F = F_U - F_L$ is the interquantile range [Härdle and Simar 2003]. Data are quite noisy. Some features seems to be indistinguishable in all TC (e.g. the mean cell-tracker intensity per cell CCIA), other features seem to vary more among different TC. The distribution of most features seems reasonably approximated by a normal distribution (although a statistical analysis is necessary to prove this claim), although there are some exceptions (particularly the features that describe cell-cell adhesion in Figure 3.4.8)

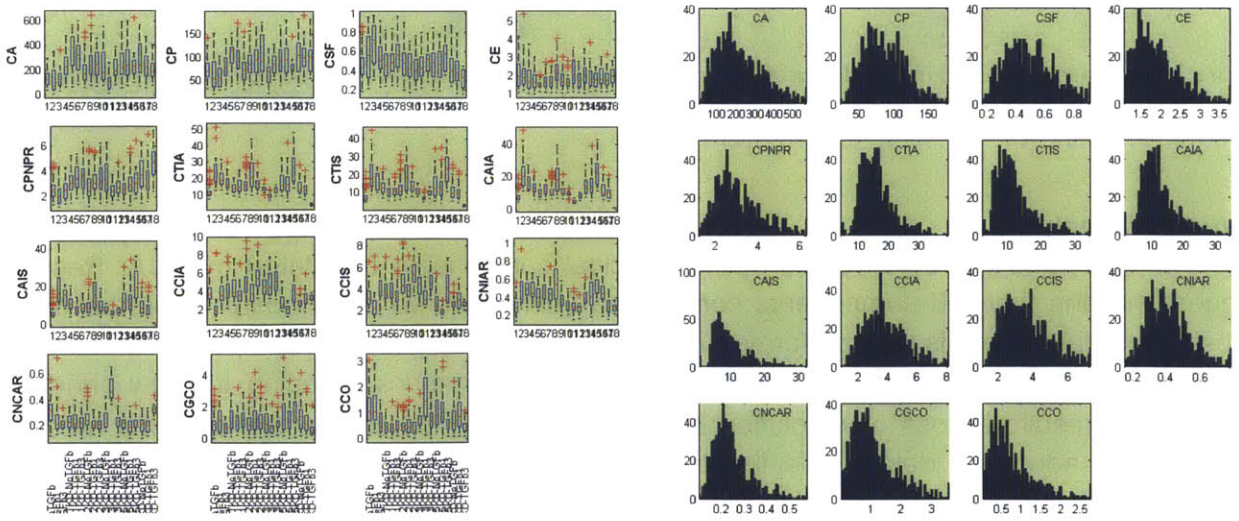


Figure 3.4.6: Statistical distribution of features that quantify cell cytoplasm. Left: Box plots of the features for the 18 TC considered in this study. Right: Histogram of each feature over the entire experiment cell population. The description of the features is available in Table 3.3-2.

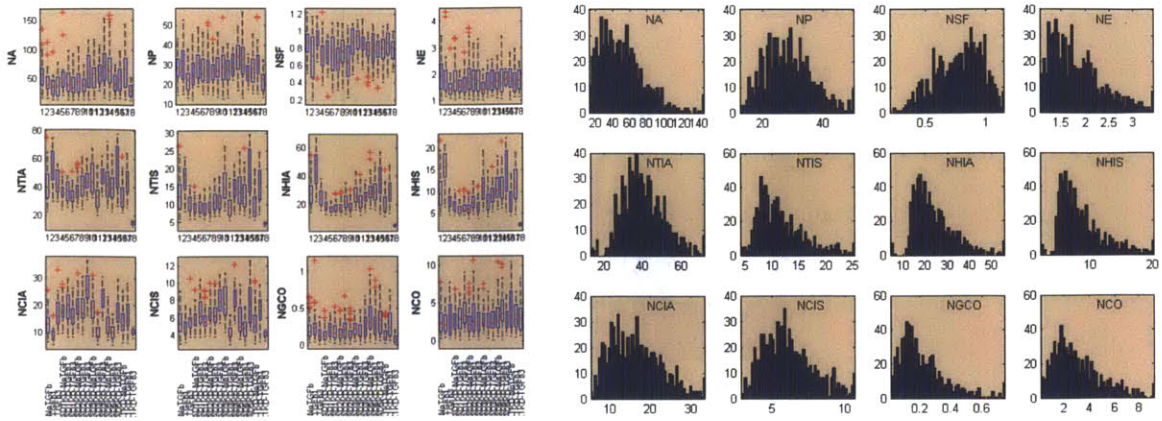


Figure 3.4.7: Statistical distribution of features that quantify cell nuclei. Left: Box plots of the features for the 18 TC considered in this study. Right: Histogram of each feature over the entire experiment cell population. The description of the features is available in Table 3.3-2.

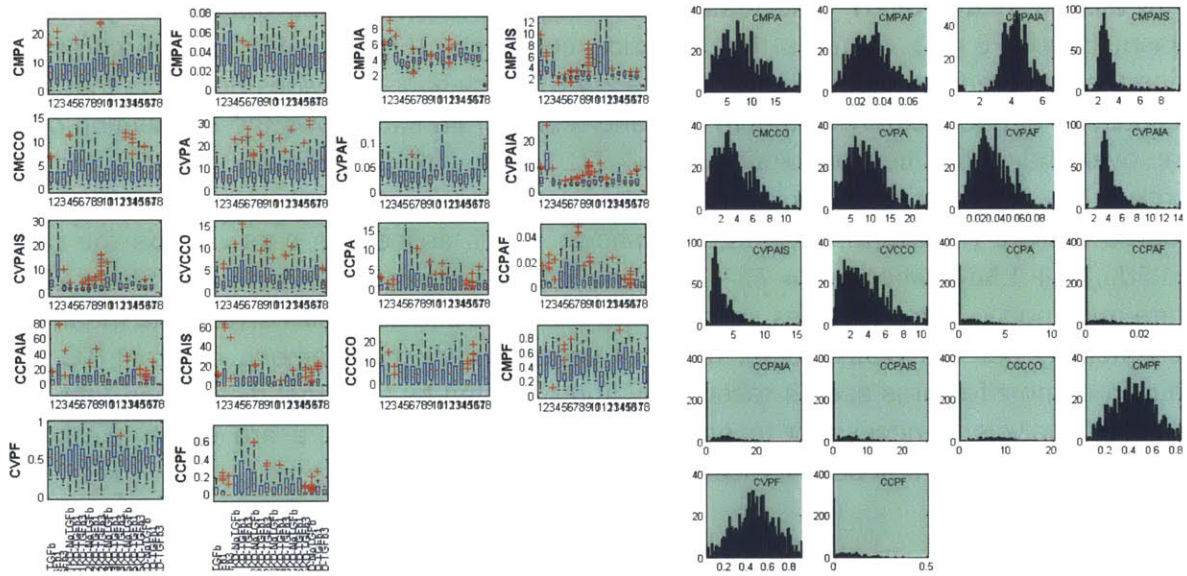


Figure 3.4.8: Statistical distribution of features that quantify cell microenvironment. Left: Box plots of the features for the 18 TC considered in this study. Right: Histogram of each feature over the entire experiment cell population.. The description of the features is available in Table 3.3-2.

Statistical Analysis

Figure 3.4.9 shows the eigenvalues of the principal components calculated by PCA. Results suggest that the first 5-6 components contain the bulk of the variance observed in the data. Subsequent data analysis was performed using the first $N_R = 5$ principal components, because this choice maximized the magnitude of the $D_{i,j}$ distance metrics. Keeping too few principal components results in loss of resolution between different TC because the systems are not described in adequate detail. Keeping too many principal components results in loss of resolution probably due to the effect of random variations.

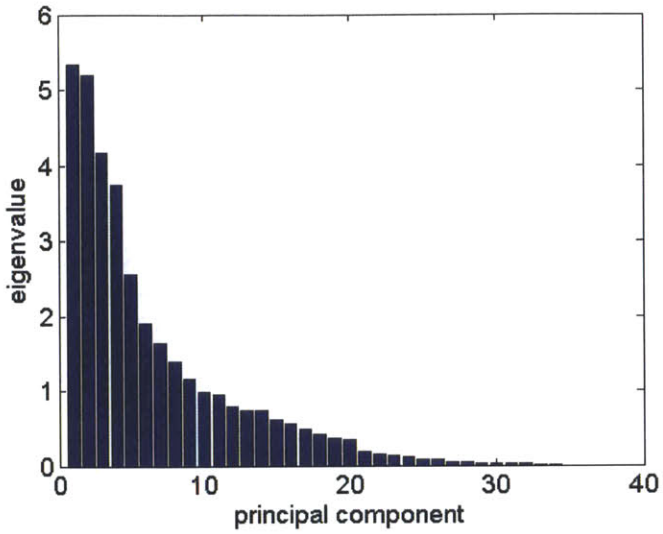


Figure 3.4.9: Eigenvalues of the principal components.

Figure 3.4.10 shows scatter plots for the first 3 reduced normalized features of the 18 treatment conditions. Each point corresponds to a cell of the corresponding TC. Each figure shows results for a particular kind of cells (WT or shRNA treated). Each color corresponds to a particular stimulus (No TGFβ, TGFβ1, TGFβ3). In some cases the segregation of points corresponding to cells of different TC is visible (e.g. SMAD3$^-$), in others not.

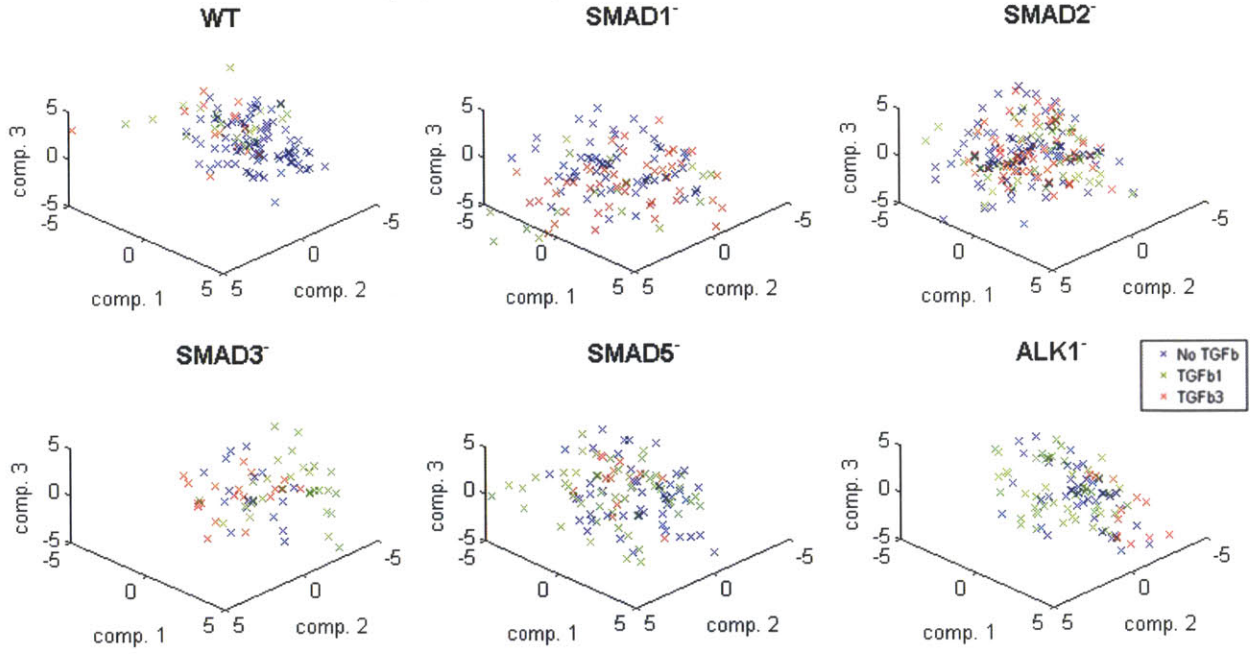


Figure 3.4.10: Scatter plots of the 3 reduced normalized features after PCA for the 18 TC considered in this study. Each figure corresponds to a particular kind of cell. Each color to a particular kind of stimulus.

Figure 3.4.11 shows the results of hierarchical clustering the response of the six treatment conditions (TC) that correspond to no TGFβ stimulation (Figure 3.4.11a),TGFβ1 stimulation (Figure 3.4.11b), and TGFβ3 stimulation (Figure 3.4.11c).The colormap shown in Figure 3.4.11c

assigns blue color to "similar" TC and red color to "different" TC, as quantified by the distance metric $D_{i,j}$. Results suggest that in the absence of TGFβ stimulation, the responses of the six kinds of cells considered is quite similar. The response of WT cells to TGFβ1 is similar to the responses of SMAD5$^-$, less similar to the response of ALK1$^-$ and SMAD2$^-$ cells, and much less similar to the response of SMAD1$^-$ and SMAD3$^-$ cells to TGFβ1. The response of WT cells to TGFβ3 is quite similar to the responses of SMAD1$^-$, SMAD5$^-$ and SMAD2$^-$, and much less similar to the response of SMAD3$^-$ and ALK1$^-$ cells to TGFβ3. Results of Figure 3.4.11 suggest that both SMAD2 and SMAD3 knock-down affect the response of WT cells to both TGFβ1 and TGFβ3, SMAD1 knock-down affects the response of WT cells to TGFβ1 much more than TGFβ3, ALK1 knock-down affects the response of WT cells to TGFβ3 much more than TGFβ1, and SMAD5 knock-down has minor effects to the response of WT cells to TGFβ isoforms.

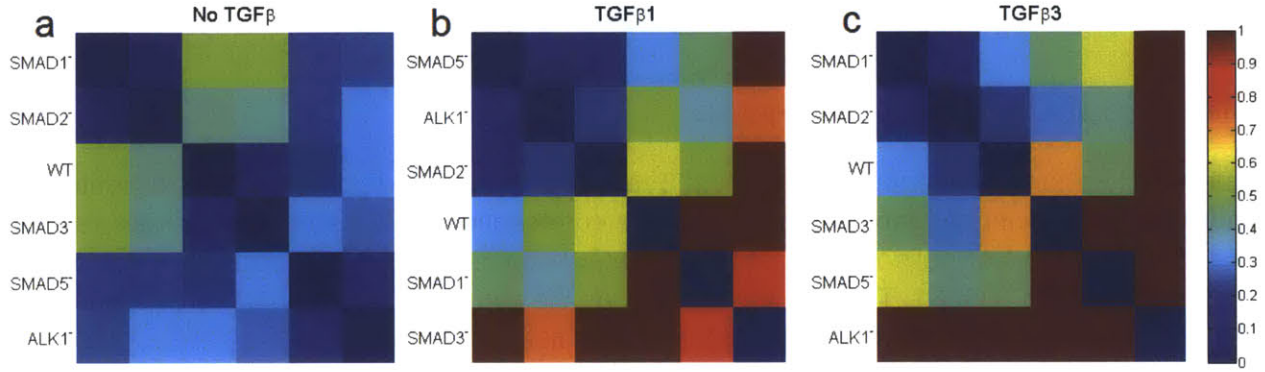


Figure 3.4.11: Results of hierarchical clustering the 6 TC that correspond to no TGFβ stimulation (a), TGFβ1 stimulation (b), and TGFβ3 stimulation (c) based on the Mahalanobis distance metric $D_{i,j}$.

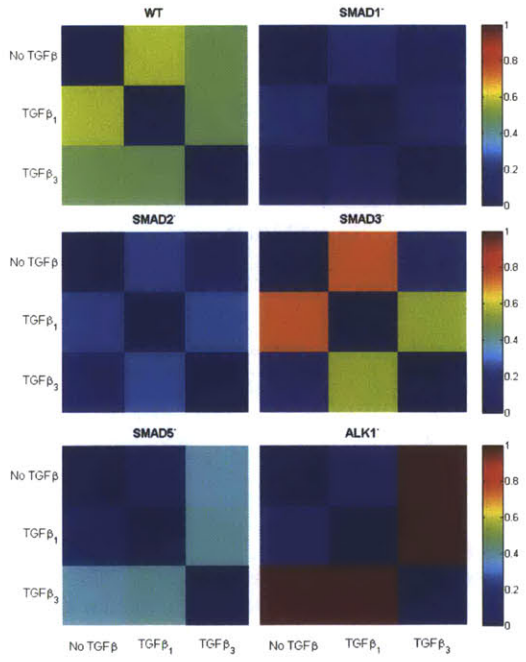


Figure 3.4.12: 3×3 cell response matrices that contain the Mahalanobis distances $D_{i,j}$ between the 3 TC that correspond to the three stimuli (No TGFβ, TGFβ1, TGFβ3) for each cell kind.

Figure 3.4.12 shows the 3×3 cell response matrices $\mathcal{D}_c$ (distances $D_{i,j}$ between the 3 TC that describe the cell response to the three stimuli) for six kinds of cells. Each $\mathcal{D}_c$ describes how the response of each cell type (WT or X^-) differs as a function of the applied stimulus. The results of Figure 3.4.12 suggest that WT, $SMAD5^-$ and $ALK1^-$ cells respond to TGFβ1 and TGFβ3 somewhat different compared to the No TGFβ case. $SMAD1^-$, and $SMAD2^-$ seem to have indistinguishable responses to all three stimuli. $SMAD3^-$ cells respond to TGFβ1 (but no TGFβ3) different than No TGFβ. $SMAD5^-$ cells respond to TGFβ3 (but no TGFβ1) slightly different than No TGFβ. Finally $ALK1^-$ cells response to TGFβ3 (but no TGFβ1) much more differently than No TGFβ.

The calculations shown in Figure 3.4.11 and Figure 3.4.12 were based on a set of features that consists of features that quantify the cell cytoplasm, features that quantify the cell nucleus, and features that describe the interaction of the cell with its micro-environment (Table 3.3-2). Figure 3.4.13 shows how the response matrices $\mathcal{D}_c$ depend on the sub-set of features utilized in the calculations of the Mahalanobis distance. They show that the three groups of features provide similar results.

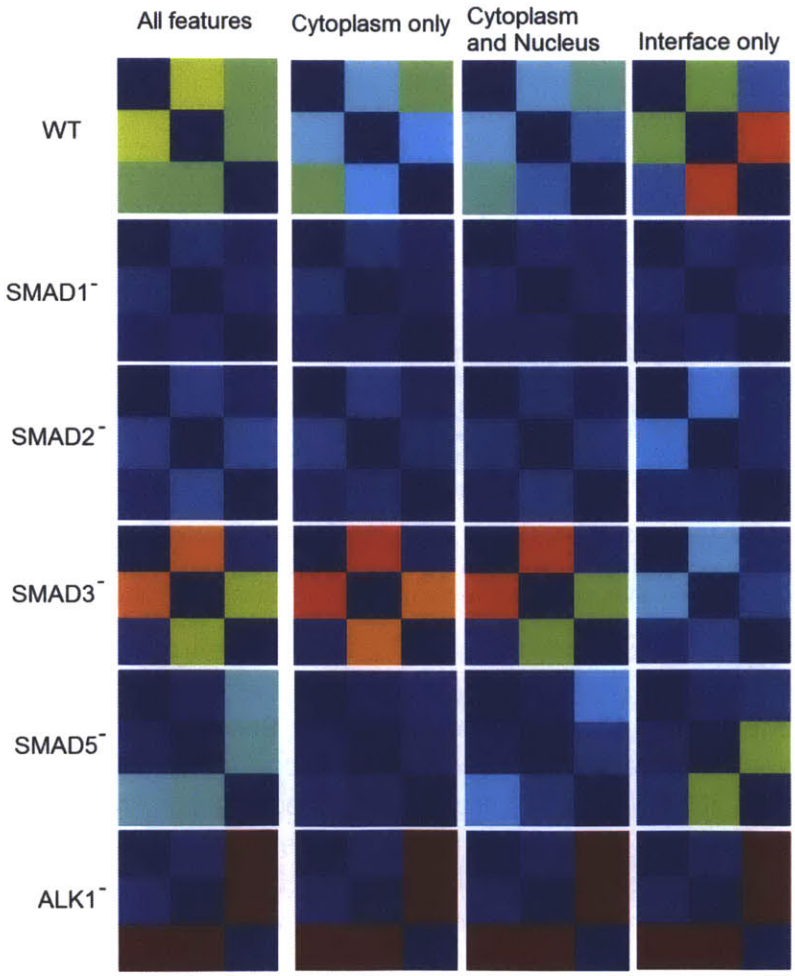


Figure 3.4.13: Comparison of the response matrices shown in Figure 3.4.12 as a function of the sub-set of features utilized in the calculations of the Mahalanobis distance.

3.4.4 Immuno-blotting results

Figure 3.4.14 shows immuno-blotting results of αSMA expression in lysates from the 21 TC. This assay includes ALK5$^-$ cells, which have not been included yet in the imaging assay.

Results show that in most cell kinds both TGFβ1 and TGFB3 stimuli increase αSMA expression compared to no TGFβ stimulus. Knocking down any one of the SMADs and ALKs considered in this study seems to have no effect on αSMA expression in the absence of TGFβ stimulation.

Knocking down the components of canonical TGFβ signaling pathway (SMAD2,3 and ALK1,5) has variable effects. Knocking down ALK5 and SMAD3 increases αSMA expression induced by TGFβ1 and TGFβ3 stimulation. Knocking down SMAD2 increases αSMA expression induced by TGFβ1 but not by TGFβ3. Knocking down ALK1 seems not to affect αSMA expression induced by TGFβ1 and TGFβ3.

Knocking down components of non-canonical TGFβ signaling part has different results. Knocking down SMAD1 expression reduces αSMA expression response to TGFβ1 and TGFβ3. Knocking down SMAD5 seems not to affect the αSMA expression response to both TGFβ1 and TGFβ3.

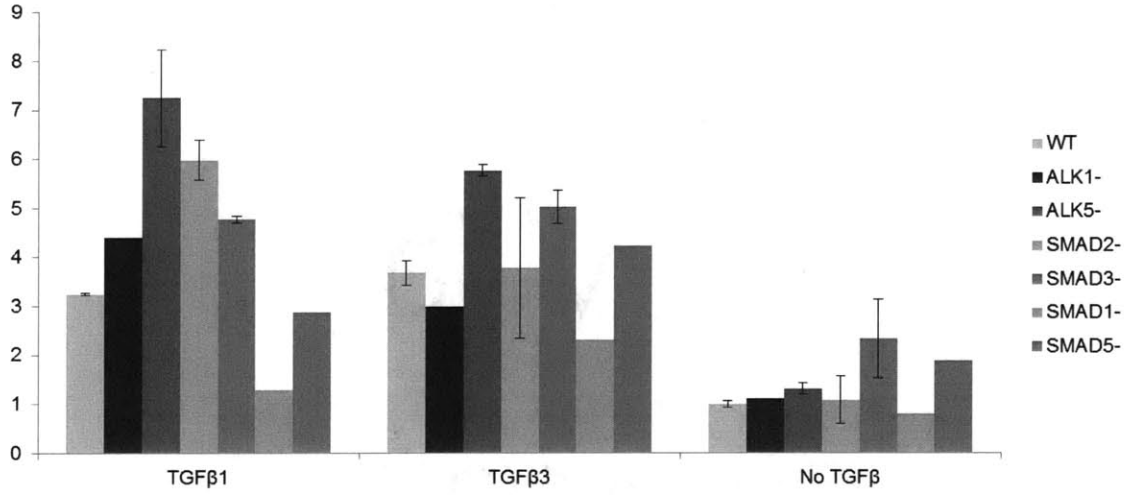


Figure 3.4.14: Quantification of αSMA expression in the 21 TC considered in this study by immune-blotting (the 18 TC considered shown in imaging study plus the three TC for the ALK5$^-$ cells). The expression of αSMA is quantified by densitometry and is normalized based on the expression of αtubulin at the same samples.

3.5 Discussion

3.5.1 Image Informatics in 3 Dimensional Cell-Matrix Samples

This Chapter describes an image informatics methodology for studying signal transduction in cells that interact with a 3D matrix. In the proposed methodology, shRNA is used to stably knock down one component of the pathway of interest in cells, which are then seeded inside a biologically-relevant 3D matrix. The response of cells inside the 3D to particular stimuli is quantified by spectral multi-photon microscopy and subsequent imaging using the methods described in Chapter 2, which enable high-content 3D imaging of complex cell-matrix systems. Each treatment condition (cells treated with a particular shRNA vector and stimulated by a particular input) is quantified by describing each cell in its images using a vector of features. The

role of each pathway component is inferred by quantifying the effect of its knock-down to the response of the cell-scaffold system.

This is the first time that image informatics have been attempted in the study of cell signaling in cells that interact with a 3D matrix. Previous implementations used cells cultured on flat plastic surfaces. The inclusion of the surrounding matrix can provide more physiological relevant results and enable the application of image informatics for *in vitro* imaging-based screening related to biological processes, where the matrix is a key regulator of cell phenotypes. Examples of such case include the interaction of cell with biomaterials and the interaction of cancer cells with their surrounding matrix. Furthermore, in the approach described here, the cell-matrix system is quantified by quantifying both cell morphology as well as the immediate microenvironment in the proximity of cells (matrix, cells, void), in contrast with the standard image informatics approach that utilized only features that describe cell morphology.

Implementing image informatics in 3D cell-matrix systems is much more experimentally challenging compared to the standard implementation on flat plastic surfaces. This work utilized spectral multi-photon imaging for two reasons. First, multi-photon imaging enables 3D imaging of the cell-scaffold system, since standard fluorescence microscopy (used in previous image informatics studies) is not able to quantify 3D structures. Second, spectral detection combined with the image analysis pipeline described in Chapter 2 enables high-content imaging, which reduces the number of required experiments and enables quantifying the cell-scaffold system using a larger number of assays.

3.5.2 Probing the Role of SMAD Pathway Components in TGFβ Isoform Signaling

The developed methodology is applied in a pilot study of TGFβ signaling in fibroblasts seeded inside porous collagen matrixes via the SMAD signal transduction pathway. The objective of the study is to collect evidence that can explain whether and how the two major TGFβ isoforms (TGFβ1 and TGFβ3) stimulate cells in different ways via the SMAD pathway. The two TGFβ isoforms are structurally similar and their effects are known to be mediated by the SMAD pathway. However, it is not known if different SMAD components are responsible for their different effects to cells.

This pilot study is of significant biological relevance. TGFβ isoforms are known to be involved in several important biological processes including wound healing. TGFβ1 and TGFβ3 are associated with very different outcomes in wound healing. TGFβ1 is associated with wounds that heal by wound contraction and scar formation (repair). TGFβ3 is experimentally shown to reduce scar formation and wound contraction, therefore favoring regeneration (see Chapter 4). The study is implemented in fibroblasts seeded inside porous collagen-GAG scaffolds. Fibroblasts can differentiate into myofibroblasts, a key process that is believed to affect the wound healing outcome significantly. The porous collagen-GAG scaffolds used in this study are very similar to the ones utilized clinically to induce regeneration in skin and nerve nerves [Yannas et al. 1982, 1989; Chamberlain et al. 1998]. Such collagen scaffolds provide a 3D microenvironment where cells adhere extensively both *in vitro* (Figure 3.4.2 and Figure 3.4.3) and *in vivo* (Chapter 4).

The main results of this pilot study (Figure 3.4.11 and Figure 3.4.12) show that almost all components of the SMAD pathway have some involvement in TGFβ signaling. So far it is known that some of the SMAD pathway components are involved in TGFβ signaling (SMAD2,3) and

some components are involved in BMP signaling (SMAD1,5,8). The results of this pilot study show that indeed knocking down both "canonical" components (SMADS2,3) affects the response of WT cells to both TGFβ isoforms, in agreement with the known critical role of these SMADS in TGFβ signaling. Results show that knocking down SMAD1 affects the response of WT cells to TGFβ1 more than their response to TGFβ3. On the contrary, knocking down the TGFBR1 receptor ALK and to less extent SMAD5 affect the response of WT cells to TGFβ3 more than the response to TGFβ1 (Figure 3.4.11).

While the response of WT cells to TGFβ1 and TGFβ3 is slightly different than the unstimulated response, the response of SMAD1$^-$ and SMAD2$^-$ cells to TGFβ1 or TGFβ3 is similar to the unstimulated case (Figure 3.4.12). However, the unstimulated responses of SMAD1 and SMAD2 are similar and slightly different than the unstimulated response of WT cells. SMAD3$^-$ cells respond more[6] to TGFβ1 and less to TGFβ3 compared to WT cells (Figure 3.4.12). The response of SMAD5$^-$ and ALK1$^-$ cells to TGFβ1 is similar to the unstimulated case. SMAD5$^-$ cells respond less to TGFβ3 compared to WT cells, while ALK1$^-$ cells respond much more to TGFβ3 compared to WT cells.

A comparison of the results shown in Figure 3.4.11 and Figure 3.4.12 highlight the multi-dimensional nature of the problem. The distance metric $D_{i,j}$ shown in Figure 3.4.11 and Figure 3.4.12 corresponds a distance between points in a N_R-dimension coordinate system, where each point corresponds to a particular TC.

Although the analysis presented here is preliminary, it suggests that apart from the two "canonical" SMAD2,3, the non-canonical SMAD1,5 affect TGFβ signaling. The involvement of non-canonical SMAD1,5 and TGFBR receptors (ALK1) could cause TGFβ1 and TGFβ3 to induce different effects to cells. SMAD1 seems to affect mostly TGFβ1 signaling, while SMAD5 and ALK1 seem to affect mostly TGFβ3 signaling. These conclusions are subject to more detailed statistical analysis. A letter of warning is that image informatics cannot provide direct evidence about a molecular interaction. Instead they provide evidence that can guide future biochemical/genetic experiments that can give definitive proof of these claims.

3.6 Summary and Future Work

This chapter describes an image informatics methodology that can be used to study signal transduction pathways inside 3D matrices. The methodology is applied in a pilot study of TGFβ isoform signaling in primary fibroblasts inside porous collagen scaffolds. The results of the pilot study suggest that non-canonical SMADS (SMAD1,5) and receptor ALK1 could be involved in a mechanism that causes cells to respond differently to TGFβ1 and TGFβ3 signaling.

The presented data are not complete and conclusive. Specifically, there is need to increase the sample size in several of the TC presented here (see Figure 3.4.5) and to include the imaging results of the ALK5 knock-down cells. This will be done by including several imaging experiments conducted recently. There is a need for a more detailed statistical analysis that can provide statistical evidence on the significance of the results presented in this Chapter. Of particular interest is to quantify the statistical significance of the Mahalanobis distances $D_{i,j}$

[6] Here, the term "more" response to some TGFβ isoform means that the response of cells to that TGFβ isoform is more drastic compared to the unstimulated response (No TGFβ).

shown in Figure 3.4.11 and Figure 3.4.12. It is also possible to increase the number of features calculated per cell and evaluate how the number and identity of features used affect the result.

This study emphasizes the role of the insoluble matrix as a regulator of signal transduction pathways in cells interacting with 3D matrices. The response of each cell is quantified not only based on its morphology, but also based on its binding pattern with the surrounding matrix. Here, the binding pattern was quantified based on the proximity between the cell and the surrounding matrix. More accurate ways to evaluate cell interactions with the matrix should focus on labeling particular components of the cell adhesion mechanism or utilize FRET reporters of focal adhesion molecules [Ritt et al. 2013]. Furthermore, additional information could be inferred by the morphology of the matrix itself. For example Figure 3.4.4 shows that certain TC treatments alter scaffold porosity significantly.

The methodology described in this Chapter can be applied to study the effect of matrix properties on regulating signaling induced by soluble regulators such as TGFβ isoforms. Specifically, it is known that signal transduction pathways induced by cell adhesion cross-talk with signaling pathways stimulated by soluble regulators (see [Gomperts et al. 2009] and the Background section of Chapter 5). In this study all experiments took place using collagen-GAG scaffolds of identical properties. However, experiments can take place inside scaffolds of various physic-chemical properties. Such experiments could use scaffolds of varying mean pore size [O'Brien et al. 2004], scaffolds of varying cross-linking density [Soller et al. 2012], or scaffolds of varying surface chemistry (see Chapter 5). The later could be implemented by "doping" the collagen with ECM components, such as the recombinant fibronectin fragments described in Appendix E. Targeting the surface chemistry can modulate the type and amount of integrin ligands available to cells for adhesion, and therefore control the signaling pathways excited by cell adhesion.

Large-scale experiments in 3D matrices require high throughput due to the large size and the complexity of the samples. The implementation of the image informatics described here is based on a spectral multi-photon microscope which provides high-content images, however its frame throughput is moderate. Large-scale experiments would benefit great by automatic sample handling, microfluidic devices for handling solutions into the 3D cell-scaffold samples, and high-throughput high-content 3D imaging instruments, such as the multi-focal multi-photon microscopes (MMM) with spectral-lifetime imaging capability under development in the So Lab [Bird et al. 2004; Kim et al. 2007].

3.7 Acknowledgements

This study would not have been possible without Chengpin Chen and Jin Guo, who contributed in the biochemical methodology development, and collected the data where this study is based upon.

3.8 Literature Cited

Adiga U, Malladi R, Fernandez-Gonzalez R, and de Solorzano CO. High-throughput analysis of multispectral images of breast cancer tissue. *IEEE Trans. Image Processing* **15**(8): 2259-68, 2006. (doi: 10.1109/TIP.2006.875205).

Al-Kofahi Y, Lassoued W, Lee W, and Roysam B. Improved Automatic Detection and Segmentation of Cell Nuclei in Histopathology Images, *IEEE Trans. Biomed. Eng.* **57**(4): 841-52, 2010.

Allen M, and Jones JL. Jekyll and Hyde: the role of the microenvironment on the progression of cancer, *J. Pathol.* **223**(2): 162-76, 2011.

Ask K, Bonniaud P, Maass K, Eickelberg O, Margetts PJ, Warburton D, Groffen J, Gauldie J, and Kolb M. Progressive pulmonary fibrosis is mediated by TGF-beta isoform 1 but not TGF-beta3. *Int J Biochem Cell Biol.* **40**(3):484-95, 2008.

Bakal C, Aach J, Church G, and Perrimon N., Quantitative Morphological Signatures Define Local Signaling Networks Regulating Cell Morphology. *Science* **316**(5832): 1753–6, 2007.

Bakal C, Linding R, Llense F, Heffern E, Martin-Blanco E, Pawson T, and Perrimon N. Phosphorylation networks regulating JNK activity in diverse genetic backgrounds. *Science* **322**(5900): 453-6, 2008.

Bandyopadhyay B, Fan J, Guan S, Li Y, Chen M, Woodley DT, and Li W. A "traffic control" role for TGFbeta3: orchestrating dermal and epidermal cell motility during wound healing. *J Cell Biol.* **172**(7): 1093-105, 2006.

Berg RH, Evaluation of spectral imaging for plant cell analysis, *J. Microscopy* **214**(2): 174-81, 2004.

Bestvater F, Spiess E, Stobrawa G, Hacker M, Feurer T, Porwol T, Berchner-Pfannschmidt U, Wotzlaw C, and Acker H, Two-photon fluorescence absorption and emission spectra of dyes relevant for cell imaging, *J. Microsc.* **208**(2): 108-15, 2002.

Bird DK, Eliceiri KW, Fan C-H, and White JG. Simultaneous Two-Photon Spectral and Lifetime Fluorescence Microscopy. *Applied Optics* **43**(27): 5173-82, 2004.

Bjornsson CS, Lin G, Al-Kofahi Y, Narayanaswamy A, Smith KL, Shain W, and Roysam B. Associative image analysis: a method for automated quantification of 3D multi-parameter images of brain tissue. *J Neurosci Methods* **170**(1): 165-78, 2008. (doi: 10.1016/ j.jneumeth.2007.12.024).

Candy JV. *Bayesian Signal Processing. Classical, Modern and Particle Filtering Methods*, Wiley, 2009.

Chamberlain LJ, Yannas IV, Hsu HP, Strichartz G, and Spector M. Collagen-GAG substrate enhances the quality of nerve regeneration through collagen tubes up to level of autograft. *Exp. Neurol.* **154**(2): 315-29, 1998.

Chamberlain LJ, Yannas IV, Arrizabalaga A, Hsu HP, Norregaard TV, and Spector M. Early peripheral nerve healing in collagen and silicone tube implants: myofibroblasts and the cellular response. *Biomaterials* **19**(15): 1393-403, 1998b.

Chen C-T, *Linear System Theory and Design*, Third Edition, Oxford University Press USA, 1998

Chen S-C, Zhao T, Gordon GJ, and Murphy RF. A novel graphical model approach to

segmenting cell images. *2005 IEEE Symp. Comput. Intell. Bioinf. Comput. Biol.*: 1079, 2006.

Cheng J, and Rajapakse JC. Segmentation of Clustered Nuclei With Shape Markers and Marking Function, *IEEE Trans. Biomedical Engineering* **56**(3): 741-8, 2009.

Coelho LP, Shariff A, and Murphy RF. Nuclear segmentation in microscope cell images: A hand-segmented dataset and comparison of algorithms, *ISBI 2009 IEEE Int. Symp. Biomed. Imag*: 518-21, 2009.

Coulot L, Kirschner H, Chebira A, Moura JMF, Kovacevic J, Osuna EG, and Murphy RF. Topology preserving STACS segmentation of protein subcellular location images. *2006 IEEE Intl. Symp. Biomed. Imaging*: 566-9, 2006.

Dempster AP, Laird NM, and Rubin DB. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. Royal Statist. Soc. B.* **39**(1): 1-38, 1977.

Deodhar AK, and Rana RE. Surgical physiology of wound healing: a review. *J. Postgrad. Med.* **43**(2): 52-6, 1997.

Derynck R, and Miyazono K. (Editors). The TGF-beta Family. Cold Spring Harbor Laboratory Press, 2007.

Derynck R, and Zhang YE. Smad-dependent and Smad-independent pathways in TGF-beta family signalling. *Nature* **425**(6958): 577-84, 2003.

Desmoulière A, Chaponnier C, Gabbiani G. Tissue repair, contraction, and the myofibroblast. *Wound Repair Regen.* **13**(1): 7-12, 2005.

Dima AA, Elliott JT, Filliben JJ, Halter M, Peskin A, Bernal J, Kociolek M, Brady MC, Tang HC, Plant AL. Comparison of segmentation algorithms for fluorescence microscopy images of cells. *Cytometry A* **79**(7): 545-59, 2011.

Douglas HE. TGF-β in wound healing: a review. *J. Wound Care* **19**(9): 403-6, 2010.

Feng XH, and Derynck R. Specificity and versatility in tgf-beta signaling through Smads. *Annu. Rev. Cell Dev. Biol.* **21**: 659-93, 2005.

Freyman TM, Yannas IV, Yokoo R, and Gibson LJ. Fibroblast contraction of a collagen-GAG matrix. *Biomaterials* **22**(21): 2883-91, 2001.

Gomperts BD, Kramer IM, and Tatham PER. *Signal Transduction*. Academic Press, 2009.

Gough NR, and Yaffe MB. Focus issue: conquering the data mountain, *Sci. Signal.* **4**(160): eg2, 2011.

Goumans MJ, Valdimarsdottir G, Itoh S, Rosendahl A, Sideras P, and ten Dijke P. Balancing the activation state of the endothelium via two distinct TGF-beta type I receptors. *EMBO J.* **21**(7): 1743-53, 2002.

Goumans MJ, Valdimarsdottir G, Itoh S, Lebrin F, Larsson J, Mummery C, Karlsson S, and ten Dijke P. Activin receptor-like kinase (ALK)1 is an antagonistic mediator of lateral TGFbeta/ALK5 signaling. *Mol Cell.* **12**(4): 817-28, 2003.

Goumans MJ, Liu Z, and ten Dijke P. TGF-beta signaling in vascular biology and dysfunction. *Cell Res.* **19**(1): 116-27, 2009.

Green JA, and Yamada KM, Three-dimensional microenvironments modulate fibroblast

signaling responses, *Adv. Drug Deliv. Rev.* **59**(13): 1293-8, 2007.

Griffith LG, and Swartz MA. Capturing complex 3D tissue physiology in vitro. *Nat Rev Mol Cell Biol.* **7**(3): 211-24, 2006.

Grinnell F. Fibroblasts, myofibroblasts, and wound contraction, *J. Cell Biol.* **124**(4): 401-4, 1994.

Grinnell F. Fibroblast biology in three-dimensional collagen matrices, *Trends Cell Biol.* **13**(5): 264-9, 2003.

Gu L, Zhu Y, Yang X, Guo Z, Xu W, and Tian X. Effect of TGF-beta/Smad signaling pathway on lung myofibroblast differentiation, *Acta Pharmacologica Sinica* **28**: 382-91, 2007.

Guo J. *The study of circulating tumor cells and TGF-β signaling in fibroblasts inside porous collagen scaffolds by emerging fluorescent microscopy techniques.* Ph.D. Dissertation, Department of Chemistry, Institute of Biomedical Sciences, Fudan University, China, 2013. (in Chinese).

Härdle W, and Simar L. *Applied Multivariate Statistical Analysis*, Springer, 2003.

Hall M. Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning.: *Proc. 7th Int. Conf. Mach. Learn.* Morgan Kaufmann Publishers Inc: 359-366, 2000.

Hagwood C, Bernal J, Halter M, and Elliott J, Evaluation of Segmentation Algorithms on Cell Populations Using CDF Curves, IEEE Trans. *Medical Imaging* **31**(2): 380-90, 2012.

Harley BA, Freyman TM, Wong MQ, and Gibson LJ. A new technique for calculating individual dermal fibroblast contractile forces generated within collagen-GAG scaffolds. *Biophys J.* **93**(8): 2911-22, 2007.

Hastie T, Tibshirani R, and Friedman J. *The Elements of Statistical Learning; Data Mining, Inference, and Prediction.* Second Edition, Springer, 2009.

Hinz B. Masters and servants of the force: the role of matrix adhesions in myofibroblast force perception and transmission. *Eur J Cell Biol.* **85**(3-4): 175-81, 2006.

Hinz B. Formation and function of the myofibroblast during tissue repair. *J Invest Dermatol.* **127**(3): 526-37, 2007.

Hinz B. Tissue stiffness, latent TGF-beta1 activation, and mechanical signal transduction: implications for the pathogenesis and treatment of fibrosis. *Curr Rheumatol Rep.* **11**(2): 120-6, 2009.

Hinz B, Phan SH, Thannickal VJ, Galli A, Bochaton-Piallat ML, and Gabbiani G. The myofibroblast: one function, multiple origins. *Am J Pathol.* **170**(6): 1807-16, 2007.

Hsu WC, Spilker MH, Yannas IV, and Rubin PAD. Inhibition of conjunctival scarring and contraction by a porous collagen-glycosaminoglycan implant. *Invest. Opthamol. Vis. Sci.* **41**: 2404-11, 2000.

Jain AK, Murty MN, and Flynn PJ. Data clustering: a review. *ACM Computing Surveys* **31**(3): 264-323, 1999. (doi: 10.1145/33199.331504).

Jones TR, Carpenter AE, Golland P, Voronoi-based segmentation of cells on image manifolds, *ICCV Workshop Comput. Vision Biomed. Image Appl.:* 535-43, 2005.

Kim KH, Buehler C, Bahlmann K, Ragan T, Lee WC, Nedivi E, Heffer EL, Fantini S, So PT. Multifocal multiphoton microscopy based on multianode photomultiplier tubes. *Opt*

Express **15**(18): 11658-78, 2007.

Klass BR, Grobbelaar AO, and Rolfe KJ. Transforming growth factor β1 signaling, wound healing and repair: a multifunctional cytokine with clinical implications for wound repair, a delicate balance. *Postgrad. Med. J.* **85**: 9-14, 2009.

Lamprecht MR, Sabatini DM, and Carpenter AE. CellProfiler: free, versatile software for automated biological image analysis. *Biotechniques* **42**(1): 71-5, 2007.

Lee MJ, Ye AS, Gardino AK, Heijink AM, Sorger PK, MacBeath G, Yaffe MB. Sequential application of anticancer drugs enhances cell death by rewiring apoptotic signaling networks. *Cell* **149**(4):780-94, 2012. doi: 10.1016/j.cell.2012.03.031.

Lin G, Adiga U, Olson K, Guzowski JF, Barnes CA, and Roysam B. A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. *Cytometry A* **56**(1): 23-36, 2003.

Madri JA, Pratt BM, and Tucker AM. Phenotypic modulation of endothelial cells by transforming growth factor-beta depends upon the composition and organization of the extracellular matrix. *J. Cell Biol.* **106**(4): 1375-84, 1988.

Massagué J. The transforming growth factor-beta family, *Annu. Rev. Cell Biol.* **6**: 597-641, 1990.

Massagué J. TGF-beta signal transduction, *Annu. Rev. Biochem.* **67**, 753-91, 1998.

Massagué J. How cells read TGF-beta signals. *Nat Rev Mol Cell Biol.* **1**(3): 169-78, 2000.

Massagué J, and Gomis RR. The logic of TGFβ signaling, *FEBS Lett.* **580**(12): 2811-20, 2006.

Masuzaki R, Zhao SR, Csizmadia E, Yannas I, and Karp SJ. Scar formation and lack of regeneration in adult and neonatal liver after stromal injury. *Wound Repair Regen.* **21**(1): 122-30, 2013. (doi: 10.1111/j.1524-475X.2012.00868.x).

Murphy GF, Orgill DP, and Yannas IV. Partial dermal regeneration is induced by biodegradable collagen-glycosaminoglycan grafts. *Lab Invest.* **62**(3): 305-13, 1990.

Nir O, Bakal C, Perrimon N, and Berger B. Inference of RhoGAP/GTPase regulation using single-cell morphological data from a combinatorial RNAi screen, *Genome Res.* **20**(3): 372-80, 2010.

O'Brien FJ, Harley B A, Yannas IV, and Gibson LJ. Influence of freezing rate on pore structure in freeze-dried collagen-GAG scaffolds, *Biomaterials* **256**: 1077-86, 2004. (doi: 10.1016/S0142-9612(03)00630-6)

Perlman ZE, Slack MD, Feng Y, Mitchison TJ, Wu LF, and Altschuler SJ, Multidimensional drug profiling by automated microscopy. *Science* **306**(5699): 1194-8, 2004.

Ritt M, Guan JL, and Sivaramakrishnan S. Visualizing and manipulating focal adhesion kinase regulation in live cells. *J Biol Chem.* **288**(13): 8875-86, 2013. (doi: 10.1074/ jbc.M112.421164).

Rosado-Toro JA, and Rodriguez JJ. Cell splitting using dynamic programming, *IEEE Southwest Symp. Image Anal. Interp. (SSIAI)*: 33-6, 2012.

Scott JD, and Pawson T. Cell signaling in space and time: where proteins come together and when they're apart, *Science* **326**(5957): 1220-4, 2009.

Shah M, Foreman DM, and Ferguson MW. Neutralisation of TGF-beta 1 and TGF-beta 2 or exogenous addition of TGF-beta 3 to cutaneous rat wounds reduces scarring. *J Cell Sci.* **108**(3): 985-1002, 1995.

Soille P. *Morphological Image Analysis: Principles and Applications*, Second Edition, Springer, 2003.

Soller EC, Tzeranis DS, Miu K, So PT, and Yannas IV. Common features of optimal collagen scaffolds that disrupt wound contraction and enhance regeneration both in peripheral nerves and in skin. *Biomaterials* **33**(19): 4783-91 2012. (doi:10.1016/j.biomaterials. 2012.03.068).

Stegemann JP, Hong H, and Nerem RM. Mechanical, biochemical, and extracellular matrix effects on vascular smooth muscle cell phenotype. *J Appl Physiol.* **98**(6): 2321-7, 2005.

Surut Y, and Phukpattaranont P. Preliminary Results of Connected Object Segmentation Using Gradient Direction, *2nd Int. Conf. Infor. Eng. Comp. Sci. (ICIECS)*: 1-4, 2010.

Tan CS, and Linding R. Experimental and computational tools useful for (re)construction of dynamic kinase-substrate networks. *Proteomics* **9**(23): 5233-42, 2009. (doi: 10.1002/ pmic.200900266).

Troxel KS. *Delay of skin wound contraction by porous collagen-GAG matrices*, PhD Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 1994.

Usuki J, Matsuda K, Azuma A, Kudoh S, and Gemma A. Sequential analysis of myofibroblast differentiation and transforming growth factor-β1/Smad pathway activation in murine pulmonary fibrosis, *Journal of Nippon Medical School* **79**: 46-59, 2012.

Vokes MS, and Carpenter AE. Using CellProfiler for automatic identification and measurement of biological objects in images. *Curr Protoc Mol Biol.* Chapter 14: Unit 14.17, 2008. (doi: 10.1002/0471142727.mb1417s82).

Waddington SN, Crossley R, Sheard V, Howe SJ, Buckley SM, Coughlan L, Gilham DE, Hawkins RE, and McKay TR. Gene delivery of a mutant TGFβ3 reduces markers of scar tissue formation after cutaneous wounding. *Mol Ther.* **18**(12): 2104-11, 2010. (doi: 10.1038/mt.2010.174).

Wählby C, Sintorn IM, Erlandsson F, Borgefors G, and Bengtsson E. Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. *J Microsc.* **215**(1): 67-76, 2004.

Wollman R, and Stuurman N. High throughput microscopy: from raw images to discoveries. *J Cell Sci.* **120**(21): 3715-22, 2007.

Xiong G, Zhou X, and Ji L. Automated Segmentation of Drosophila RNAi Fluorescence Cellular Images Using Deformable Models, *IEEE Trans. Circ. Syst.* **53**(11): 2415-24, 2006.

Yannas IV. *Tissue and organ regeneration in adults*, Springer, New York, 2001.

Yaffe MB. Signaling networks and mathematics, *Sci. Signal.* **1**(43): eg7, 2008.

Yannas IV, Burke JF, Orgill DP, and Skrabut EM. Wound tissue can utilize a polymeric template to synthesize a functional extension of skin, *Science* **215**(4529): 174-6, 1982.

Yannas IV, Lee E, Orgill DP, Skrabut EM, and Murphy GF. Synthesis and characterization of a model extracellular matrix that induces partial regeneration of adult mammalian skin, *Proc.*

Natl. Acad. Sci. USA **86**(3): 933-7, 1989.

Zhang G, and Neubert TA. Use of stable isotope labeling by amino acids in cell culture (SILAC) for phosphotyrosine protein identification and quantitation, Methods Mol. Biol. **527**: 79-92, 2009.

Zhang S, Binari R, Zhou R, and Perrimon N. A genomewide RNA interference screen for modifiers of aggregates formation by mutant Huntingtin in Drosophila, *Genetics* **184**(4): 1165-79, 2010.

Zimmer C, Labruyere E, Meas-Yedid V, Guillen N, and Olivo-Marin J-C. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing. *IEEE Trans. Med. Imaging* **21**(10): 1212–21, 2002. (doi: 10.1109/TMI.2002.806292).

Chapter 4: Imaging Peripheral Nerve Regeneration Induced by Collagen Scaffolds

4.1 Chapter Overview

This chapter describes an *ex vivo* imaging study in an established animal model of peripheral nerve (PN) regeneration. This study follows up a recent study that demonstrated a strong inverse relationship between wound contraction and quality of PN regeneration 9 weeks post-injury, and focuses on the early response (1 and 2 weeks post-injury) of PN wound healing. Transected nerves are treated with two kinds of tubular collagen scaffolds known to cause significantly different wound healing outcomes 9 weeks post-injury. The objective of the study is to trace back the performance of these materials during the early events of wound healing, and provide evidence about how these scaffolds regulate critical early processes of PN wound healing that affect its final outcome. Of particular interest is to quantify the formation of the capsule of contractile cells that forms around injured PN during these early time points. The available evidence supports the hypothesis that PN wound healing outcome depends on the "race" between the formation of new nerve tissue and the formation of a contractile cell capsule at the wound edges that impedes regeneration.

4.1.1 Motivation

Severe injury in adult organs is generally irreversible. The spontaneous wound healing process triggered by severe organ injury does not result in the formation of new organ tissue of normal morphology and function [Yannas 2001]. Current options to replace injured organs (transplantation and auto-grafts) suffer from limited availability and side effects. There is urgent need to develop treatments for inducing regeneration in injured organs. Despite numerous research efforts, at the moment there are few clinically applied procedures that can change the wound healing process in ways that induce regeneration (formation of organ tissue of normal morphology and function).

Although it is known that certain biomaterials can induce regeneration, understanding how these materials work in multiple levels (tissue-level, cell-level, molecular-level) is still an area of active research. Such understanding would very valuable for the systematic development of novel biomaterial-based treatments that could induce improved regeneration in a wider range of injured organs.

4.1.2 Chapter Description

This Chapter is a study of PN regeneration induced by biomaterials in the organ-scale and cell-scale. Section 4.2 provides extensive background on wound healing and induced regeneration in peripheral nerves, Section 4.3 describes the animal experiment and the imaging methods used to analyze the *ex vivo* nerve samples, Section 4.4 provides the results of the imaging study, Section 4.5 discusses the findings, and Section 4.6 suggests directions for future research.

4.1.3 Contributions to this study

This study follows-up the peripheral nerve regeneration study conducted by Dr. Eric Soller for his PhD thesis. Specifically, Dr. Soller designed the animal study, fabricated the scaffold tubes, supervised the animal experiments, sacrificed the animals, prepared the harvested nerves (fixation, sectioning), and contributed in result interpretation. Dr. HP Xu (VA Hospital, Jamaica Plains, MA) performed the animal surgeries. While Dr. Soller harvested PN samples 1, 2 or 9 weeks post-transection, his PhD worked focused on the 9-week response. This study focuses on imaging and quantifying the nerve samples collected by Dr. Soller 1 and 2 weeks post-injury. I acknowledge the valuable help of Melissa Buydash in staining nerve samples, and αSMA immunofluorescence imaging and IF image analysis. The author (DT) contributed in section preparation, protocol development, imaging, image processing, and statistical analysis.

4.2 Background

4.2.1 Wound Healing in Adults

Repair and Regeneration

Organ injury initiates a complex biological process called wound healing. The outcome of wound healing is classified as "regeneration" when the resulting tissues match the structure and function of the initial organ. The outcome is classified as "repair" when the resulting tissues do not match the structure of the initial organ, and therefore fail to function normally. The outcome is called "spontaneous" when wound healing takes place in the absence of any specific treatment (e.g. grafts, drugs).

The outcome of spontaneous wound healing depends on several parameters. The most important are the nature of tissues damaged, the size of the injury, and the age of the patient. In adults, the outcome of spontaneous wound healing is repair [Yannas 2001]: at the site of injury site a scar (collagenous-rich abnormal ECM) forms and the dimensions of the organ alter. "Severe" injury depends on the nature of tissues injured and the size of the defect. Several cell-rich tissues can regenerate spontaneously (e.g. epidermis, axons after crush). Injuries involving only such tissues are "reversible". However, when injury involved vascularized ECM-rich tissues (e.g. dermis) the response changes dramatically and apart from very small defects, the outcome of wound healing is repair. Such injuries are therefore called "irreversible".

The response to wound healing varies among different species [Poss 2010]. Some species have the ability to regenerate throughout their adult life. Others (including humans) respond to injury very differently during early gestation compared to late gestation and adulthood [Gurtner et al. 2008]. These observations provide hope that humans possess an intrinsic regenerative capability that if properly "guided" can led to inducing regeneration in severe injuries in adults.

Current Approaches for Induced Regeneration

At the moment, there are three major approaches to replace injured organs: organ transplantation, auto-grafting, and *in vivo* induced regeneration. The first two are widely implemented in clinical practice, however they suffer from limited availability and complications. Induced regeneration has at the moment few successful clinical applications [Yannas 2001], however it is a promising field of active research.

The objective of regenerative medicine is to develop treatments that can induce regeneration in injured organs [Orlando et al. 2008]. Such treatments would change the wound healing process *in vivo*, leading to the formation of new tissue of normal structure and function. Current research efforts towards this goal include the use of biomaterials [Yannas 2001], decellularized organs [Uygun et al. 2010; Badylak et al. 2011; Orlando et al. 2011], small molecules [Zhu et al. 2011], and stem cells [Pomerantz and Blau 2004; Tanaka and Galliot 2009].

This study focuses on the application of biomaterials to induce regeneration in severe irreversible peripheral nerve injuries. Peripheral nerves and skin are among the few organs where commercially available biomaterial treatments for inducing regeneration exist [Abou Neel et al. 2013]. By studying how the properties of biomaterial affect wound healing in these organs the objective is to acquire understanding that could be utilized in the development of biomaterials that could induce regeneration in a wider variety of organs.

4.2.2 Irreversible Injury and Spontaneous Wound Healing in Peripheral Nerves

The Clinical Need for Peripheral Nerve Regeneration Treatments

Severe injury in peripheral nerves is an important clinical challenge. Approximately 5% of all accidents in US involve peripheral nerve injury (PNI) [IJkema-Paassen et al. 2004]. In US, approximately 200000 patients are treated each year for PNI that requires surgical intervention [Madison et al. 1992]. The current clinical practice for PNI is that small-size injuries are treated by suturing the nerve stumps, while larger defects are treated using nerve autografts [Schmidt and Leach 2003; Bozkurt et al. 2012]. Both approaches suffer from severe limitations. The PNI defect size that can be treated by suturing is limited, because the required tension to bring stumps in proximity impedes nerve regeneration and increases scar [Schmidt and Leach 2003; IJkema-Paassen et al. 2004]. Perineural suturing after fascicle alignment is possible but induces intraneurial fibrosis [IJkema-Paassen et al. 2004]. Peripheral nerve autografts (e.g. from the sural nerve) are biocompatible, have been used to bridge gaps at least 1.5 - 7cm long [IJkema-Paassen et al. 2004], and are considered today's golden standard. Nevertheless autografts result in a second PNI, and their success rate (fraction of patients that regain useful function) is 40-50%. Clearly, there is significant clinical need in developing better treatments for inducing regeneration in severely injured peripheral nerves).

Several methods have been considered as alternatives to autograft treatment of severe PNI. The use of non-autologous nervous tissues has larger supply compared to autografts, however there is danger of disease transmission and need for immunosuppression. The use of processed non-autologous tissues (for example decellularized organs obtained by removing immunogenic cells through chemical means) is a promising extension [Uygun et al. 2010; Badylak et al. 2011; Orlando et al. 2011]. A promising family of PNI treatments is based on using biomaterials to graft the injured organ to change the wound healing outcome and induce regeneration [Schmidt and Leach 2003].

Peripheral nerve anatomy

The peripheral nervous system (PNS) consists of the nerves and ganglia residing outside of the central nervous system (CNS, the brain and the spinal cord). The PNS connects the CNS to the limbs and organs, transmitting motor and sensory signals. Peripheral nerves (PN) contain two kinds of tissue elements: parenchymal (neuronal) and stromal (connective) [Soller 2011].

Parenchymal PN components mediate PNS function. They consist of nerve fibers surrounded by Schwann cells (SC) and often myelin. Each nerve fiber consists of a neuronal cell body (axon) that conducts action potentials. Nerve fibers may be unmyelinated or myelinated. In myelinated nerve fibers myelin is surrounded by SC cytoplasm, which is ensheathed in a basement membrane that contains collagen IV, laminin, hyaluronan, fibronectin, entactin, versican, chondroitin-sulfate and heparin sulfate proteoglycans [Carbonetto 1984; Tona 1993; Yannas 2001]. Non-myelinated SC interact with multiple nerve fibers.

The stroma of PN consists of three kinds of vascularized connective tissue structures [Thomas 1963; Flores et al. 2000; Kerns 2008]. The *endoneurium* is the innermost connective tissue that surrounds, and organizes the parenchymal components (axon-SC units). Endoneurium consists of a loose mesh of collagen and other ECM molecules, fibroblasts, and blood vessels surrounded by pericytes. The *perineurium* is the thin connective tissue that surrounds endoneurium and parenchymal components forming fascicles. Perineurium is composed of multiple layers of flattened perineurial cells (epitheloid, myofibroblasts) covered by basal lamina. Perineurial cells form tight, impermeable connections with each other forming a blood-nerve barrier that protects axons in the endoneurium [Parmantier et al. 1999]. The *epineurium* is the dense connective tissue that surrounds, holds together and protects the nerve fascicles (in many cases, PN contain multiple fascicles). After PN injury the pattern of ECM molecule expression changes dynamically.

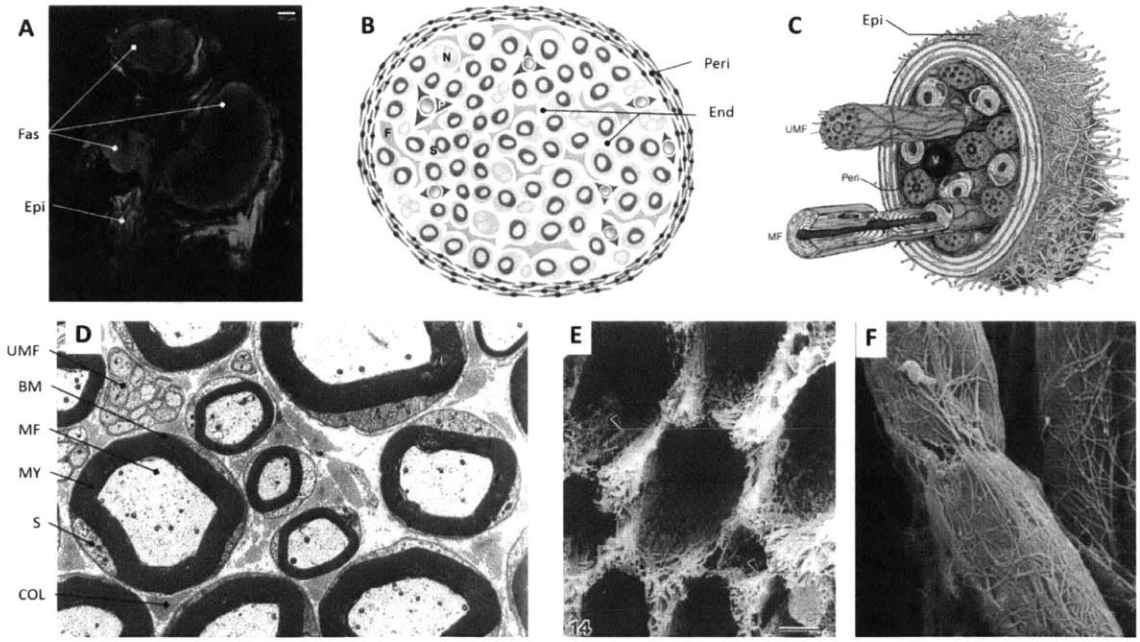


Figure 4.2.1: Structure and morphology of normal peripheral nerve tissues. **A**: Nonlinear image of a cross-section of a rat sciatic nerve. Red: cell cytoplasm. Green: collagen I. **B**: Schematic of a fascicle of a peripheral nerve [Joseph et al. 2004]. **C**: Schematic of the components of a peripheral nerve fascicle [Mizisin and Weerasuriya 2011]. **D**: Transition electron microscopy of endoneurial tissue in peripheral nerves [Schmalbruch 1986]. **E**: Scanning electron microscopy of endoneurial tissue in peripheral nerves after tissue decellularization by detergent shows the endoneurial collagen sheaths around axons and Schwann cells [Ushiki and Ide 1990]. **F**: Scanning electron microscopy showing endoneurial collagen fibrils surrounding the basement membrane of Schwann cells around myelinated nerve fibers [Friede and Bischhausen 1990]. MF: Myelinated nerve fiber, UMF & N: unmyelinated fiber, F: fibroblast, V: Blood vessel, End: Endoneurium, Peri: Perineurium, Epi: epineurium, S: Schwann cells, BM: basement membrane, MY: myelin, COL: collagen fibrils.

Irreversible Injury in Peripheral Nerves

As discussed in Section 4.2.1, the wound healing outcome after injury in adult organs depends on the nature of the injured tissue and the dimensions of the defect [Yannas 2001]. Studies of induced regeneration need to consider injuries that are severe enough so that they cannot regenerate spontaneously.

Several kinds of peripheral nerve injury (PNI) are considered in the literature, which differ in the identity of tissues injured and the extent of the injury (Figure 4.2.2). When only axons are injured (neurapraxia, axonotmesis) without damaging Schwan cells and surrounding basement membranes, the injury is considered reversible because axons after an initial period of degeneration and swelling will grow again and reach their targets. Such injuries, which can occur by nerve crushing, are therefore appropriate for studies of induced regeneration. More severe PNI include damage to the endoneurial tissue, the perineurial tissue, and finally complete transection of the peripheral nerve (nerotmesis). These injuries cause a defect to both the parenchymal components and the stroma tissues. Nerve transection creates a gap of length L between the two resulting nerve stumps.

After nerve transection the wound healing outcome depends on the length L of the gap between the two nerve stumps [Yannas et al 2007]. Nevertheless, nerve transection is a severe nerve injury frequently encountered in practice that usually is irreversible, therefore PN transection is a well-defined defect that can be used to study induced regeneration.

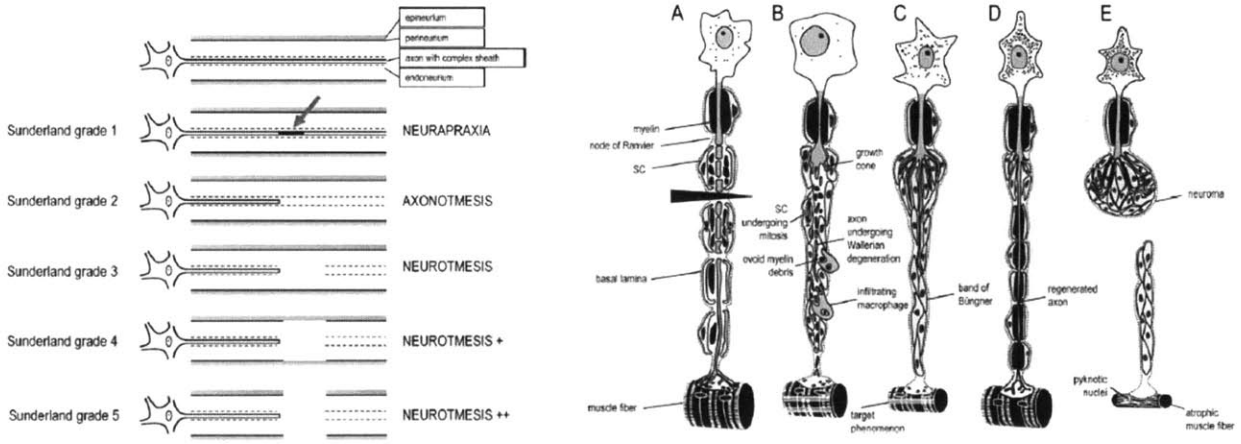


Figure 4.2.2: Peripheral nerve injury and its effects to axons. **Left**: The five degrees of nerve injury according to Sunderland as classified based on the identity of the identity of tissues injured. **Right**: Schematic of the fate of an axon during peripheral nerve wound healing. A, B: Wallerian degeneration, C: Axon sprouting and elongation through Schwann cell columns (bands of Buengner), D: When PN regenerates successfully, axon crosses the gap and reconnects with the target organ, E: When axon sprouts fail to cross the injury site a neuroma is formed [Deumens et al. 2010].

Wound Healing in Transected Peripheral Nerves

The formation of nervous tissue (myelinated axons, glia, supporting ECM) during wound healing in transected peripheral nerves consists of five main steps [Williams et al. 1983; Yannas 2001; Schmidt and Leach 2003], see Figure 4.2.3:

1) exudate flows into the gap from both stumps (within a few days).

2) formation of fibrin matrix from plasma precursors in the exudate (within 1 week).

3) migration of endoneurial cells (schwann cells, endothelia, fibroblasts) into the fibrin matrix and remodeling of the fibrin matrix into endoneurial tissue (within approximately 4 weeks).

4) elongation and myelination of axon sprouts from the proximal stump towards the distal stump through endoneurial tissue (lags a few weeks behind endoneurial cell migration).

5) myelination of axons that bridged the gap and reach target organ (within a couple of years).

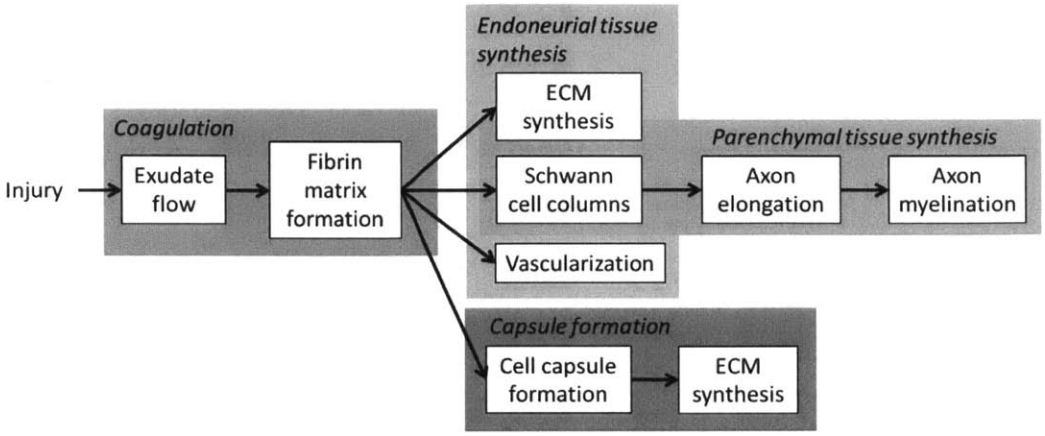


Figure 4.2.3: Schematic of the elementary processes of PN wound healing after transection.

A detailed description of these elementary steps can be found in section 4.2.4. Successful regeneration of transected PN requires the successful outcome of all five steps. The successful outcome of each step depends strongly on the outcome of previous steps:

- The formation of a continuous large-diameter fibrin matrix that bridges the two nerve stumps (step 2) cannot take place unless the exudate provides necessary fibrin components (step 1) [Williams et al. 1984].
- The fibrin matrix (step 2) functions as the scaffold for the migration of endoneurial cells (fibroblasts, Schwann cells, endothelia cells) into the gap (step 3) [Fu and Gordon 1997].
- Axons cannot cross the gap and reach the target organ (step 4) unless endoneurial cells form appropriate structures (Schwann cell columns and vascularization) throughout the gap (step 3), and unless axons are attracted by chemical clues from the distal stump.
- Axons cannot survive and get myelinated (step 5) unless they interact with schwann cell columns (step 3) and manage to reach the target organ (step 4) [Fu and Gordon 1997].

Experimental data suggest that in parallel with nerve tissue synthesis, a structurally distinct tissue forms in the outer part of the newly-formed tissue, see Figure 4.2.3. This tissue, referred to as "contractile cell capsule", consists of contractile cells and connective tissue of circumferential-longitudinal orientation. The role of this capsule in the wound healing outcome is of particular interest in this study. According to the "pressure cuff" theory the contractile cell capsule impedes nerve regeneration by interfering with the steps of nerve tissue synthesis.

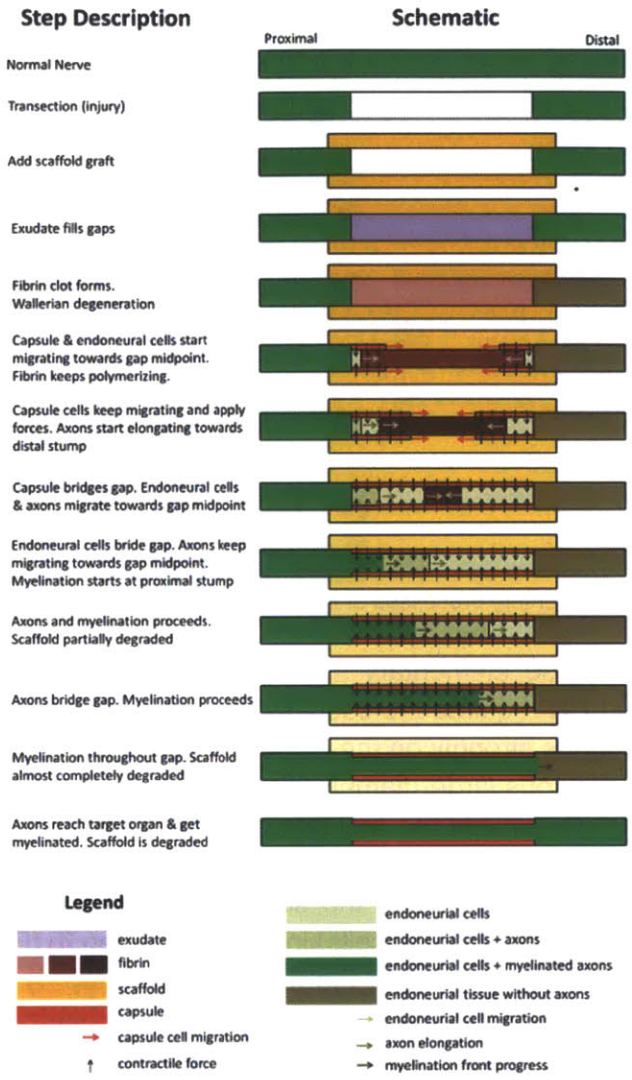


Figure 4.2.4: Schematic of the response of the elementary processes of wound healing after peripheral nerve transection. The tissue composition of the nerve is shown in lateral sections at various time instances.

Peripheral Nerve Wound Healing Outcomes

Depending on the outcome of the elementary processes of PN wound healing, the outcome of the wound healing process is classified as:

- Nerve trunk: all five steps succeed at least partially: an endoneurial bridge forms, a fraction of the axons manage to elongate through the gap and get myelinated. The nerve trunk consists of endoneurial tissue organized in mini-fascicles, surrounded by a contractile cell capsule.
- Tissue cable: an endoneurial bridge forms but axons fail to elongate through the gap. Tissue cable consists of axon-free endoneurial tissue surrounded by a contractile cell capsule.
- Neuroma: an endoneurial bridge does not form because either a fibrin matrix didn't bridge the gap or because endoneurial cells didn't bridge the gap. Neuroma consists of connective tissue and immature axons (proximal stump only) surrounded by a contractile cell capsule that ensheaths each stump.

4.2.3 Elementary Processes of Peripheral Nerve Wound Healing

This section summarizes experimental observations for the elementary steps of wound healing This study considers the standard Lundborg model for studying peripheral nerve regeneration (PNR): the peripheral nerve is transected, and the resulting nerve stumps are separated by a gap and tabulated in a conduit. The majority of *in vivo* studies on PN wound healing have been conducted in transected rat sciatic nerves treated with silicone conduits [Lundborg et al. 1982a, c; Williams et al. 1983, 1984, 1985, 1987; Williams 1987; Kemp 2009]. This treatment has been shown to result in some degree of regeneration when the resulting nerve injury is up to 10mm [Yannas et al. 2007]. Data from transected rat sciatic nerves treated with silicone conduits will be used as the baseline for comparison with other treatments. Unless otherwise mentioned, in this section data come from experiments in transected rat sciatic nerves entubulated with silicone conduits.

Exudate Flow and Fibrin Matrix Formation

After nerve transection, exudate liquid flows from both nerve stumps into the gap. A 10mm gap of 11μl volume treated by an empty silicone conduit fills within approximately 12 hours [Williams et al. 1985]. Through the process of coagulation plasma components in the exudate are converted into a precursor fibrin matrix. The formation of the fibrin matrix takes place over a period of few days as additional plasma precursors enter from the nerve stumps and diffuse inside the gap. This precursor matrix consists mostly of fibrin fibers, recognized by a characteristic 20nm periodicity, and contains macrophages that remove cell debris [Williams et al. 1987]. Meanwhile, at the distal nerve stump macrophages and Schwann cells remove debris of myelinated axons (Wallerian degeneration) and start releasing axon growth promoting factors. Myelin degradation is accomplished by Schwann cells the first two days, and mostly by infiltrating macrophages afterwards [Fu and Gordon 1997].

One week post-injury the resulting fibrin matrix has a characteristic tampered shape (its diameter diminishes near the gap midpoint) and does not make contact with the surrounding conduit even in gap lengths as small as L=4mm. Instead, the fibrin matrix is separated from the conduit by a fluid [Lundborg et al. 1982a,c; Williams et al. 1983, 1984, 1985]. One week post-injury the fibrin matrix does not stain for laminin, stains strongly for fibronectin, and contains endoneurial cells (migrating into the gap) only in the proximity of nerve stumps [Longo et al. 1984; Liu 1992; Zhao et al. 1992]. Fibrin fibers have preferable longitudinal-circumferential orientation both at 3 days and 7 days post-injury [Williams et al. 1983; Schröder et al. 1993; Weis et al. 1994].

The formation of a fibrin cable does not always result in the subsequent formation of a nerve trunk. Although a fibrin cable may form in 20mm gaps treated with silicone conduits [Williams et al. 1987], a myelinated nerve trunk generally does not form for gaps longer than 10mm [Yannas et al. 2007]. The possibility of forming a nerve trunk after forming a fibrin cable decreases as the gap length increases [Lundborg et al.1982c], possibly due to the small diameter of the resulting fibrin matrix.

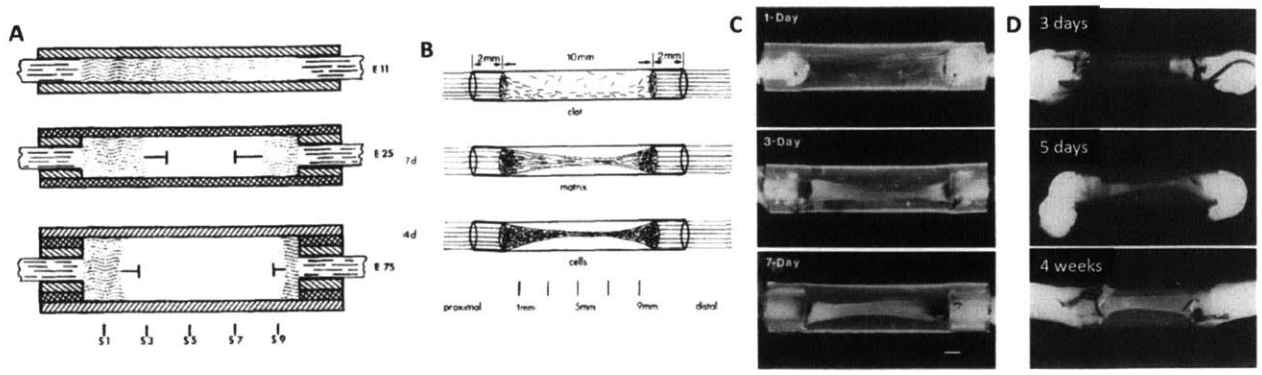


Figure 4.2.5: Exudate flow and fibrin matrix formation in transected nerves treated by silicone conduits. **A:** Schematic of exudate flow in 10mm gaps treated by empty silicone conduits of inner diameter (ID) 1.2mm (E11), 1.8mm (E25) or 3.1mm (E75) [Williams et al. 1985]. **B**: schematic of fibrin matrix and subsequent nerve trunk formation in transected nerves treated with silicone conduits [Longo et al. 1984]. **C**: Fibrin matrix formation in a 10mm rat sciatic nerve gap model treated with saline pre-filled 1.8mm ID silicone conduits [Williams et al. 1987]. **D**: Formation of fibrin matrix (3d, 5d) and nerve trunk (4w) a 6 mm rat sciatic nerve gap model treated with a 1.4mm ID silicone conduit [Liu 1992].

Endoneurial Tissue Synthesis

After fibrin matrix formation, several kinds of cells migrate from both stumps into the gap, using the fibrin matrix as a scaffold. Experimental observations suggest that two distinct cell migration phenomena take place in parallel from both nerve stumps [Williams et al. 1985], see Figure 4.2.4:

- Endoneurial cells (Schwann cells, fibroblasts and endothelia cells) migrate in the core of the fibrin matrix. Endoneurial cells form critical cell structures (blood vessels, Schwann cell columns) that support axon elongation, and remodel the extracellular matrix (fibroblasts, Schwann cells) by degrading the fibrin/fibronectin matrix and synthesizing endoneurial stroma.
- Perineurial cells migrate in the outer surface of the fibrin cable. These perineurial cells are believed to be the origin of the capsule of contractile cells and connective tissue that forms in outer surface of the nerve trunk.

In some cases endoneurial cells manage to migrate through the fibrin matrix, bridge the gap between the two stumps, and synthesize endoneurial tissue that connects the two nerve stumps (an endoneurial bridge). In 10 mm gaps treated with silicone conduits, the two fronts of endoneurial cells meet sometime between 2 and 3 weeks post-injury [Williams et al. 1983, 1985; Müller et al. 1987]. In more challenging wounds the two fronts of migrating endoneurial cells do not meet and an endoneurial bridge does not form. The formation of the fibrin matrix affects strongly the formation of the endoneurial tissue [Williams et al. 1985]. One possible reason for failing to form an endoneurial bridge is that the fibrin matrix precursor did not bridge the gap [Williams et al. 1984]. A second possible reason is that the progress of the endoneurial cell front is much slower than the formation of the contractile cell capsule.

As endoneurial cells migrate inside the fibrin matrix towards the gap midpoint, they gradually remodel the fibrin matrix, and change its composition and organization. The fibrin matrix is progressively replaced by endoneurial tissue consisting of Schwann cell columns, blood vessels, and extracellular matrix. Endoneurial tissue may be organized in mini-fascicles.

Schwann cells are the key cells of peripheral nerve regeneration [Fu and Gordon 1997] because axon elongation through the gap requires the presence of Schwann cell columns. After nerve

transection, Schwann cells at the stumps are stimulated by inflammation, change their phenotype from "myelination mode" (myelinate axons, contain basement membrane) to "growth support mode", proliferate, and migrate in the fibrin matrix forming columns called bands of Buengner. Schwann cell proliferation takes place in two steps. The first step peaks 2-3 days post-injury and is caused by loss of axon contact and mitogens secreted by myelin-digesting macrophages. The second step takes place later, when Schwann cells in the gap establish contact with elongating axons [Fu and Gordon 1997]. The formation of a fibrin clot is not by itself sufficient to promote Schwann cell migration into the gap. Schwann cell adhesion to fibrin promotes proliferation [Akassoglou et al. 2002a] but impedes migration; instead Schwann cells can migrate on fibronectin [Akassoglou et al. 2003]. Molecular clues (such as VEGF) released by cells at the distal nerve stump seem to be necessary to induce Schwann cell migration [Sondell et al. 1999]. Accordingly, it was shown that the fibrin matrix formed when the conduit distal end is left open (PO) is similar to the fibrin matrix formed when the conduit distal end entubulates the distal nerve stump (PD); nevertheless, Schwann cell and axon migration into the fibrin clot is observed only in the PD case [Williams et al. 1984]. Accordingly, the endoneurial cell front does not reach the middle of a 5mm gap within 4 weeks when the distal end of the silicone conduit is left open [Lundborg et al. 1982c]. Although the presence of nerve tissue at the distal conduit end can guide the elongation of axons, it seems that Schwann cells show no preferential growth towards nervous tissues [Zhao et al. 1992]. Schwann cells migrate into the fibrin matrix and form columns even when axon elongation is prevented [Zhao et al. 1992]. However, the formation of basement membrane by Schwann cells is stimulated by the presence of axons [Carbonetto 1984; Liu 1992]. The inability of inducing regeneration in large gaps is attributed to limited migratory ability of Schwann cells [Fu and Gordon 1997].

The formation of vasculature is reported to lag slightly behind the front of migrating Schwann cells and fibroblasts [Williams et al. 1983, 1984, 1985; Schröder et al. 1993]. Endothelia cells migrate inside the gap and form blood vessels even when Schwann cells do not migrate [Williams et al. 1984; Aebischer et al. 1990]; however the opposite is not expected. Most of the new blood vessels are located in the outer part of the endoneurial tissue, close to the capsule [Schröder et al. 1993; Zochodne and Nguyen 1997]. Early blood vessels have small diameter (2-8μm). Over time the diameter of new blood vessels increases, and 8 weeks post-injury there are more and larger blood vessels in a nerve trunk compared to a normal nerve [Jenq and Coggeshall 1985a; Müller et al. 1987; Zochodne and Nguyen 1997]. Experimental results show that the final diameter of the nerve trunk correlates with the extent of neovascularization [Zhao et al. 1992]. It is proposed that neovascularization drives Schwann cell migration and axon elongation in the gap and therefore affects significantly the wound healing outcome [Schröder et al. 1993; Zochodne and Nguyen 1997; Kemp et al. 2009].

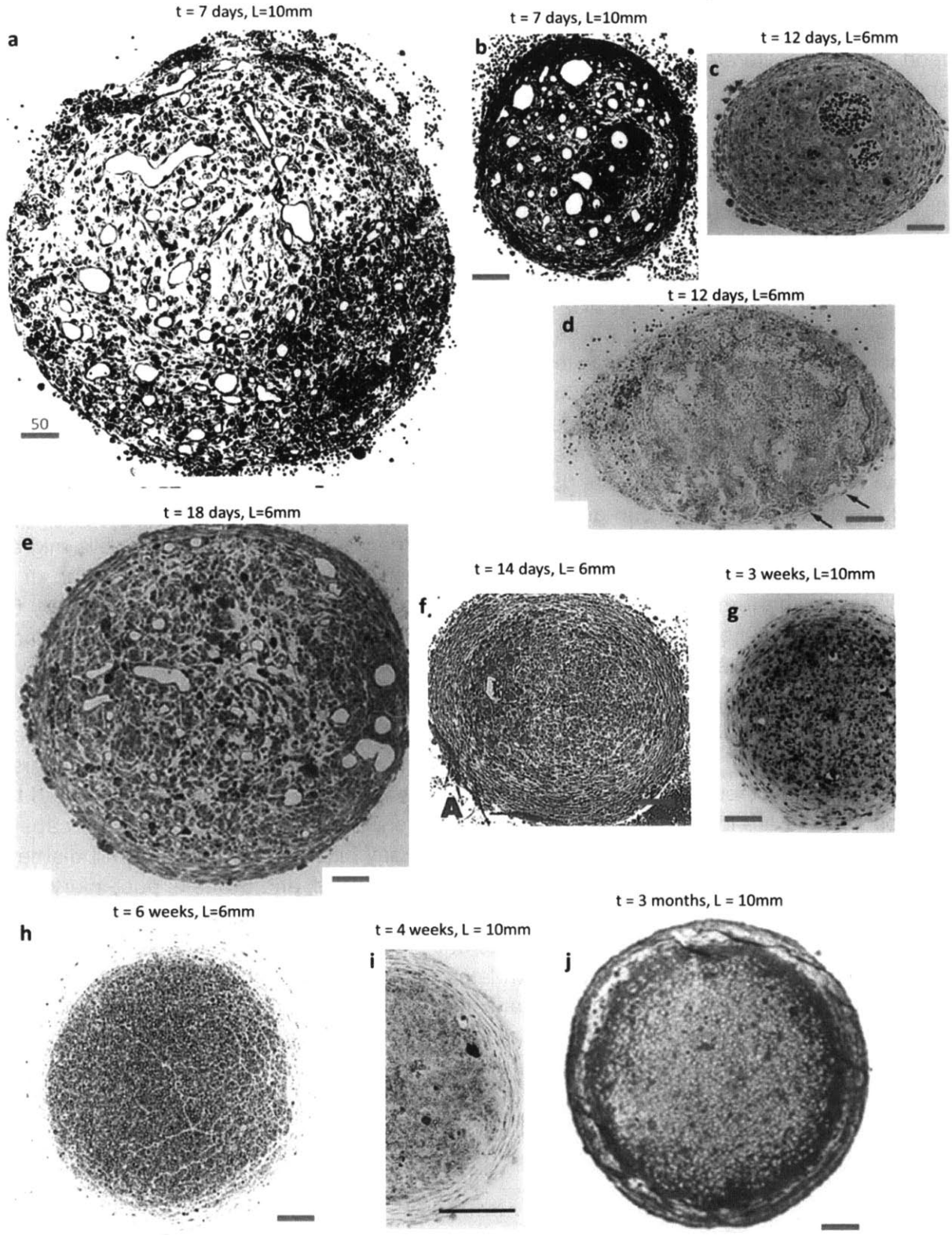


Figure 4.2.6: Cross sections of transected rat sciatic nerves of various gap length L (6 – 10 mm) treated with silicone conduits, at various time points t post-injury (7 days - 3 months). All samples are located in the gap midpoint except (a) which is located at the proximal nerve stump. The inner diameter of the silicone conduits is either 1.1mm (a-e) or 1.2mm (f-j). All sections demonstrate the presence of the contractile cell capsule around the nerve regenerate.

References: (a,b): [Schröder et al. 1993], (c-e): [Weis et al. 1994], (f,h,j): [Lundborg et al. 1982],(g): [Williams et al. 1985], (i): [Williams et al. 1984].

The ECM produced by endoneurial cells (fibroblasts, Schwann cells) contains collagen, fibronectin and basement membrane proteins (laminin, proteoglycans, glycosaminoglycans). Newly-synthesized collagen fibers that take the place of fibrin matrix have been observed as soon as 8 days post-injury [Liu 1992]. One week post-injury the fibrin matrix stains for fibronectin but not for laminin. Two weeks post-injury endoneurial cells in the gap stain for both fibronectin and laminin. In both proximal and distal nerve, the fibronectin/laminin staining pattern reveals tubular structures of mean radius around 25 µm (proximal side) and 15µm (distal side) [Longo et al. 1984]. The expression pattern key components (proteoglycans, laminin) found in SC basement also changes drastically over time reflecting the SC state [Tona et al. 1993].

The front of Schwann cells is followed by the front of elongating axons. At locations reached by elongating axons, the newly synthesized endoneurial tissue gets organized into mini-fascicles (also referred as "regeneration units"): groups of Schwann cells wrapping one or more elongating axons separated by collagen fibers, and surrounded by elongated perineurial-like cells. It is believed that mini-fascicle formation is caused by the compromise of the blood-nerve barrier imposed at the perineurium. [Fu and Gordon 1997] mentions that axons attached to each mini-fasicle are sprouts of a common parent axon. When axon elongation is prevented, the newly-formed endoneurial tissue contains Schwann cell columns but no mini-fascicles, which suggests that axon presence is necessary for mini-fascicle formation [Popovic et al. 1994; Zhao et al. 1992]. The presence of more axons correlated with the formation of fewer and larger mini-fascicles [Popovic et al. 1994].

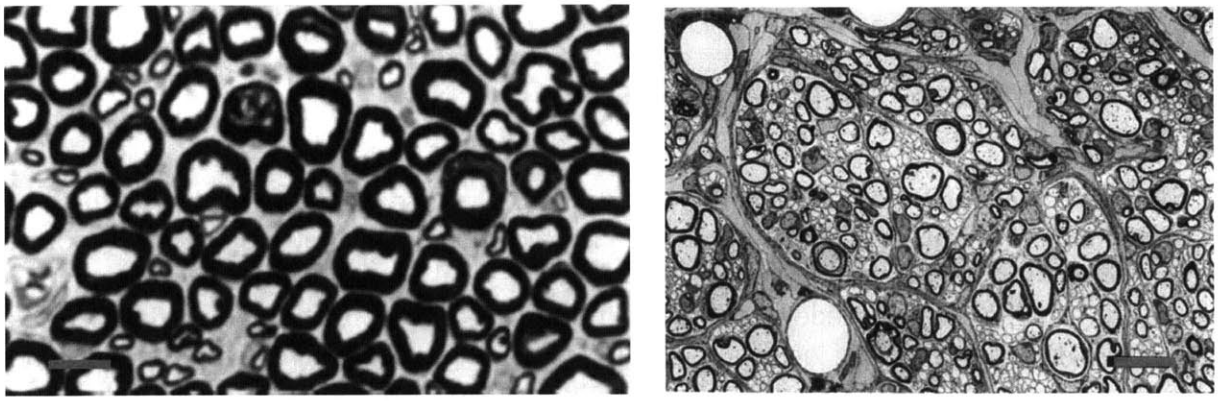


Figure 4.2.7: Formation of mini-fascicles in nerve trunks. **Left:** OsO_4 stain of myelinated axons in normal endoneurium [Chamberlain et al. 1998a]. **Right:** OsO4 staining demonstrates mini-fascicle organization of the newly-formed endoneurial tissue 8 weeks post-injury in a 8mm gap treated with a silicone conduit: myelinated and unmyelinated axons supported by SC are surrounded by elongated fibroblasts [Jenq et al. 1985]

Parenchymal Tissue Synthesis

Soon after nerve transection, surviving axons undergo major morphological and molecular expression changes (chromatolysis) and switch from "transmitting mode" to "grow mode" [Fu and Gordon 1997]. The proximal stump swells, possibly due to "damming" of proteins and organelles transported via the axons. Surviving axons usually degenerate to the closest node of Ranvier, where they subsequently sprout. A single axon gives rise to 50-100 branches. Most sprouts disappear soon. The basement membrane formed by Schwann cells supports and

enhances axon elongation [Fu and Gordon 1997]. When an insoluble substrate (endoneurial tissue) has been synthesized by Schwann cells and fibroblasts inside the gap, multiple axon sprouts (average 5, maximum 20 per axon) will migrate in the gap. If no support structure for axon sprouts is available, axons cannot elongate into the gap, and a neuroma forms.

Axon sprout migration into the gap is guided by chemical and morphologic clues [Fu and Gordon 1997; Chen et al. 2007]. Axons do not grow in exudate [Lundborg et al. 1982a], and they do not interact directly with the fibrin matrix or granulation tissue [Williams et al. 1983; Liu 1992]. Instead, elongating axons interact extensively with columns of Schwann cells [Lundborg et al. 1982a], which synthesize proteins known to facilitate axon growth in vitro such as fibronectin and laminin [Longo et al. 1984]. The liquid surrounding the nerve trunk contains neurite-promoting factors, known to promote axon growth [Carbonetto 1984; Longo et al. 1984]. If axon sprouts manage to cross the gap, they will continue to elongate inside the empty (due to Wallerian degeneration) endoneurial sheaths at the distal nerve stump. The degradation of myelin in the distal stump during Wallerian degeneration is therefore a necessary prerequisite for axon elongation and reconnection to the target organ [Fu and Gordon 1997].

Axon myelination progresses from the proximal stump towards the distal stump, and lags behind axon elongation. Myelination is initiated when Schwann cells contact the axolemma of elongating axons. Schwann cells synthesize and wrap myelin sheath around axons. The thickness of the myelin sheath is determined by the diameter of the axon [Fu and Gordon 1997]. Schwann cell myelination activity is inhibited by fibrin [Akassoglou 2002] but it is enhanced by interactions with axons and by the presence of collagen extracellular matrix.

If the daughter sprouts of an axon fail to bridge the gap and reconnect to the target organ, axons sprouts contract back to proximal stump. After target connection few of the daughter axons survive, all but few of the daughter sprouts of each axon will degrade over a period of a few years [Fu and Gordon 1997; Gordon 2009].

Capsule Formation

Several studies have highlighted that soon after an insoluble fibrin matrix forms, perineurial cells migrate from the nerve stumps on the outer surface of the fibrin matrix towards the gap midpoint [Williams et al. 1983; Schröder et al. 1993; Weis et al. 1994]. This cell migration takes place in parallel with the migration of endoneurial cells in the core of the fibrin matrix. Within 2 weeks post-injury, a tissue composed of concentric layers of longitudinally-circumferentially aligned contractile cells and collagenous matrix forms in the outer part of the nerve trunk. This tissue, referred in this study as "capsule", surrounds the newly-synthesized endoneurial tissue. In the literature this tissue is referred to as "epineurial cells" [Lungborg et al. 1982a,b,c], "circumferential cells" [Williams et al. 1984, 1985; Müller et al. 1987], "perineurial connective tissue"/"perineurial tube"/"perineurial ensheathment" [Jenq and Coggeshall 1985a; Schröder et al. 1993; Weis et al. 1994], "epineurium"/"epineurial-like tissue"/"epineurual sheath" [Jenq and Coggeshall 1985b; Jenq et al. 1987; Aebischer et al. 1989, 1990], "connective tissue capsule" [Kemp et al. 2009], or "contractile cell capsule"/"myofibroblast capsule" [Chamberlain et al. 2000; Soller et al. 2012]. Many studies refer to the capsule cells as "perineurial" cells because the capsule is in continuum with the perineurium of the intact nerve stumps [Williams et al. 1984; Weis et al. 1994], and because 7 days post-injury capsule cells stain positively for MUC1 (also known as EMA), a marker of perineurial cells [Schröder et al. 1993; Weis et al. 1994].

Interestingly, MUC1 is also expressed in myofibroblasts [Kamoshida and Tsutsumi 1998]. Studies at later time instances show that elongated cells in the capsule stain positively for αSMA [Chamberlain et al. 1998a, 2000], the major marker of contractile myofibroblasts (MFB) [Desmoulière et al. 2005]. Perineurial cells have several structural properties of MFB (spindle shape, pinocytotic vesicles, dense actin filaments and well-developed Endoplasmic reticulum [Chamberlain et al. 1998a]) and also stain for vimentin, a marker of cells of mesenchyme origin [Weis et al. 1994].

Experimental observations suggest that the composition of the capsule changes over time. At early time points the capsule consists entirely of cells with extensive cell-cell contacts. Over time the capsule becomes progressively richer in extracellular matrix [Lundborg et al. 1982a], consisting mostly of circumferentially-oriented collagen fibers [Schröder et al. 1993]. 2 weeks post-injury the capsule also stains for fibronectin [Longo et al. 1984]. This observation is related to the presence of MFB in the capsule, since the EDA splice variant of fibronectin is a known prerequisite of MFB differentiation [White et al. 2008].

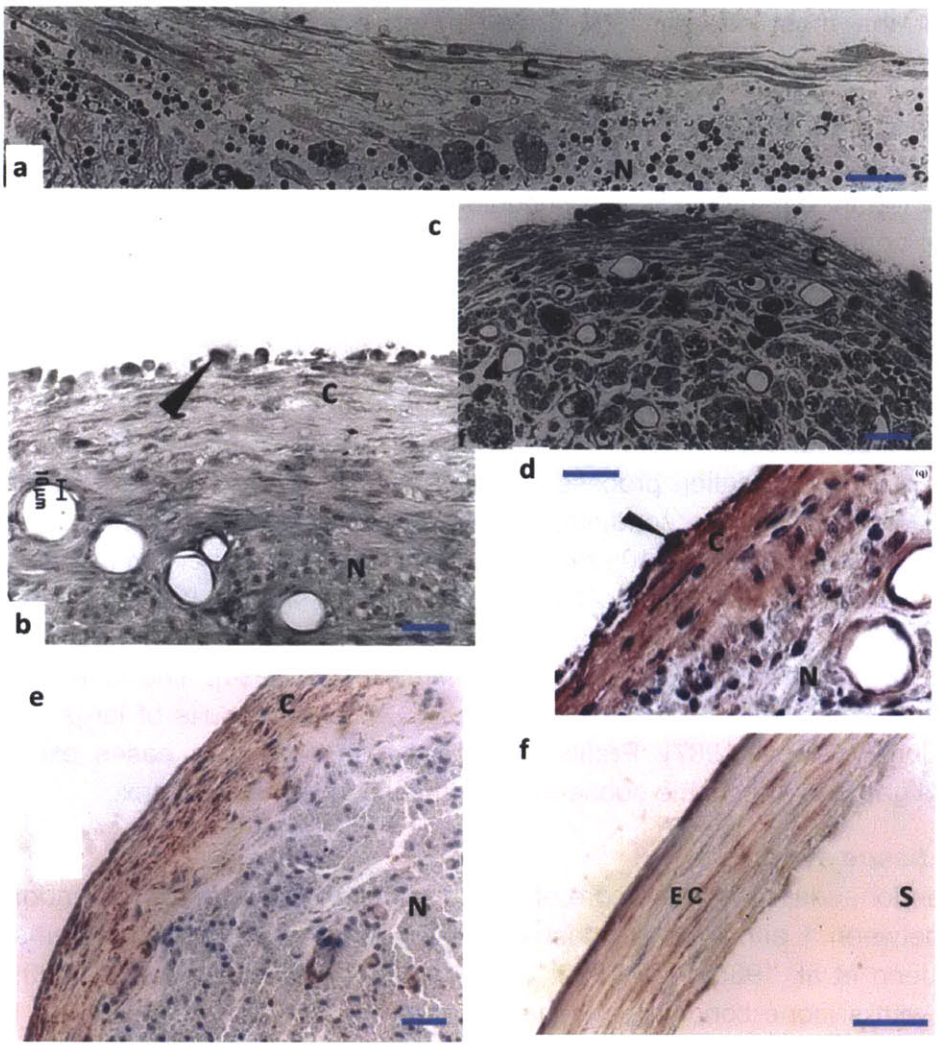


Figure 4.2.8: Snapshots of contractile cell capsules formed in transected nerves treated with silicone conduits at various time points post-injury. **a**: Longitudinal section of OsO_4 treated epon-embedded tissue. 6mm gap, 7 days post-injury, 1.2mm ID silicone conduit [Weis et al. 1994]. **b**: Cross section of epon-embedded tissue. Midpoint of a 10 mm gap, 6 weeks post-injury, 1.5mm ID silicone conduit. Arrow points macrophages surrounding a collagenous

capsule [Chamberlain et al. 1998a]. **c**: Cross section of OsO_4 treated epon-embedded tissue. Midpoint of a 6mm gap, 18 days post-injury, 1.2mm ID silicone conduit [Weis et al. 1994]. **d**: Cross section of paraffin embedded tissue immune-stained for αSMA. Midpoint of a 10 mm gap, 6 weeks post-injury, 1.5mm ID silicone conduit [Chamberlain et al. 1998a]. **e**: Cross section of paraffin embedded tissue immune-stained for αSMA. Midpoint of a 10 mm gap, 60 weeks post-injury, 1.5mm ID silicone conduit [Chamberlain et al. 1998a]. **f**: Cross section of paraffin embedded tissue immune-stained for αSMA. 10 mm gap, 60 weeks post-injury, 1.5mm ID silicone conduit. Image shows the contractile cell capsule that forms in the outer surface of the silicone conduit [Chamberlain et al. 1998a]. C: contractile cell capsule around nerve tissue. N: nerve tissue, S: silicone conduit, EC: external contractile capsule around the silicone conduit.

4.2.4 Kinetics of Elementary Processes of Peripheral Nerve Wound Healing

This study focuses on the early response of contractile phenotypes during PN wound healing and its regulation by biomaterials, therefore emphasizes the role of mechanics and kinematics in determining the wound healing outcome. This section summarizes experimental observations about the kinetics of the elementary steps of PN wound healing (Figure 4.2.4). Unless otherwise mentioned, data presented in this section are obtained from transected rat sciatic nerves where gaps of length L=10 mm were treated with silicon conduits of 1.2 mm ID. Even though silicone tubes are not the most effective PN grafts in terms of regenerative ability, this section summarizes finding from silicone tubes because they are the most widely used in the literature.

Exudate Flow and Fibrin Matrix Formation

After nerve transection, exudate flows out of each nerve stump at rate 0.9µl/hour (rat sciatic nerve). Inside 1.2mm ID empty silicone conduits, this exudate flow rate corresponds to a front propagating at approximately 138 mm/week. The time required for exudate to fill the gap inside the conduit is analogous to the gap volume (Figure 4.2.5). Exudate fills a 10 mm gap treated by a 1.2mm ID empty silicone conduit within 12h (11µl volume) [Williams et al. 1985].

The formation of the fibrin matrix happens progressively within a few days after injury. In L = 10 mm gaps treated with 1.2mm ID empty silicone conduits, a matrix forms within 3 days post-injury. Since it takes 12h for exudate to fill this gap [Schröder et al. 1993; Weis et al. 1994], the time constant of the coagulation process is estimated on the order of 12 - 15 hours. In gaps treated with 1.2mm ID silicone conduits, 3 days post-injury the fibrin matrix seems to cover almost the whole volume of the 10mm gap; 7 days post-injury the fibrin matrix diameter is reduced significantly and the fibrin cable is more tamper-shaped [Lundborg et al. 1982a,c; Williams et al. 1983, 1984, 1985]. This fibrin diameter reduction is attributed to mechanical contraction or retraction [Schröder et al. 1993; Weis et al. 1994]. The kinetics of fibrin clot formation can be modulated by the experimentalist. Silicone conduits of larger ID delay fibrin matrix formation [Williams 1987]. Prefilling the gap with a solution eases exudate diffusion throughout the gap and affects the subsequent formation of the fibrin matrix.

Endoneurial tissue synthesis

Migration of endoneurial cells in the core of the fibrin matrix and formation of endoneurial tissue takes place between 1 and 4 weeks post-injury [Lundborg et al. 1982a; Williams et al. 1983, 1984, 1985; Jenq et al. 1985a; Liu 1992; Schröder et al. 1993; Kemp et al. 2009]. In 10 mm gaps treated with silicone conduits, endoneurial cell migration can be described by two fronts that start from each nerve stump after a latency period of 3 days (coincides with the formation of a solid fibrin matrix) and progresses with mean velocity 2.5 mm/week towards the gap midpoint [Williams et al. 1983, 1984; Müller et al. 1987; Schröder et al. 1993].

Fast Schwann cell migration is expected to favor regeneration. 1 week post-injury Schwann cells lacking basement membranes have migrated 1mm away from the stumps [Williams et al. 1983; Liu 1992]. Bands of Buenger start forming 1 week post-injury [Liu 1992], and are visible 2 weeks post-injury [Longo et al. 1984]. The formation of basement membrane by Schwann cells in contact with elongating axons and is visible as early as 2 weeks post-injury [Carbonetto 1984; Liu 1992].

Similarly, the formation of nee blood vessels can be described by two fronts that start from each nerve stump after a latency period of 5-7 days and progress with mean growth rate 2.5-3 mm/week (measured between 2 to 3 weeks post-injury). A 10 mm gap is fully vascularized within approximately 3 weeks [Williams et al. 1983; Schröder et al. 1993]. The progress of blood vessel fronts approximately matches the progress of the fronts of migrating Schwann cells and fibroblasts [Williams et al. 1983, 1984, 1985; Müller et al. 1987; Schröder et al. 1993].

Mini-fascicles are visible as early as 12 days post-injury [Lundborg et al. 1982a; Scaravilli 1984], and persist at later time points [Lundborg et al. 1982b; Schröder et al. 1993]. In challenging nerve injuries (gap lengths 15-30mm) treated with collagen guides, mini-fascicles of mean diameter 25-50 μm are observed at 6, 9, or 12 weeks post-injury [Yoshii et al. 2003; Bozkurt et al. 2012; Soller et al. 2012].

Axon Elongation and myelination

In 10 mm gaps treated with silicone conduits, nonmyelinated axons are observed up to 1 mm away from the proximal stump within 1 week, have bridged half the gap within 2 weeks, and the whole gap within 3 weeks. This corresponds to an axon front that starts from the proximal stump after a latency period of approximate 6 days, and grows with mean rate 1.5 mm/week (measured between week 1 to 2 post-injury) and 6 mm/week (measured between week 2 to 3 post-injury) [Williams et al. 1983, 1984]. These kinematic parameters agree with several independent observations [Lundborg et al. 1982a; Schröder et al. 1993; Weis et al. 1994; Kemp et al. 2009]. Axon elongation is much slower inside granulation tissue (1.75 mm/week) but much faster when axons follow the empty endoneurial tubes after nerve crush (4.4mm/day) [Williams et al. 1983; Popovic et al. 1994].

Metabolic studies in axotomized neurons, suggest that the growth potential of regenerating axons peaks around 3 weeks post-injury [Fu and Gordon 1997]. Between 4 and 8 weeks post-injury, axon number and diameter increase in 5 or 10 mm gaps treated with silicone or collagen conduits [Kemp et al. 2009]. 9 months post-transection, in the midpoint of a 4mm gap treated with silicone conduits the number of axons is double compared to that of a normal nerve due to axon sprouting. In the distal stump of 4mm or 8mm gaps treated with silicone conduits, there are 50% more unmyelinated and myelinated axons compared to normal nerves [Jenq et al. 1985a].

In 10 mm gaps treated with silicone conduits, myelinated axons are observed within 1mm of the proximal stump within 2 weeks post injury, reach the gap midpoint within 3 weeks, and bridge the gap within 4 weeks [Williams et al. 1983]. The progress of axon myelination front can be described by a front that starts after a latency period of 11-12 days and grows with mean rate 3-4 mm/week [Williams et al. 1983, 1984]. These kinematic parameters agree with several independent experimental observations [Lundborg et al. 1982a; Liu 1992; Weis et al. 1994]. The maturation of myelination fibers (myelin thickness increase) continues for much longer. In

comparison, the estimated mean rate of the axon myelination front in autografts is 1-2mm/w [Bozkurt et al. 2012].

Capsule formation

Axial growth: In transected nerves treated with silicone tubes, as soon as the fibrin matrix is formed, cells start migrating from the nerve stumps matrix towards the gap midpoint on the outer surface of the fibrin [Williams et al. 1983; Schroeder et al. 1993; Weis et al. 1994]. In 10mm gaps treated with silicone conduits 7 days post-injury capsule cells have migrated 2-3 mm into the gap, and 12 days post-injury capsule cells have migrated 4.75mm. The migration of capsule cells along the nerve axis can be described by two fronts that start from each stump after a latency period of approximately 3 days that grow with mean rate 3.85 - 5 mm/week. Capsule cells cover the fibrin matrix in a 10mm gap within 14 days post-injury [Williams et al. 1983; Schröder et al. 1993; Weis et al. 1994], and a 6mm gap within 12 days post-injury. The axial progress of the capsule cell front on the fibrin matrix surface starts earlier and is faster than the progress of the endoneurial cell front that takes place simultaneously in the fibrin matrix core [Williams et al. 1985; Schröder et al. 1993].

Radial growth: In transected nerves treated with 1.1mm ID silicone conduits, the thickness of the capsule 1 week post-injury is progressively decreasing further away from the nerve stumps: 4 layers thick (about 20μm) at the proximal stump and a single cell layer (5μm) 2mm inside the gap [Schröder et al. 1993; Weis et al. 1994]. The thickness of the capsule increases at least until 21 days post-injury [Schröder et al. 1993]. 12 days post-injury images show a capsule 40μm thick close the midpoint of a 10mm gap (z=5mm), and 25-40μm thick close to the proximal stump of a 6mm gap [Weis et al. 1994]. 2 weeks post-injury the capsule thickness is aproximately 35 μm 1mm away from the nerve stumps, and 50 μm (10-15 cell layers) in the midpoint of a 6mm gap. 16 days post injury the capsule thickness 3mm away from the proximal stump (10mm gap, PBS filling) is 21μm [Müller et al. 1987]. 18 days post-injury the thickness of the capsule is 30μm approximately constant along a 6mm gap. 4 weeks post-injury the capsule is 36μm-thick 1mm away from the proximal stump (z=1), 40μm in the gap midpoint (z=5mm) and 55μm 1mm away from the distal stump (z=9mm) [Lundborg et al. 1982a; Williams et al. 1983; Schröder et al. 1993]. 6 weeks post-injury there is a collagenous (few cells) 50 μm-thick capsule in the midpoint of a 6mm gap [Lundborg et al. 1982a; Chamberlain et al. 2000]. In comparison, 6 weeks post-injury neuromas are ensheathed by collagenous capsules 40-50μm thick [Chamberlain et al. 2000]. After 6 weeks the thickness of the capsule does not increase significantly [Chamberlain et al. 2000]. 3 months post-injury, the nerve trunk formed after using silicone conduit to treat a 10mm gap is covered by a capsule approximately 40μm thick [Lundborg et al. 1982b]. The thickness of the capsule in the midpoint of 4 or 8mm gaps decreases between 8 weeks and 9 months post-injury from approximately 40μm to approximately 15μm [Jenq et al. 1985a]. The corresponding time constant of the radial grow of the contractile capsule from the initial (single cell layer) to the final thickness is approximately 1 week.

The structure of the contractile capsule also seems to change over time. 2 weeks post-injury the capsule consists mostly of contractile cells [Lundborg et al. 1982a] and there is extensive cell-cell contact between capsule cells (focal contacts connected to actin cytoskeleton, tight junctions, desmosomes) [Weis et al. 1994]. 21 days post-injury the capsule is more collagen-

rich, and capsule cells have more tight junctions [Schröder et al. 1993]. Between 2 and 5 weeks post-injury there is progressive synthesis of ECM layers between capsule cell layers [Williams et al. 1985; Aebischer et al. 1990; Schröder et al. 1993; Weis et al. 1994]. At later time points the capsule consists mostly of ECM and few contractile cells [Lundborg et al. 1982a]. The corresponding time constant for ECM formation in the capsule is also on the order of 1 week.

The following two tables summarize the parameters for a simple model of PN wound healing based on published observations on the kinetics of elementary processes of PN wound healing, [Buydash 2013]. Table 4.2-1 shows processes whose progress can be modeled as fronts that start from the nerve stumps and propagate towards the gap center. Examples of such processes are the migration of Schwann cells into the gap and the elongation of axons. Table 4.2-2 shows elementary processes that can be modeled as reactions where the presence of "reactants" leads to the formation of a "result". Examples of such processes include the synthesis of ECM and the radial growth of the contractile cell capsule.

Propagation Processes	**Substrate**	**Latency time [days]**	**Propagation velocity [mm/week]**	**References**
Exudate flow (EX)	-	0	137.7	Williams et al. 1985
Endoneurial cells (FB, SC)	Fibrin matrix or guide*	3	2.5	Williams et al. 1983, 1984; Schröder et al. 1993
Schwann cells (SC)	Fibrin matrix or guide*	3	1.3 2.2 2.5	Müller et al. 1987 Müller et al. 1987 Kemp et al. 2009
Vascularization	Fibrin matrix	5-10	2.5-3	Williams et al. 1983, 1984
Axon elongation	SC	6	1.5 (1w≤t<2 w) 6.0 (2w≤t<3 w)	Williams et al. 1983, 1984
Axon myelination	AX	11-12	3-4	Williams et al. 1983, 1984
Perineurial cell migration	Fibrin or conduit*	3	3.85-5	Williams et al. 1983; Schröder et al. 1993; Weis et al. 1994

Table 4.2-1: Kinetic parameters of elementary PN wound healing processes that can be modeled as propagating fronts.

Reaction Processes	**Reactant**	**Result**	**Reaction Lifetime [days]**	**References**
Coagulation	EX	Fibrin (FB)	<0.75	Schröder et al. 1993; Weis et al. 1994
Endoneurial matrix synthesis	SC, FB	Endoneurial ECM		
Myelination maturation	SC+AX	Myelin thickness	>>7	
Capsule radial growth	Capsule cells	Capsule thickness (t)	7	
Capsule matrix synthesis	Capsule cells	Capsule ECM	7	

Table 4.2-2: Kinetic parameters of elementary PN wound healing processes that can be modeled as a reaction.

4.2.5 Wound Contraction during Peripheral Nerve Wound Healing

The Role of Wound Contraction: Experimental Observations

Since the 1980's, several researchers have noticed the presence of multiple concentric layers of elongated cells and extracellular matrix (ECM) in the outer part of nerve regenerates. This tissue in this thesis is referred to as "the contractile cell capsule". Although some researchers recorded several features of this tissue (e.g. its formation rate along the axial axis [Williams et al. 1983, Schröder et al. 1993], and several protein markers expressed in capsule cells [Weis and Schröder 1994]), the role of the contractile cell capsule in the fate of PN wound healing was not clear. After the finding that the quality of the newly-formed tissue in skin wound healing is inversely related to wound contraction [Yannas et al. 1989], which is mediated by contractile myofibroblasts located mostly at wound edges [Troxel 1994; Tomasek et al. 2002; Desmoulière 2005], research revealed the expression of α smooth muscle actin (αSMA) in the contractile cell capsule in PN wound healing [Chamberlain et al. 1998a, 2000]. αSMA is the major marker of myofibroblasts (MFB, [Desmoulière et al. 2005]). αSMA-positive contractile cells are found in the edges of many kinds of injured organs, including the bed and edges of skin wounds [Murphy et al. 1990; Troxel 1994]. Despite the different geometry of skin and PN, in both cases of skin and PN wound healing MFB share several features that have mechanical significance (Figure 4.2.9):

i) αSMA-positive cells are extremely elongated and contain significant acto-myosin contractile apparatus [Hinz 2007], suggesting that they are capable of exerting significant forces. *In vitro* models suggest that these forces lie in the 10-100 nN/cell range [Harley et al. 2007].

ii) the long axes of neighboring MFB are aligned over significant areas [Murphy et al. 1990; Troxel 1994; Chamberlain et al. 1998a]. Since MFB stress fibers are parallel to their long axes [Murphy et al. 1990], this suggests that elementary forces applied by individual MFB add up into a resulting macroscopic force that is applied along a particular direction. In the case of skin wound healing, MFB axes lie on the wound bed plane, so that the resulting MFB force lies also on the wound bed plane [Murphy et al. 1990; Troxel 1994]. In the case of PN wound healing, MFB in the contractile capsule are aligned either circumferentially or along the nerve axis, so that the forces applied by capsule MFBs tend to generate hoop stress in the nerve regenerate (pressure cuff).

The presence of the contractile capsule in PN wound healing has been studied mostly at late time points (at least 6 weeks post-injury; at that time the final fate of the wound healing process is determined]) of the wound healing process [Chamberlain et al. 1998, 2000; Soller et al. 2012]. A recent study of the correlation between the contractile cell capsule and the outcome of PN wound healing focused on a challenging 15mm rat sciatic nerve wound treated with a collagen scaffold library of 5 members of progressively increased cross-linking (equivalently, progressively increasing *in vivo* degradation half-life). This study demonstrated that 9 weeks post-injury, the cross-linking of the collagen material modulated both the quality of the newly-formed nerve tissue, as well as the thickness of the contractile capsule. More importantly, it revealed a significant correlation between the quality of the newly-formed nerve tissue (described by number and size of axons in the gap of a 15mm gap) and the thickness of the observed contractile cell capsule at the gap midpoint [Soller et al. 2012]. These findings suggest that one possible way that biomaterials induce regeneration is by blocking the formation of the contractile cell capsule.

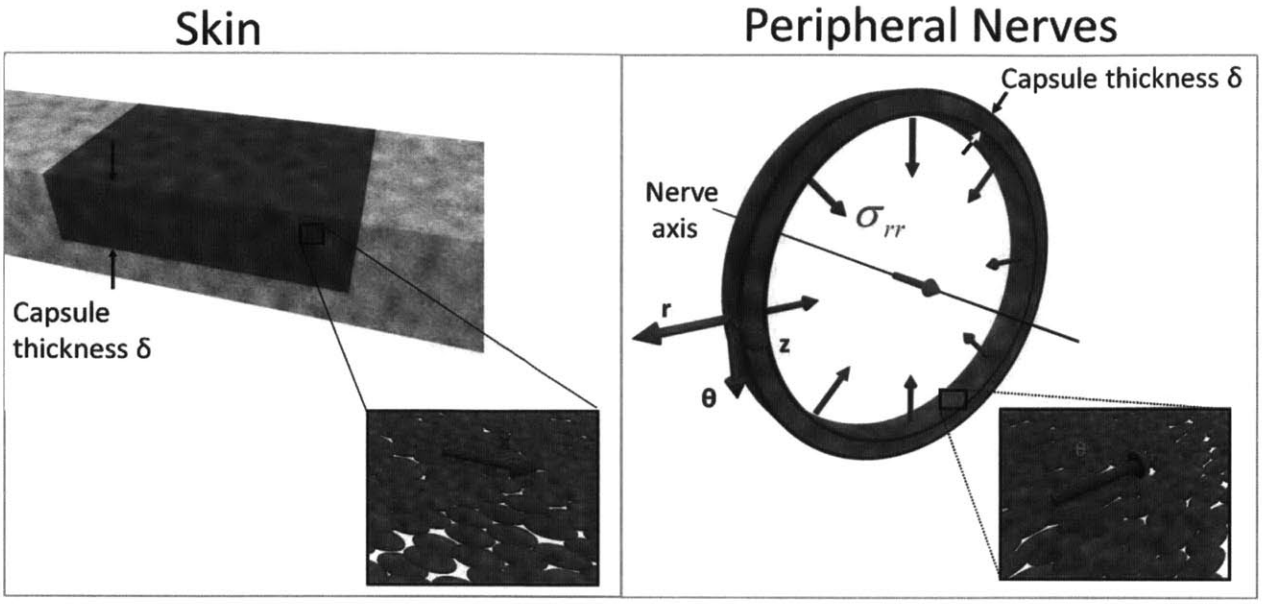


Figure 4.2.9: Geometry of the contractile capsule formed during skin and peripheral nerve wound healing. **Left**: In skin wound healing the capsule consists of contractile cells whose axis are parallel to the dermis plane. The macroscopic effect of their forces is shrinkage of the wound area (available for the formation of new tissue) and deformation of the surrounding tissue. **Right**: In PN wound healing, contractile cells are located in the outer surface of the nerve tissue and are oriented mostly circumferentially. The hoop stresses applied by capsule cells cause radial compression stress that compress the newly-formed tissue and are believed to impede synthesis of new tissue.

Wound contraction kinetics in peripheral nerves

At the moment there is no published study that reports *in vivo* measurements of contractile forces applied by MFB during wound healing in skin, PN or other organs. One way to infer the response of contractile force application, is to monitor its suggested outcomes, and specifically wound contraction, i.e. the deformation of the injured organ during wound healing. This section summarizes previous observation about how the diameter of the nerve trunk varies during nerve regeneration.

In PN wound healing, during the remodeling of the fibrin matrix and the formation of the nerve trunk, the diameter and structure of the nerve trunk varies over space and time. The nerve trunk has a tampered shape similar to the shape of its precursor fibrin cable (its diameter diminishes close to the gap midpoint). The diameter of the nerve stump is larger in the proximity of the proximal stump compared to the proximity of the distal stump. This diameter difference has been attributed to the presence of elongating axons from the proximal stump [Williams et al. 1983]. In transected nerves treated with silicone conduits the diameter of the nerve trunk decreases significantly between 1 and 2 weeks post-injury. This reduction is attributed to fibrin matrix degradation and correlates with the presence of Schwann cells [Williams et al. 1983, 1987; Williams 1987], which are known to degrades fibrin by expressing tissue plasminogen activator [Akassoglou and Strickland 2002b]. Experimental results from 10mm gaps treated with silicone tubes report that between 2 and 4 weeks post-injury the diameter of the nerve trunk remains practically constant [Lundborg et al. 1982a; Williams et al. 1983], between 4 and 8 weeks the diameter increases slowly [Kemp et al. 2009], and between 6 and 60 weeks the nerve trunk diameter remains constant [Chamberlain et al. 2000]. The increase of nerve trunk diameter between 2 weeks and 4 weeks is attributed to axon elongation and myelination

[Williams 1987]. This hypothesis is further supported by the observation that when axon elongation is prevented, the diameter of the nerve trunk decreases between 1 and 4 weeks [Zhao et al. 1992]. In a different study, the increase of nerve stump diameter observed between 8 weeks and 9 months post-injury is attributed to increased myelination and synthesis of endoneurial connective tissue [Jenq et al. 1985a].

These findings suggest that the critical time window where wound contraction takes place during PN wound healing starts when the fibrin matrix forms and ends approximately 2 weeks post-injury. The study presented in this Chapter aims to provide some information about the possible mediator of the contractile forces that cause contraction (the contractile cell capsule) and the effect of biomaterials.

4.2.6 Inducing Regeneration in Peripheral Nerves by Biomaterials

Biomaterial Treatments for Peripheral Nerve Regeneration

This study focuses on the use of biomaterials as grafts that can induce regeneration in severely injured peripheral nerves. Biomaterials can be made of synthetic polymers or natural biopolymers. Synthetic polymers usually do not provide ligands for cell adhesion, induce foreign body response and their degradation may result in toxic products. On the other hand they can be chemically modified in order to control several mechanical and chemical properties of the graft. Grafts based on natural biopolymers contain ligands that are recognized by cell adhesion receptors, and if fabricated properly they form analogs of the natural extracellular matrix (ECM) of the injured organ [Schmidt and Leach 2003; Tzeranis et al. under preparation]. Furthermore, biomaterials can be implanted empty or seeded with cell suspensions or small molecule solutions [Yannas 2001; Yannas et al. 2007; Deumens et al. 2010].

Biomaterial treatments for PN regeneration use grafts that attempt to change the wound healing dynamics in ways that favor the formation of new nerve tissue (Figure 4.2.10). The desirable wound healing outcome is that after transection axons survive, migrate through the injury gap, reach the distal nerve stump, and eventually reach the target muscle.

Two main kinds of biomaterial grafts (scaffolds) have been used in studies of peripheral nerve regeneration (PNR): conduits (tubes) and guides (fillings).

Conduits

The major component of biomaterial treatments for PNI is a cylindrical conduit that entubulates the two stumps of the injured PN, generating a volume between the two stumps. The key properties of nerve conduits are cross-linking (affecting degradation rate & stiffness), surface chemistry, and porous structure (permeability of cells and proteins to diffuse inside the conduit) [Harley et al. 2004; Soller et al. 2012]. The most successful conduits used so far are cell-permeable conduits made of lactic acid or collagen I [Yannas 2001; Yannas et al. 2007].

Fillings

In many treatments the volume inside the conduit is left empty after stump entubulation. Several studies suggest that the addition of a filling can improve regeneration. Four kinds of fillings have been applied: solutions, gels, insoluble biomaterials, and cells. A saline solution is by far the most popular filling. Several solutions of soluble regulators of cells (e.g. cytokines, small molecules, hormones) have been tested, their efficacy depends strongly on sequence, timing, dosage, and duration. Several gels of ECM molecules such as collagen, laminin and fibronectin

have been tested without particular success in improving the wound healing outcome. If gel density is high, data suggest that gels impede regeneration. Several guides have managed to significantly improve the wound healing outcome, particularly guides made of collagen, fibrin or polyamide that contain axially-oriented pores [Yannas 2001]. Cells can be added in addition to one of the previous three fillings (in suspension, inside a gel, or inside an insoluble matrix). The presence of Schwann cells has been shown to improve the outcome [Yannas et al. 2007].

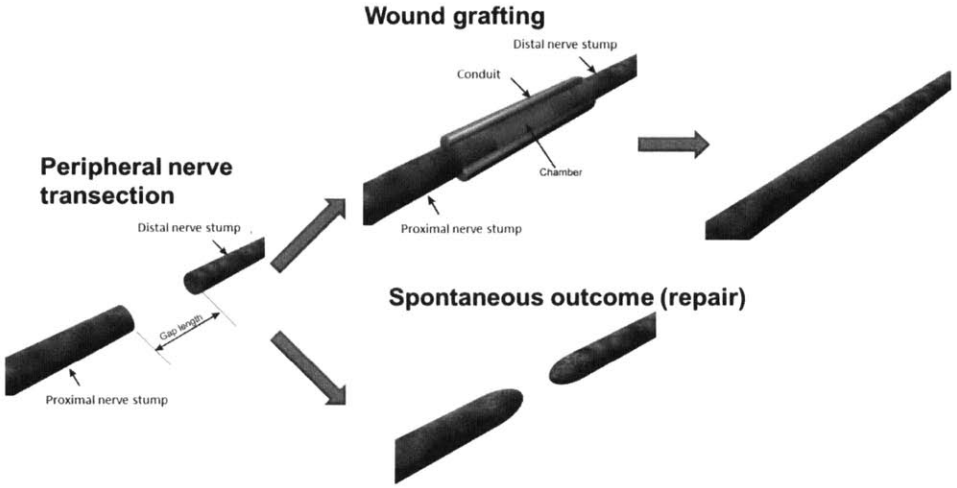


Figure 4.2.10: Biomaterial treatments aim to change the wound healing outcome and induce regeneration (formation of normal nerve tissue) in severely injured (transected) peripheral nerves that spontaneously do not regenerate.

Metrics for Evaluating the Performance of Induced Nerve Regeneration

The objective of peripheral nerve regeneration is the reconnection of transected axons with the target organ. The golden standard in evaluating PN wound healing outcome is to conduct electrophysiological studies at least 12-18 weeks post-transection [Chang et al. 1990]. An alternative easier way is to assay the quality of the newly-formed nerve tissue using morphometric assays (e.g. number and diameter of myelinated axons). Animal models of peripheral nerve regeneration suggest that morphometric assays obtained within at least 6 weeks post-injury correlate semi-quantitatively with electrophysiological measurements obtained as late as 60 weeks after injury [Arbuthnott et al. 1980; Chamberlain et al. 1998b; Yannas 2001]. Nevertheless, it is necessary to be aware of possible artifacts in evaluating the quality of nerve regeneration based on morphological features such as axon properties. Counting axons over-estimates the number of neurons that successfully regenerated into distal stump and reach target organ because each parent axon gives rise to multiple daughter sprouts [Fu and Gordon 1997; Gordon 2009].

A standardized criterion for comparing the efficacy of different treatments in inducing regeneration, is the critical axon elongation Lc metric, defined as the gap length where there is 50% probability of observing myelinated axons crossing the gap at sufficient time post-injury [Yannas et al. 2007].

4.2.7 Theories about the Origin of the "Regenerative Activity" of Collagen Scaffolds

The "regenerative activity" of a biomaterial is defined as its ability to change the wound healing outcome in ways that favor induced regeneration. In this thesis the terms "biologically active" and "biologically inactive" scaffolds refer to materials that have been shown to induce significant

or negligent improvement in the wound healing outcome compared to the spontaneous response. A more formal way to quantify the "regenerative ability" is the use if the metrics described in the previous section, for example the critical axon elongation metric utilized in PN regeneration [Yannas et al. 2007].

Several theories attempt to explain how the wound healing outcome after PN injury depends on the parameters of the injury model and applied treatment:

Neurotrophic theory

Neurotrophic theory focuses on the effect of soluble regulators on axon elongation, and emphasizes that endoneurial cell migration and axon elongation towards the distal stump is regulated by factors released from the distal stump. This theory builds upon the basic biology knowledge about the effects of growth factors on the guidance of axon growth cone [Fu and Gordon 1997; Gordon 2009]. Although this theory can explain the positive effect of entubulating the stumps with a conduit (entubulation favors the formation of a stronger cytokine field inside the gap), it cannot explain why the wound healing outcome depends strongly on the physicochemical properties of the biomaterial grafts, for example how materials expected to cause similar cytokine fields inside the gap result in very different wound healing outcome [Soller et al. 2012].

Contact guidance theory

Contact guidance theory emphasizes the positive effect of insoluble regulators on the migration of endoneurial cells inside the gap and the subsequent elongation of axons towards the distal stump. This theory emphasizes that axon elongation and endoneurial cell migration in the gap requires guidance by contact with an appropriate insoluble surface. Probably the most important example of such guidance is the elongation of axons inside into microtubes formed by Schwann cells (bands of Bungner). This theory can explain the regenerative biological activity of fillings added inside conduits: those fillings that accelerate the kinetics of the elementary PN wound healing processes (see Sections 4.2.4 and 4.2.5) such as Schwann cell migration, are the ones that result in improved wound healing outcome. Nevertheless, contact guidance_theory cannot explain sufficiently how nerve conduits affect the PN wound healing outcome, since the newly grown nerve tissue does not interact directly and it is not guided by the conduit.

Pressure cuff theory

While the previous theories focus on the formation of the nerve tissue in the core of the gap, pressure cuff theory emphasizes the adverse effects of the capsule of contractile cells that forms in the outer surface of the nerve trunk. This theory suggests that regeneration is impeded by the applied circumferential mechanical forces and the connective synthesized by contractile capsule cells. Pressure cuff theory suggests that treatments that block the formation of the contractile cell capsule should result in improved regeneration. Evidence for this inverse relationship was obtained recently 9 weeks post-injury [Soller et al. 2012]. This work attempts to provide additional evidence to support the pressure cuff theory by demonstrating that "active" biomaterials are the ones that block wound contraction during the first weeks of the wound healing process.

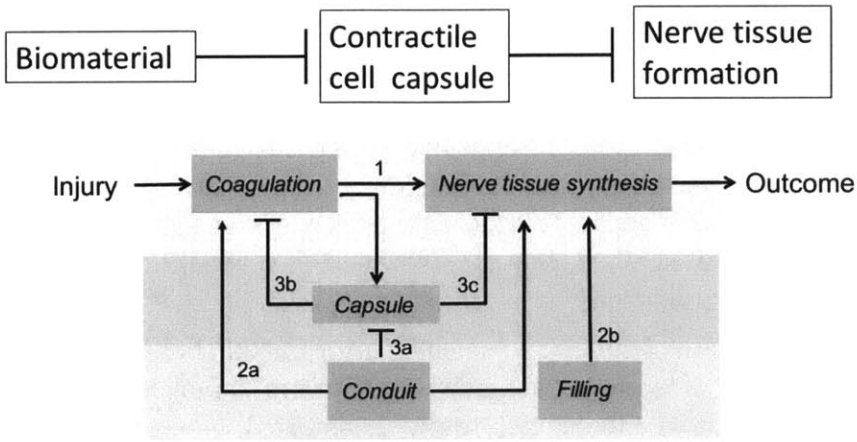


Figure 4.2.11: Theories that explain the effect of biomaterials on the wound healing outcome in injured peripheral nerves. **Top**: The pressure cuff theory suggests that biomaterials induce regeneration by preventing the formation of the contractile cell capsule, therefore preventing its adverse effects on the formation of new nerve tissue. **Bottom**: Overview of the three theories for the origin of the regenerative biological activity of collagen biomaterials. *Neurotrophic theory* suggests that conduits favor regeneration by favoring the formation of a cytokine field that favors nerve tissue formation (1). *Contact guidance theory* suggests that conduits favor regeneration by assisting in the formation of a fibrin clot that acts as a scaffold for endoneurial cell migration (2a). Contact guidance theory also suggests that biomaterial fillings favor regeneration by acting as a scaffold that accelerates endoneurial cell migration and axon elongation inside the gap (2b). *Pressure cuff theory* suggests that conduits favor regeneration by blocking the formation of the contractile cell capsule (3a), which could impede regeneration by mechanical deformation (3b, 3c) or steric hindrance due to the capsule stroma (3b, 3c) that prevent endoneurial migration and axon growth.

Discussion

At the moment none of the above theories can explain by itself all experimental observations. Pressure cuff theory could explain why the wound healing outcome depends on both the characteristics of PN injury (gap length, conduit pre-filling, nerve stump entubulation) and the properties of the biomaterial treatment (chemical composition, dimensions, pore size, cell permeability) [Soller et al. 2012]. Each theory focuses on one particular aspect of the PN wound healing. At the moment it is experimentally very hard to quantify *in vivo* and interrelate the key aspects of all three theories (i.e. how soluble regulators affect the formation of the capsule, how the capsule affects the formation of the nerve tissue, how the properties of the newly-formed nerve tissue affect the formation of the capsule).

4.2.8 Research Question

One proposed mechanism that explains how biomaterials induce regeneration in severely injured organs is the inhibition of the formation of a large-scale capsule of highly-contractile cells at the wound edges (Pressure Cuff Theory, Section 4.2.7). Previous experimental evidence from skin wound healing and PN wound healing suggests that indeed there exists a strong inverse relationship between the quality of the newly-formed tissue and the ability of a material to prevent or reduce the formation of this capsule [Yannas et al. 1989; Murphy et al. 1990; Chamberlain et al. 2000; Yannas 2001; Soller et al. 2012].

Specifically, the experiments of [Soller et al. 2012] provide strong evidence about the antagonistic relationship between regeneration and wound contraction, by identifying that 9 weeks post-injury the scaffolds that caused the formation of best nerve tissue were the ones that caused in the formation of minimal capsule thickness. While 9 weeks post-injury is a good time instant to evaluate the final outcome of wound healing, but may not be the best time point

to study the reasons that caused it. Several papers report that wound contraction in transected nerves treated with silicone materials peaks approximately 2 weeks post-injury [Williams et al. 1983, 1987; Williams 1987]. Furthermore, the thickness of the capsule at 9 weeks was measured without concern about the composition of the capsule (contractile cells, ECM). A recent study on the mechanics of wound contraction suggests that the strain imposed in the nerve tissue due to the forces applied by the contractile cells in the capsule depends on the mechanical properties of the scaffold, the capsule and the nerve [Buydash 2013]. Therefore, there exist still several open questions:

1. what is the mechanism of contractile cell capsule formation?
2. what is the effect of different collagen scaffolds on the formation and the composition of contractile cell capsule during the early phases of wound healing when wound contraction has been observed?
3. which are the molecular clues provided by scaffolds that affect the formation of the contractile cell capsule?
4. what are the mechanisms by which the contractile cell capsule can affect the formation of the new nerve tissue?

This Chapter focuses on question 2. It studies the wound healing response in challenging PN injuries grafted with the best (device "D") and the worse (device "E") scaffolds identified in [Soller et al. 2012] at those early time points (1 and 2 weeks post-injury) where mechanics and prior observation suggest that the nerve is more vulnerable to contraction. Chapter 5 provides some evidence on question 3.

4.3 Methods

4.3.1 Animal model and Sample Preparation

All animal experiments were designed and conducted by Dr. Soller [Soller 2011; Soller et al. 2012]. The experimental setup follows the established Lundborg animal model for studying the effects of conduits and fillings in PN wound healing [Lundborg et al. 1982]. Briefly, the sciatic nerve in adult Lewis rats is fully transected, and the two resulting nerve stumps are separated by a 15±1mm gap and inserted into a conduit (Figure 4.3.1). The initial study considered the effects of five kinds of collagen conduits. This study presents results for two of them (Section 4.3.2). Nerve regenerates were harvested after sacrificing animals by CO_2 inhalation 1, 2 or 9 weeks post-transection (this study focuses on 1 and 2 week data). The harvested sciatic nerve tissue was rinsed in saline, fixed *en block* in 4% paraformaldehyde for 8 hours at 4°C, treated overnight in 30% sucrose, embedded in OCT medium, snap-frozen in liquid nitrogen and stored at -20°C until sectioning. Frozen blocks were sectioned at 6 μm thickness on a microtome (Keck Imaging Facility, Whitehead Institute, MIT) and stored at -20°C until staining and imaging. All experimental protocols followed NIH guidelines (NIH Publication No. 85-23 Rev. 1985).

The chosen L=15 mm gap length is significantly larger than the L_{cr} = 9.7±1.8 mm critical gap length in rat nerves treated with silicone conduits [Yannas 2001; Yannas et al. 2007]. Such large gap was chosen to provide unambiguous evidence of induced, rather than spontaneous, reconnection between the two stumps.

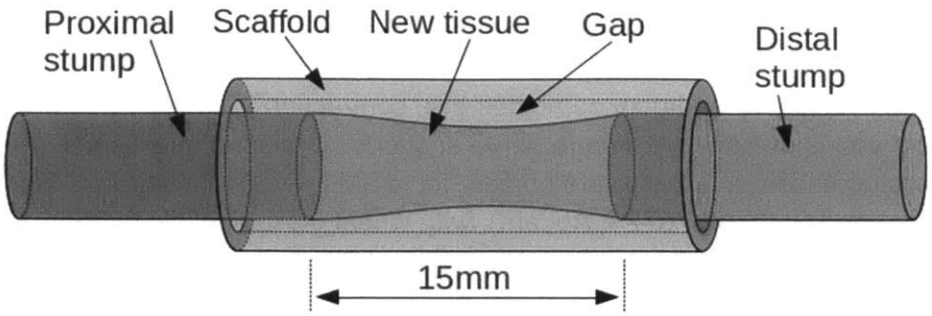


Figure 4.3.1: Schematic of the Lundborg animal model for studying the effects of collagen biomaterials on induced PN regeneration.

4.3.2 Scaffold Library

In the original study, transected rat PN were treated with five different porous collagen conduits. All scaffolds are fabricated by freeze-drying a 5% w/v suspension of micro-fibrilar collagen I in 0.05M acetic acid [Soller 2011]. After freeze-drying, the *in vivo* degradation half-life of the five scaffolds is modulated by appropriate crosslinking treatment. The result was a library of five scaffolds of 80μm mean pore diameter that differed from each other on their *in vivo* degradation half-life (or equivalently, the intermolecular cross-linking) [Harley et al. 2004; Soller 2011].

This study considers just two of these scaffolds. Specifically, scaffold "D" is cross-linked by dehydro-thermal treatment (DHT) [Yannas and Tobolsky 1967] resulting in *in vivo* degradation half-life of approximately 3 weeks. Scaffold "E" is cross-linked using the cross-linking agents EDAC (1-Ethyl-3-(3-dimethylaminopropyl)carbodiimide) and NHS (N-hydroxysuccinimide), resulting in *in vivo* degradation half-life of much larger than 9 weeks [Harley et al. 2004].

4.3.3 Fluorescent Staining

This study utilizes four kinds of fluorescence staining protocols. The details of the protocols are displayed in Table 4.3.3.

The "contraction" staining protocol focuses on imaging the distribution of cell contractility apparatus in the nerve trunk using wide-field fluorescence microscopy. This staining protocol consists of an antibody (A2547, Sigma Aldrich, St. Louis, MI) that binds α smooth muscle actin (αSMA), and the DAPI (D1306, Life Technologies) counterstains. αSMA is the major marker of myofibroblasts (MFB). In peripheral nerves αSMA is also expressed in pericytes around blood vessels. DAPI binds nucleic acids and stains cell nuclei.

The "cell content" staining protocol focuses on identifying the distribution of cells the nerve regenerate using wide-field fluorescence microscopy or spectral multi-photon microscopy. This staining protocol consists of wheat germ agglutinin (WGA) Alexa Fluor 488 conjugate (W11261, Life Technologies), Alexa Fluor 532 phalloidin (A22282, Life Technologies) and DAPI. WGA is a lectin (sugar-binding protein) that binds to N-acetylglucosamine (GlcNAcβ1-4GlcNAcβ1-4GlcNAc) and N-acetylneuraminic acid (Neu5Ac, sialic acid) residues. WGA binds to multiple glycoproteins found on cell membranes and the extracellular matrix of tissues. In this study, strong WGA signal was observed in the basement membrane of myelinated Schwann cells. WGA is expected to bind to heparin sulfate proteoglycans that contain GlcNAc found at neuromuscular junctions and at Schwann cell basement membranes [Carbonetto 1984]. WGA

was also observed in the cytoplasm of all cells. Phalloidin binds and stains F-actin in actin filaments. Large actin expression is found in contractile cells, pericytes and perineurial cells.

The "complete nerve staining" protocol combines the "contraction", "content", and "nerve tissue" staining protocols and is designed to image nerve sections by spectral multi-photon microscopy. This staining protocol combines the antibodies for αSMA (A2547, Sigma Aldrich), NF200 (ab72996, Abcam) and S100 (ab76729, Abcam), with wheat germ agglutinin (WGA) Alexa Fluor 488 conjugate (W11261, Life Technologies), Alexa Fluor 532 phalloidin, and DAPI.

	Contraction Protocol	**Cell content Protocol**	**Complete Protocol**
αSMA primary ab. (A2547)	1:400	-	1:600 (mouse)
αSMA secondary ab.	1:400 anti-mouse Alexa Fluor 488	-	1:1000 anti-mouse Alexa Fluor 350
NF200 primary ab. (ab72996)	-	-	1:1000
NF200 secondary ab.	-	-	1:400 anti-chicken tritc
S100 primary ab. (ab76729)	-	-	1:500
S100 secondary ab.	-	-	1:300 anti-rabbit alexa fluor 350
WGA - Alexa Fluor 488	-	1.67 µg/ml	0.1 µg/ml
Phalloidin - Alexa Fluor 532	-	100	200
DAPI	1E-2 µg/ml	1E-2 µg/ml	5E-3 µg/ml

Table 4.3-1: Information about the staining protocols utilized to study the *ex vivo* nerve samples.

Sectioned animal nerves are transferred out of the -20°C and equilibrated at room temperature for 30 min, washed twice with TBST for 10 min, blocked by 80 µl block reagent that contains 0.3% Triton X100 (X0909 Dako) at room temperature for 1h, treated with 60 µl primary antibody solution in antibody diluent (S0809, Dako) overnight at 4°C, washed three times with 100 µl TBS for 10 min at room temperature, treated with 60 µl secondary antibody solution in antibody diluent for 2h at room temperature, treated with 50µl counterstain solution in TBS for 30min at room temperature, washed three times with 100 µl TBS for 10 min at room temperature, add 5 µl mounting medium (17985-10 Fluorogel), add the coverslip and seal using a nail hardener.

4.3.4 Spectral Multi-photon Microscopy and Image Processing

Stained nerve samples were imaged in the spectral multiphoton microscope described in Section 2.3. The acquired images were processed using the image processing pipeline described in Chapter 2. The pipeline provides a segmentation of image pixels into various classes (myofibroblasts, banded collagen fibers, scaffold collagen, axons, schwann cells, cell nuclei) described by the class map of Fig. 2.6.4, and an estimation of the emission rates for the emission sources present in each pixel.

4.3.5 Fluorescence Imaging and Image Processing

The distribution of the fluorescently-labeled markers shown in Table 4.3-1 in *ex vivo* nerve samples was imaged by wide-field fluorescence microscopy. Stained nerve sections were

imaged in an Olympus IX81 fluorescence microscope (Olympus) at the Van Vliet Lab (Department of Material science and engineering, MIT) equipped with a mercury lamp (Prior Scientific, UK), an ORCA CCD detector (Hamamatsu, Japan), and four filter sets: bright-field imaging filter set, DAPI filter set, alexa fluor 488 filter set, and alexa fluor 549 filter set. Low magnification (4× or 10×) images were acquired in order to study their expression and abundance over the whole area of the nerve-scaffold cross-sections. Image acquisition was implemented using the microscope's control software to first focus, and then acquire the image.

For each image and channel (red, green, blue), the background noise of the detector, and the background non-specific staining signal was calculated using a custom-made GUI written in MATLAB. The sensor background is constant over all samples. The background non-specific staining varies slightly over the various tissues (nerve tissue, capsule, scaffold). For each magnification and staining protocol, common threshold values $[t_L, t_U]$ were chosen and applied to images from all nerve samples. The low-threshold value t_L is chosen so that it is larger than the background non-specific staining signals observed over all samples. The upper-threshold value t_U is chosen equal to the largest "actual" signal value observed over all images of identical magnification, channel, and staining protocol. This assures no signal saturation and maximum dynamic-range data. Images were segmented manually into the three major compartments (nerve tissue, capsule, scaffold). The mean thickness of the contractile cell capsule was measured manually at 4 locations in the image located at the 4 opposite quadrants of the image. For each experimental group (transected nerves treated with scaffolds "D" or "E"), time of animal sacrifice, and position in the wound (0, 2, or 4 mm away from the point of transection) at least 3 samples from distinct animals were quantified.

4.3.6 Statistical Analysis

One-factor analysis of variance (ANOVA) tests were performed to determine the effect of the scaffold graft ("D" or "E") on the mean thickness capsule at a each time point (1 or 2 weeks) and location in the wound (0 and 2 mm). Statistical significance was accepted for $p < 0.05$.

4.4 Results

4.4.1 Exploiting Nonlinear Spectroscopy to Distinguish Banded Collagen

In the *ex vivo* rat sciatic nerve samples considered in this study, collagen is present either inside the collagen scaffold, or in the ECM of the nerve tissue. The collagen found in ECM is found in a semi-crystaline state that causes collagen fibrils to possess a characteristic 67-nm axial periodicity (see Chapter 5). Such semi-crystaline collagen is called "banded". On the other hand, scaffolds are made by freeze-drying a collagen I suspension in acetic acid. Due to the acid treatment, collagen fibrils swell, which causes them to lose their semi-crystalline structure [Sylvester et al. 1989]. The loss of crystalinity during fabrication is the reason that in TEM images of scaffolds only a small fraction of the scaffold appears banded (Figure 5.4.10, Chapter 5). This difference in the solid state of collagen in ECM versus in the scaffold is exploited in order to distinguish them optically. This can be done because while both banded and non-banded collagen are fluorescent, only banded-collagen can generate second harmonic emission [Chen et al. 2012]. For the same reason, while gelatin (denatured collagen) is auto-fluorescent,

it cannot emit second harmonic. Figure 4.4.1 shows spectroscopic results that quantify these claims. Figure 4.4.1b shows the spectral signatures of pure microfibrillar collagen I (the raw material utilized to make the scaffolds) and the collagen scaffold itself. These data were obtained by imaging raw collagen I fibers and collagen scaffolds respectively, binning the data from all pixels corresponding to collagen, and then normalizing. The results show that the second harmonic peak of raw collagen is much larger compared to the peak of the scaffold. Figure 4.4.1a shows the ratio of the second harmonic signal to the fluorescence signal. This large difference between the ratio of second harmonic emission to fluorescence emission is exploited in distinguishing the collagen belonging to the scaffold from the collagen that has been synthesized by rat cells.

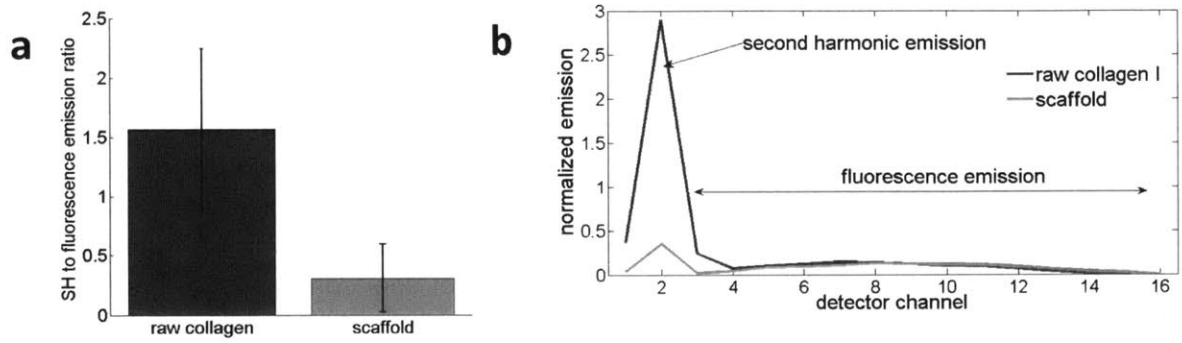


Figure 4.4.1: Exploiting nonlinear optical spectroscopy to distinguish banded collagen. **a**: Second harmonic emission per fluorescence emission in raw microfibrilar collagen I (banded) and in collagen scaffolds (swollen). Mean ± standard deviation. **b**: Spectral signatures of raw microfibrilar collagen I (banded) and in collagen scaffolds (swollen).

4.4.2 Formation of the Contractile Capsule in PN Wound Healing

This section provides imaging results that describe the morphology of peripheral nerves during the early phase (1-2 weeks post-injury) of wound healing after nerve transection. Nerves have been treated with two kinds of collagen scaffolds, named "D" and "E". Images of cross sections of *ex vivo* rat sciatic nerves treated with collagen conduits are acquired by either wide-field fluorescence microscopy, or by spectral multi-photon microscopy (Chapter 2).

Figure 4.4.2 shows a schematic that describes the basic morphology of the cross section images shown in the following figures. The newly-synthesized nerve tissue (N) is surrounded by a region consisting of cells and ECM, called the "contractile cell capsule" (C). The capsule has distinct morphology and composition compared to the nerve tissue. 9 weeks post-injury this capsule consists of circumferentially oriented contractile cells and collagen. The nerve trunk (nerve tissue and capsule) is located in the inner lumen of the conduit (scaffold, S) that was used to entubulate the two nerve stumps after transections. The scaffold is itself surrounded by another capsule of contractile cells, called then external capsule (EC). The observed morphology of the nerve trunk during wound healing is quite different compared to the normal morphology of a cross section of a PN, which consists of nerve fascicles surrounded by the dense collagenous stroma of epineurium.

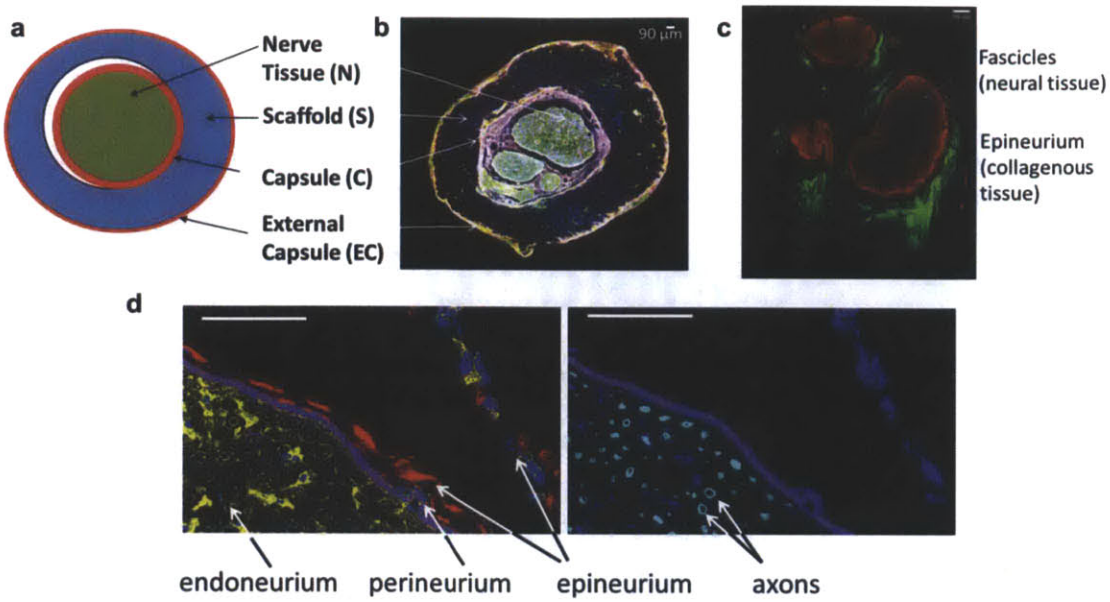


Figure 4.4.2: Morphology of cross sections of peripheral nerves during wound healing. **a**: Schematic of the four main parts. The new nerve tissue is surrounded by a capsule of contractile cells in the inner lumen of the scaffold. The scaffold itself is surrounded by the external capsule. **b**: the four main parts of the PN highlighted in a sample image of a PN during wound healing. bar = 90µm. **c**: cross section of a normal peripheral nerve. bar = 90µm. **d**: high resolution image of a cross section of a normal peripheral nerve. Red: collagen. Blue: cell nuclei. Purple: phalloidin. Yellow: WGA. Cyan: axons. bars = 50µm.

Figure 4.4.3 to Figure 4.4.6 show wide-field fluorescence images of *ex vivo* rat sciatic nerve cross sections treated with either scaffold "D" or scaffold "E", harvested 1 or 2 weeks post-injury, 0 or 2 mm away from the proximal transection point (the location where the PN was transected at the stump connected to the spinal cord). Left columns provide images of sections that have been immuno-labeled for αSMA, the major marker of myofibroblast differentiation. This labeling identifies cells expressing αSMA, which are either pericytes (contractile cells that wrap around blood vessels in peripheral nerves) or myofibroblasts. Right columns provide images of sections that have been stained by fluorescently-labeled phalloidin (a stain that binds f-actin) and wheat germ agglutinin (a lectin that binds specific kinds of polysaccharides). The phalloidin stain labels cells that express extensive acto-myosin apparatus, and therefore can apply significant contractile forces. The WGA stain is less specific, it stains glycoproteins found in cell membranes, in the Golgi, and several proteoglycans found in the basement membranes of Schwann cells.

Figure 4.4.7and Figure 4.4.8 show images of *ex vivo* rat sciatic nerve sections immuno-labeled for αSMA, and focus on the expression of αSMA in the contractile cell capsule that surrounds the nerve tissue. These images are higher magnification images of the αSMA results shown in the first column of Figure 4.4.3 to Figure 4.4.6.

Figure 4.4.9 to Figure 4.4.14 show representative high-resolution spectral multi-photon images of the contractile capsules that form around the newly-formed nerve tissue. Apart from the phalloidin (purple), WGA (yellow) and DAPI signals (blue-green), spectral multi-photon imaging can resolve simultaneously the weak fluorescence emission of collagen in the scaffold (green), and the second-harmonic emission of banded collagen (red) synthesized by rat cells during

tissue remodeling. Figure 4.4.14 highlights the locus where capsule cells come in contact with the surrounding scaffolds.

Figure 4.4.15 and Table 4.4-1 show measurements for the mean thickness of the capsule that surrounds the new tissue for the two animal groups considered in this study (grafted with scaffold D or E), at 1 or 2 weeks post-injury, and at 4 distinct locations in the wound.

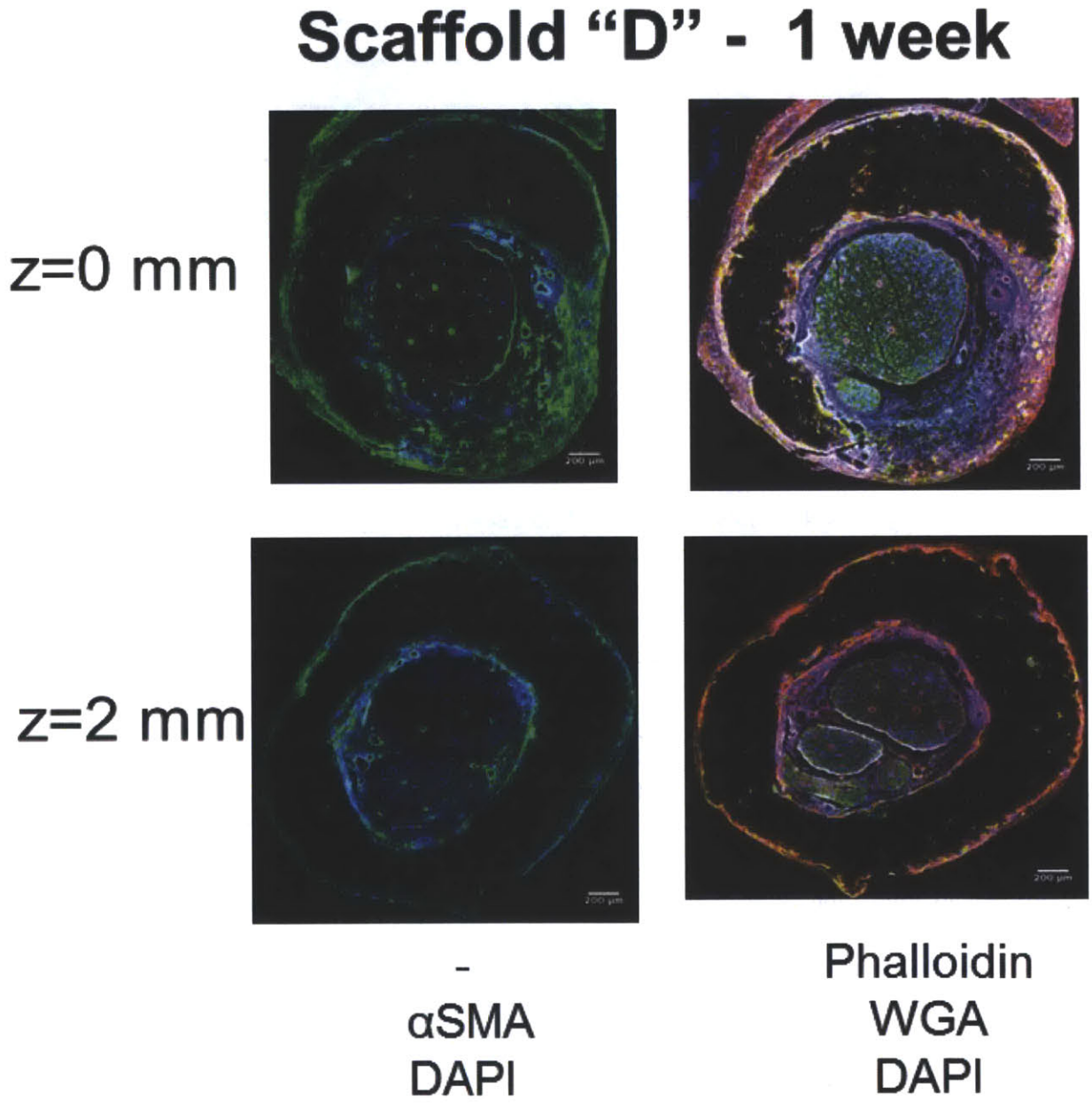


Figure 4.4.3: Response of contractile cell phenotypes in transected peripheral nerves treated with collagen scaffolds "D", 1 week post-injury at 0 and 2mm away from the proximal transection point. **Left**: Blue: DAPI. Green: alexa fluor 488 conjugated secondary antibody for mouse anti-αSMA primary antibody. **Right**: Blue: DAPI. Green: alexa fluor 488 conjugated wheat germ agglutinin, red: phalloidin. Bars: 200 μm.

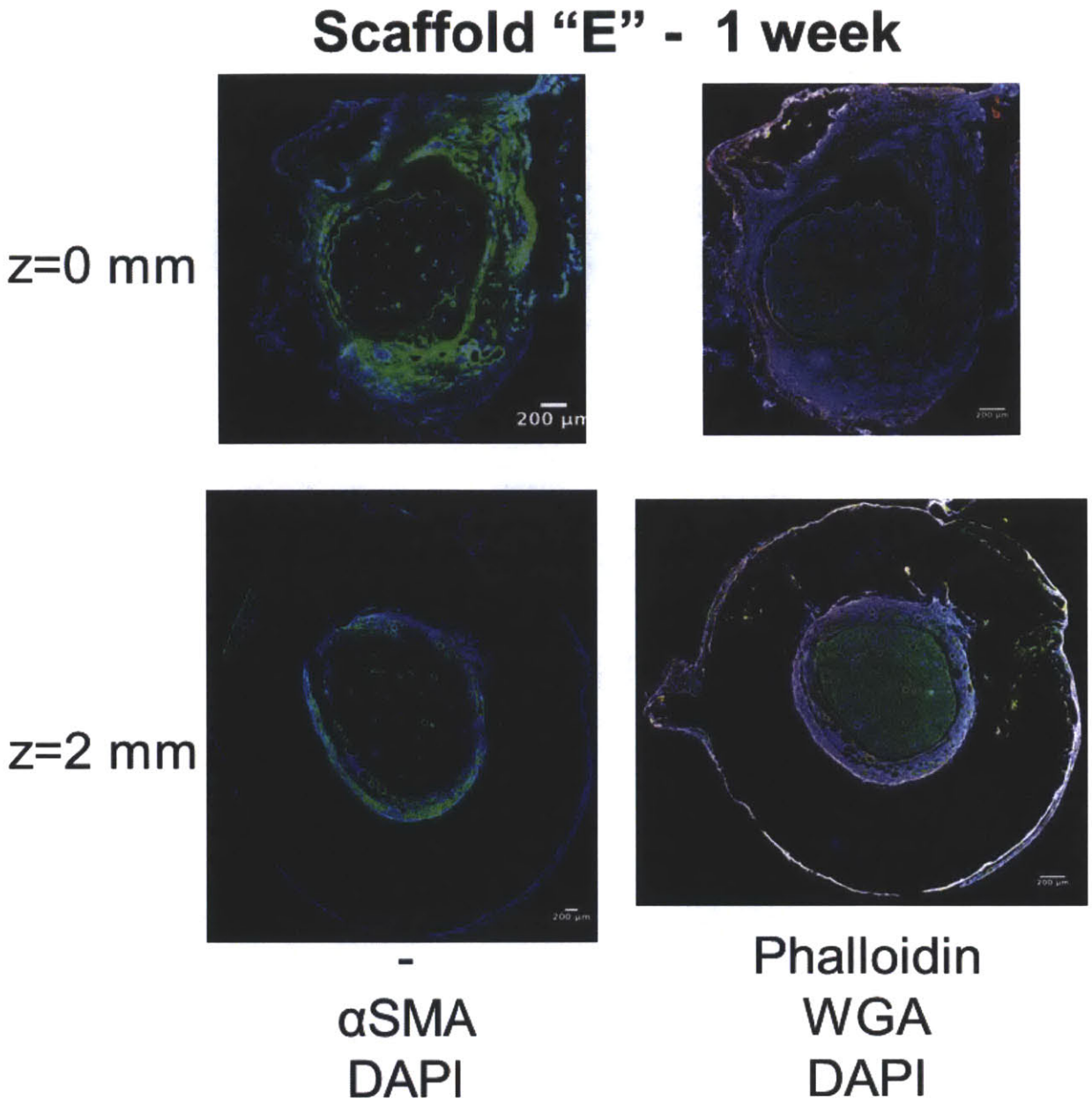


Figure 4.4.4: Response of contractile cell phenotypes in transected peripheral nerves treated with collagen scaffolds "E", 1 week post-injury at 0 and 2mm away from the proximal transection point. **Left**: Blue: DAPI. Green: alexa fluor 488 conjugated secondary antibody for mouse anti-αSMA primary antibody. **Right**: Blue: DAPI. Green: alexa fluor 488 conjugated wheat germ agglutinin, red: phalloidin. Bars: 200 µm.

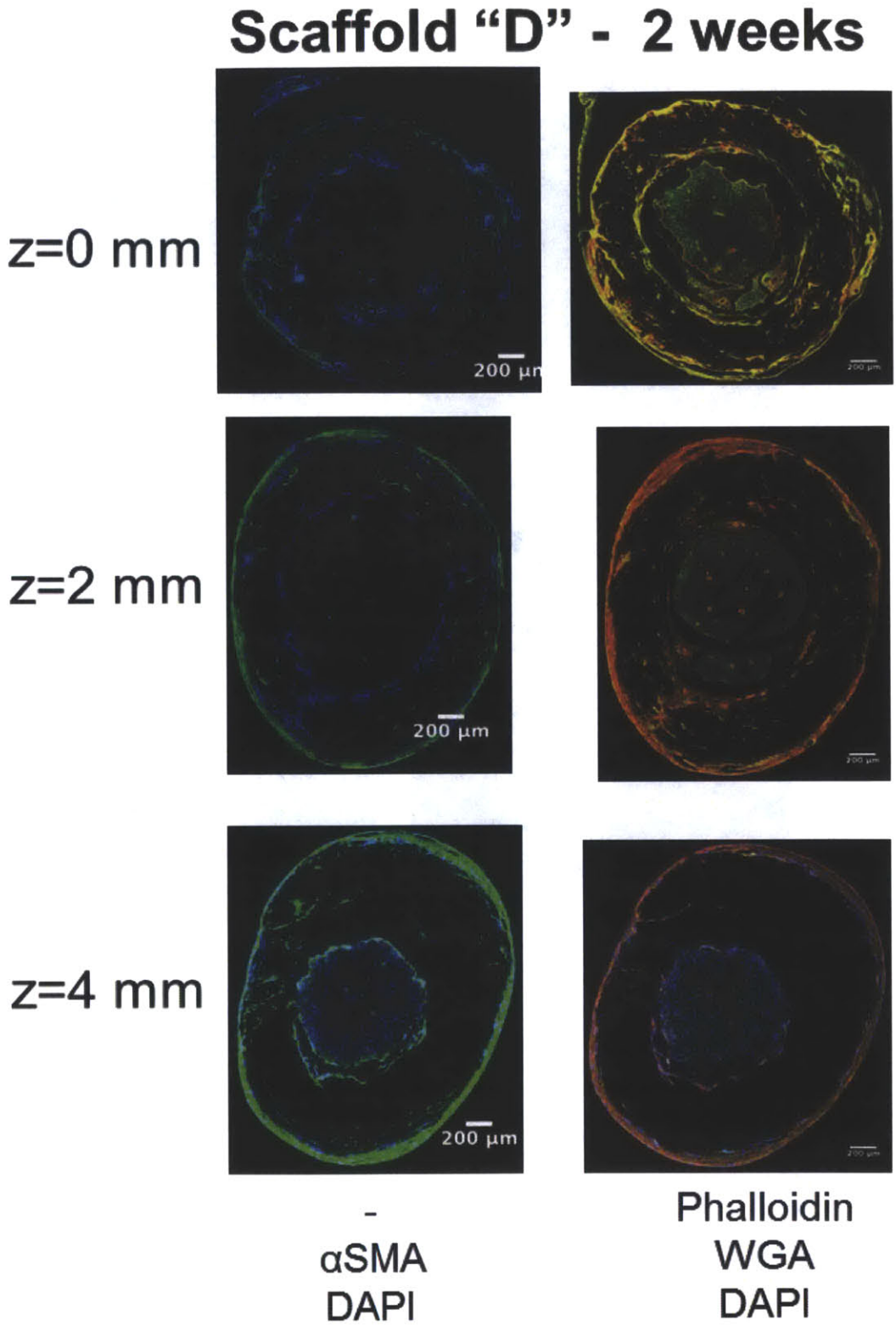


Figure 4.4.5: Response of contractile cell phenotypes in transected peripheral nerves treated with collagen scaffolds "D", 2 weeks post-injury at 0, 2 and 4 mm away from the proximal transection point. **Left**: Blue: DAPI. Green: alexa fluor 488 conjugated secondary antibody for mouse anti-αSMA primary antibody. **Right**: Blue: DAPI. alexa fluor 488 conjugated wheat germ agglutinin, red: phalloidin.

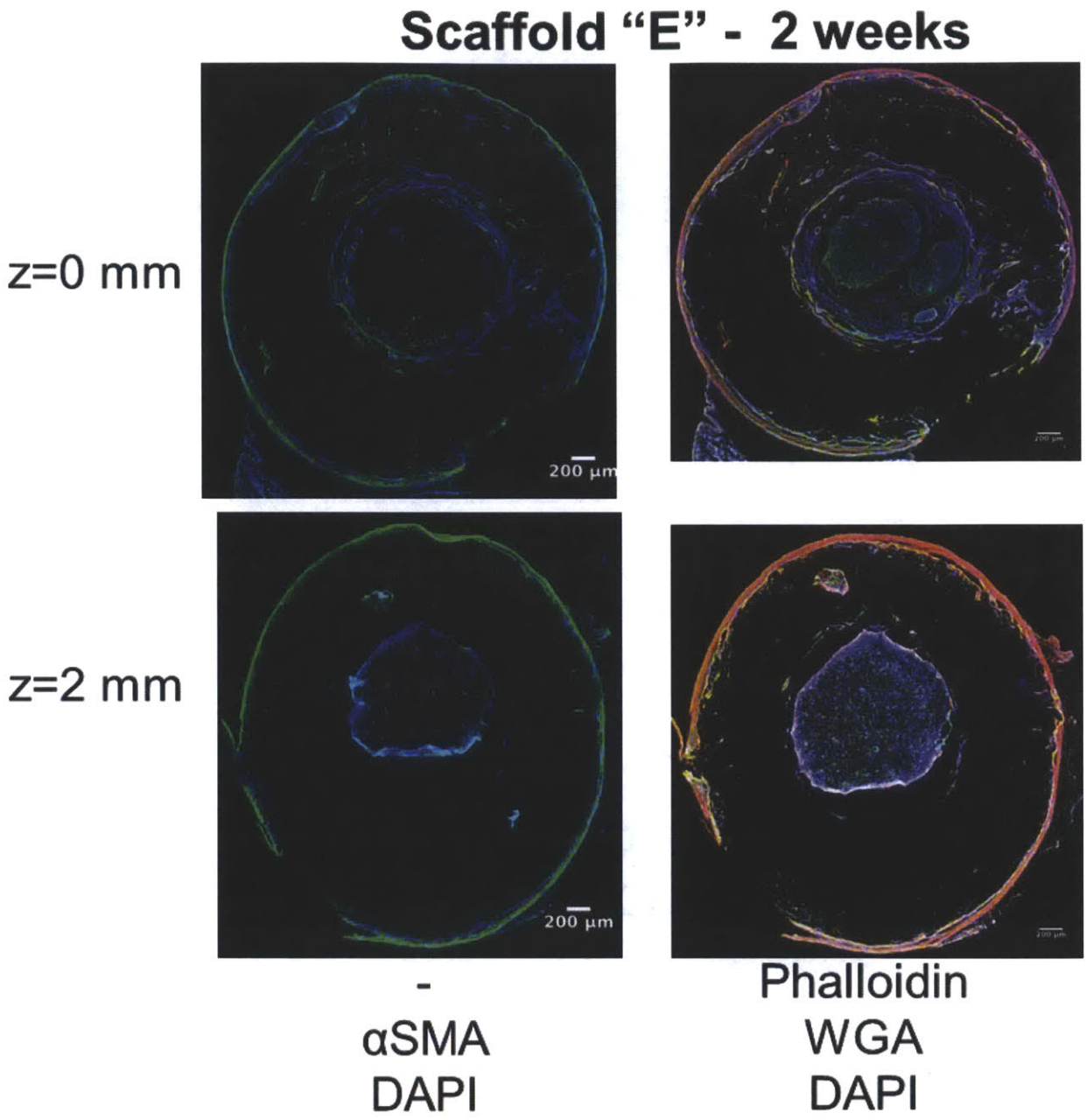


Figure 4.4.6: Response of contractile cell phenotypes in transected peripheral nerves treated with collagen scaffolds "E", 2 weeks post-injury at 0 and 2 mm away from the proximal transection point. **Left**: Blue: DAPI. Green: alexa fluor 488 conjugated secondary antibody for mouse anti-αSMA primary antibody. **Right**: Blue: DAPI. Green: alexa fluor 488 conjugated wheat germ agglutinin, red: phalloidin. Bars: 200 µm.

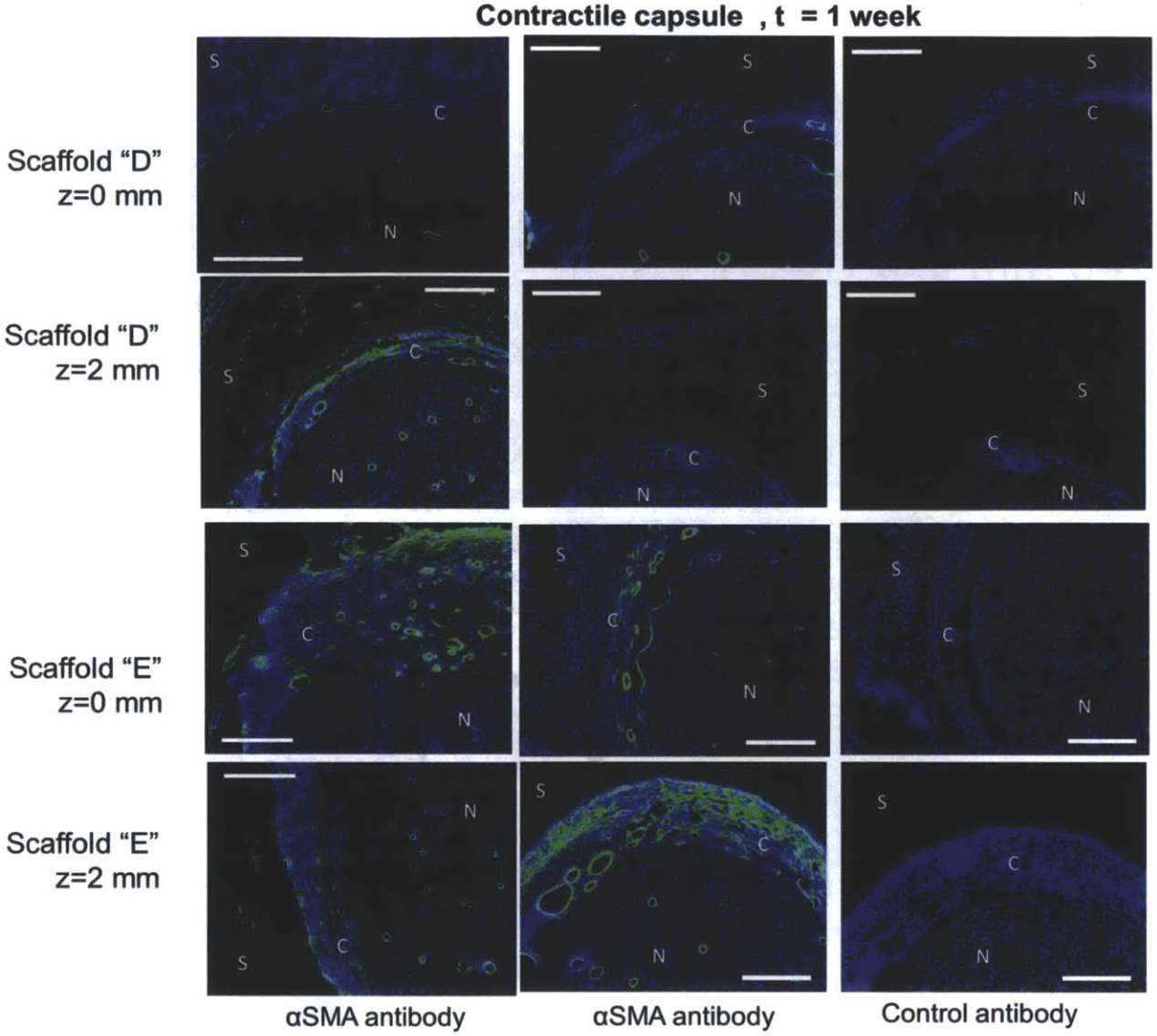


Figure 4.4.7: Immunofluoresce staining for αSMA in the contractile cell capsule at various distances z from the reference point, 1 week post-injury. The second column corresponds to zoom-in images from the same samples as Figure 4.4.3 and Figure 4.4.4. The third column shows results for the negative antibody control in neighboring sections to the results shown in the second column. Green: Alexa Fluor 488 conjugated anti-mouse secondary antibody against the primary mouse anti-αSMA antibody. Blue: DAPI. Images of the first column are provided from [Buydash 2013] (identical animal experiment, different nerve samples). All bars are 200 µm.

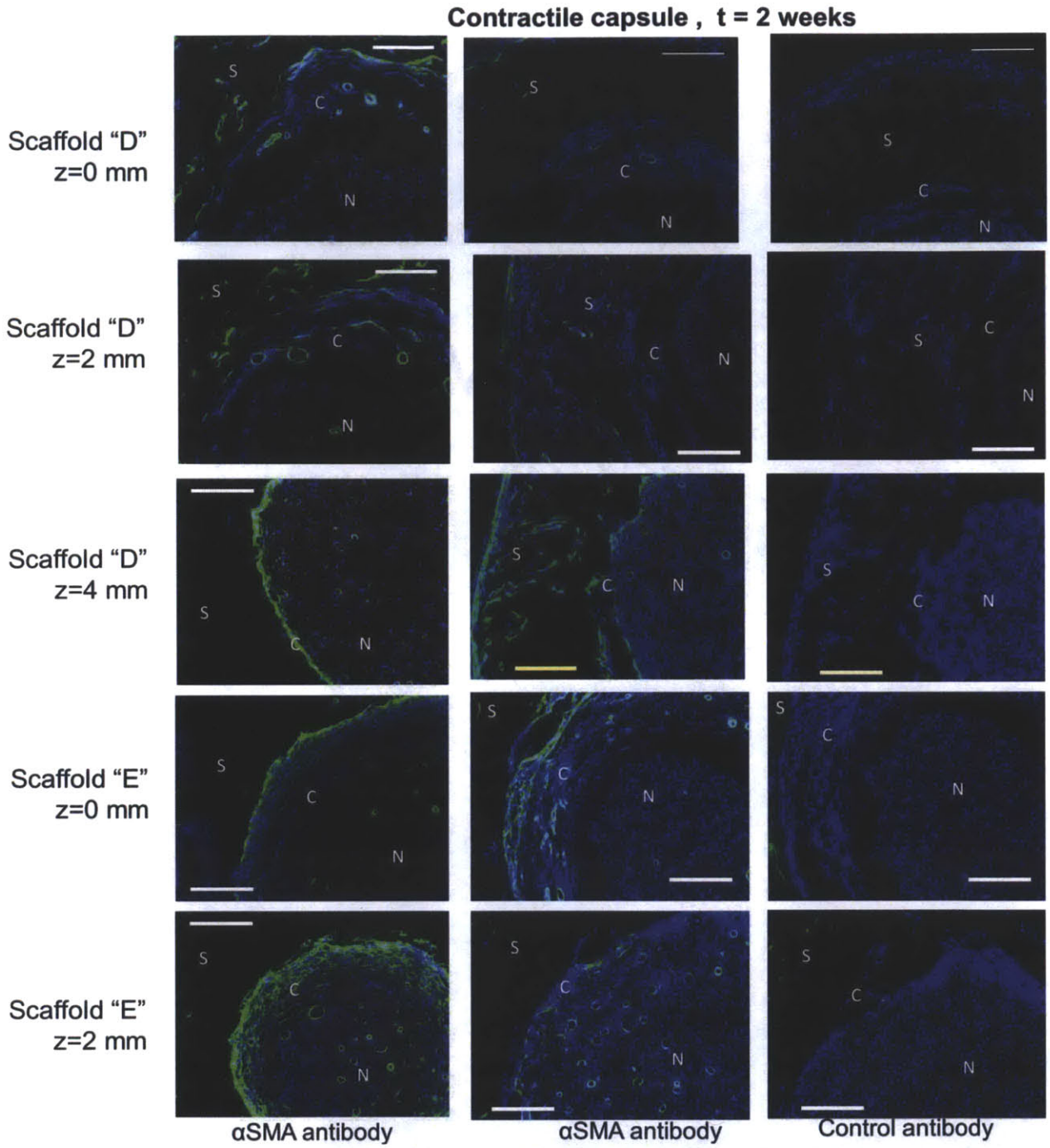


Figure 4.4.8: Immunofluoresce staining for αSMA in the contractile cell capsule at various distances z from the reference point, 2 weeks post-injury. The second column corresponds to zoom-in images from the same samples as Figure 4.4.5 and Figure 4.4.6. The third column shows results for the negative antibody control in neighboring sections to the results shown in the second column. Green: Alexa Fluor 488 conjugated anti-mouse secondary antibody against the primary mouse anti-αSMA antibody. Blue: DAPI. Images of the first column are provided from [Buydash 2013] (identical animal experiment, different nerve samples). All bars are 200 μm.

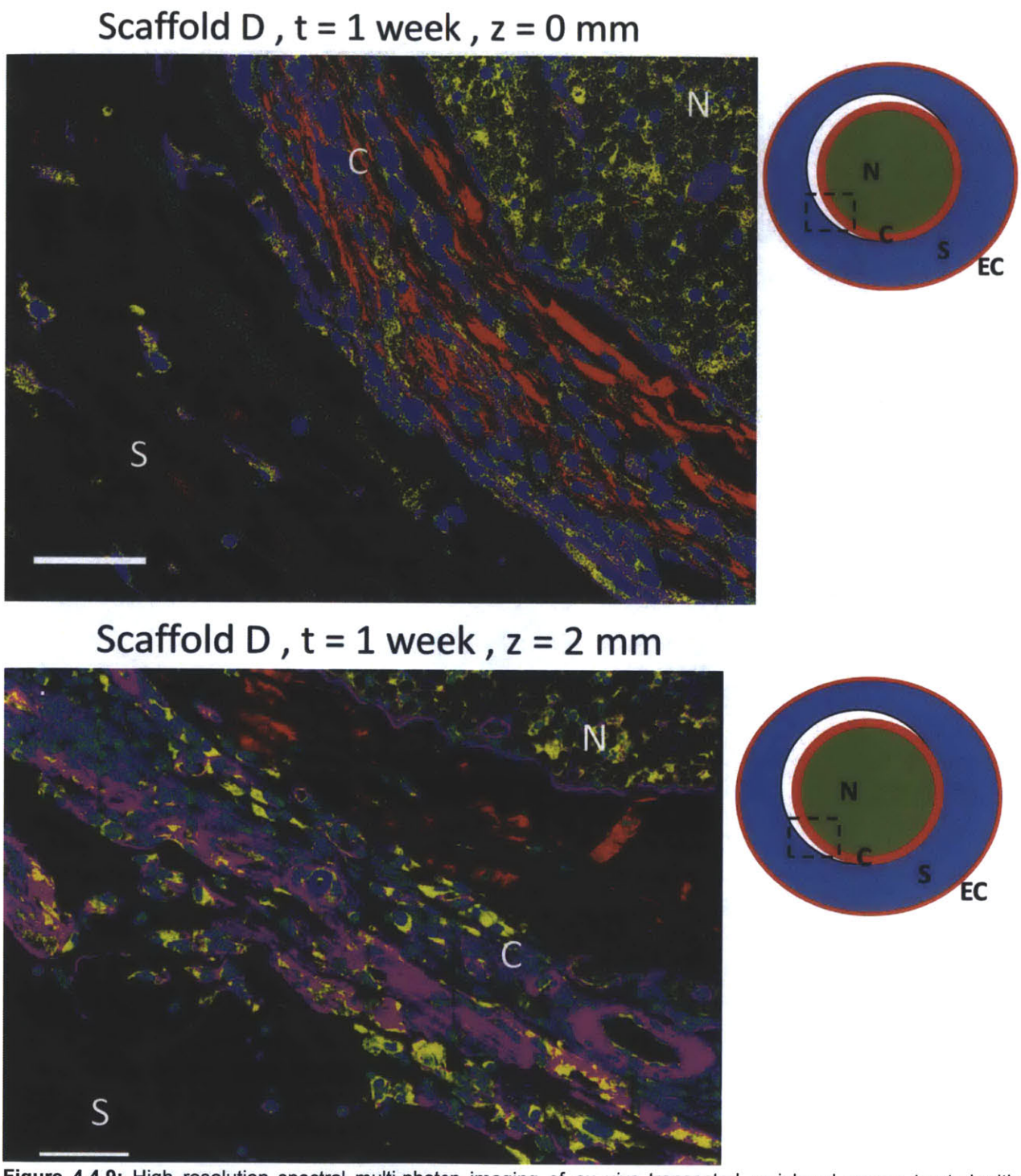


Figure 4.4.9: High resolution spectral multi-photon imaging of *ex vivo* transected peripheral nerves treated with scaffold D, 1 week post injury at z = 0 (top) and z = 2 mm (bottom) away from the proximal point of transection. Red: banded collagen (collagen synthesized by cells), green: non-banded collagen (scaffold), yellow: glycoproteins (WGA staining), purple: phalloidin, blue-green: cell nuclei. N: nerve tissue. C: Capsule. S: Scaffold. Bars = 50 µm.

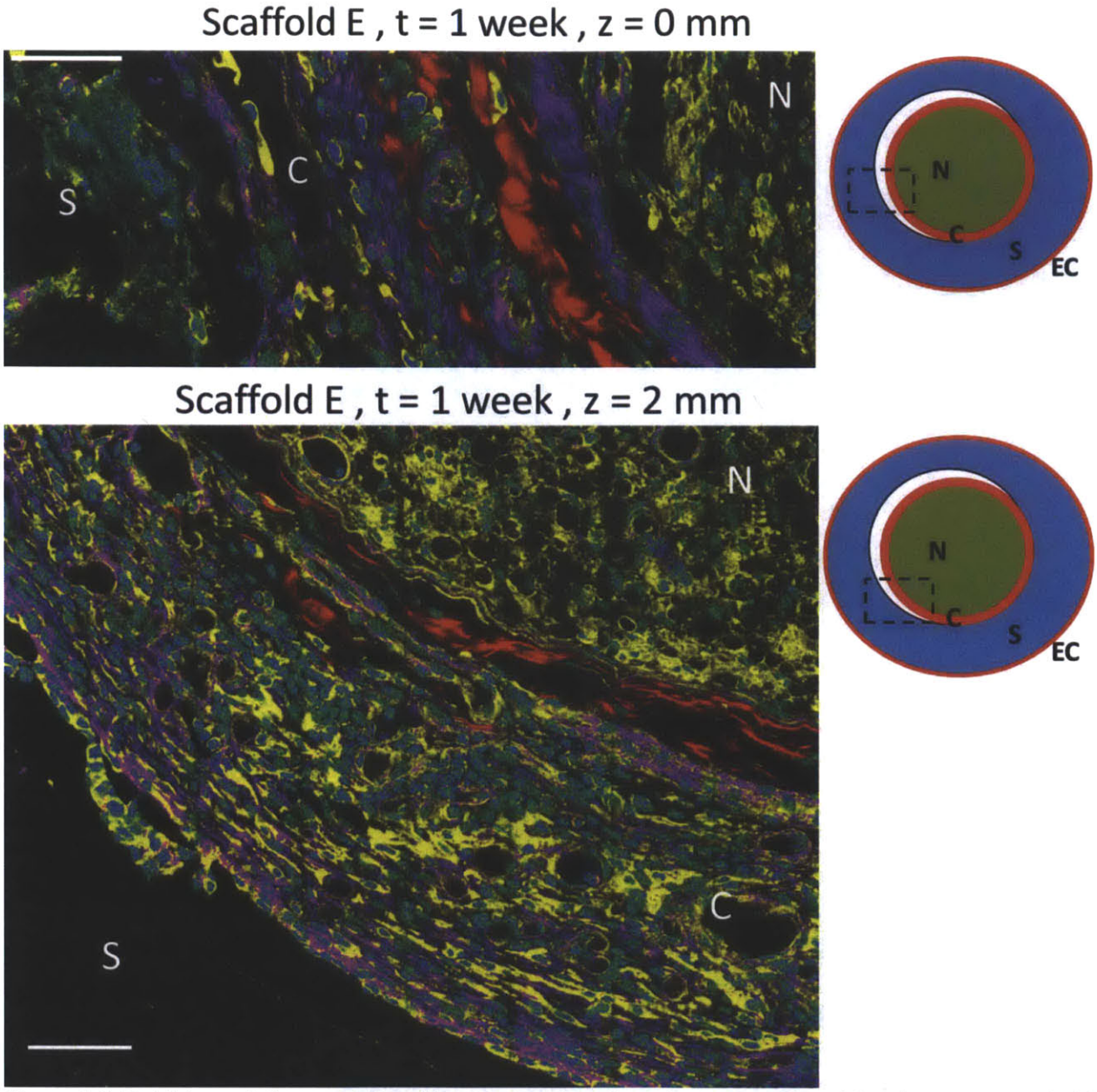


Figure 4.4.10: High resolution spectral multi-photon imaging of *ex vivo* transected peripheral nerves treated with scaffold E, 1 week post injury at z = 0 (top) and z = 2 mm (bottom) away from the proximal point of transection. Red: banded collagen (collagen synthesized by cells), green: non-banded collagen (scaffold), yellow: glycoproteins (WGA staining), purple: phalloidin, blue-green: cell nuclei. N: nerve tissue. C: Capsule. S: Scaffold. Bars = 50 μm.

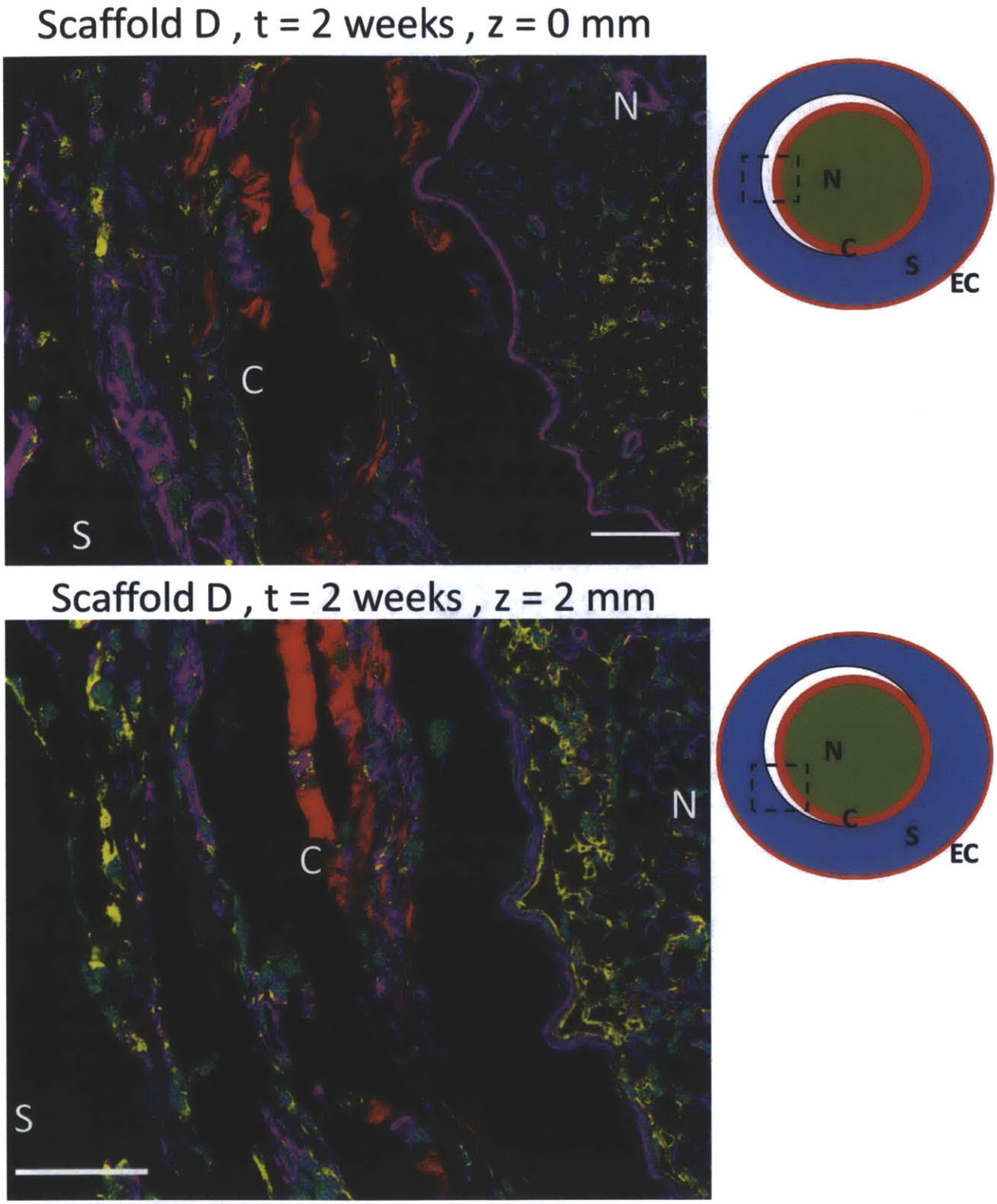


Figure 4.4.11: High resolution spectral multi-photon imaging of *ex vivo* transected peripheral nerves treated with scaffold D, 2 weeks post injury at z = 0 (top) and z = 2 mm (bottom) away from the proximal point of transection. Red: banded collagen (collagen synthesized by cells), green: non-banded collagen (scaffold), yellow: glycoproteins (WGA staining), purple: phalloidin, blue-green: cell nuclei. N: nerve tissue. C: Capsule. S: Scaffold. Bars = 50 µm.

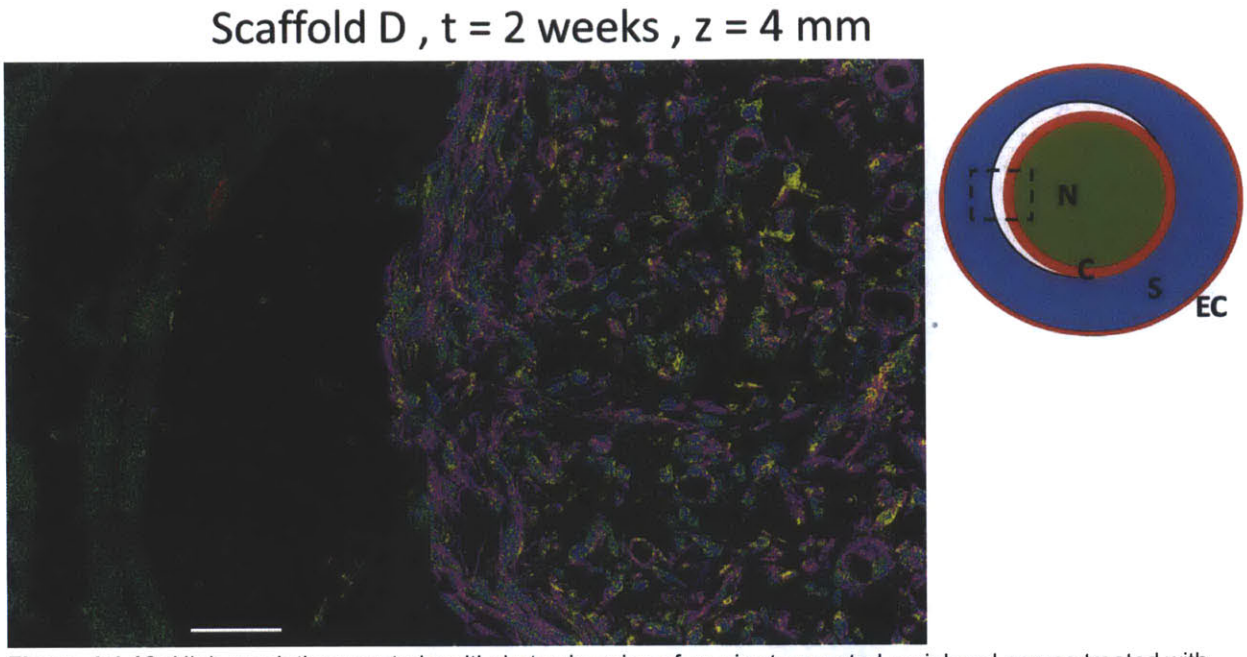
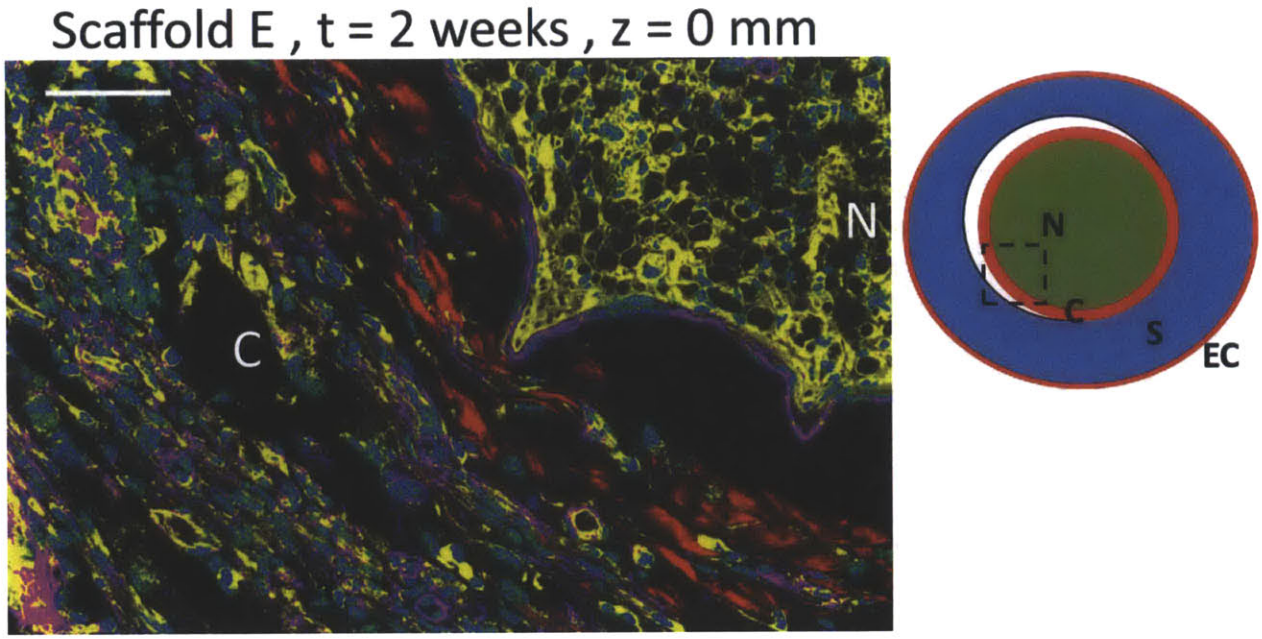


Figure 4.4.12: High resolution spectral multi-photon imaging of *ex vivo* transected peripheral nerves treated with scaffold D, 2 weeks post injury at z = 4 mm away from the proximal point of transection. Red: banded collagen (collagen synthesized by cells), green: non-banded collagen (scaffold), yellow: glycoproteins (WGA staining), purple: phalloidin, blue-green: cell nuclei. N: nerve tissue. C: Capsule. S: Scaffold. Bar = 50 μm.



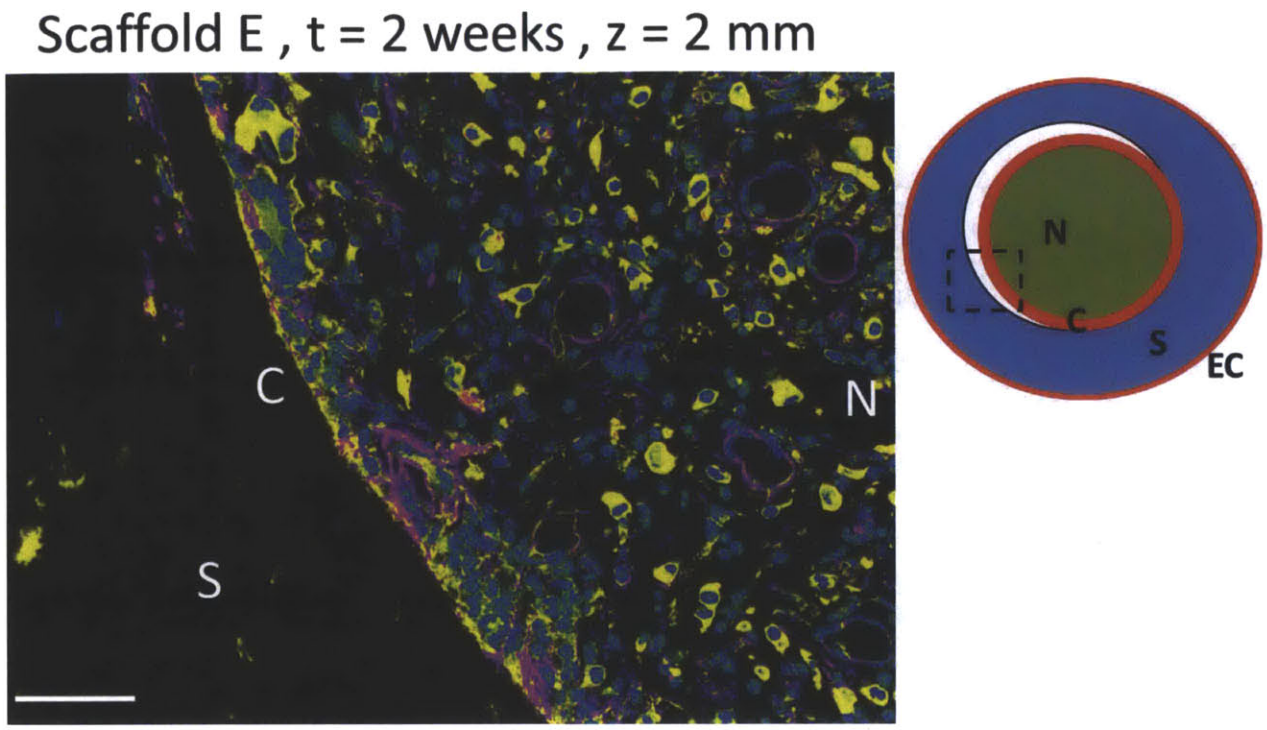


Figure 4.4.13: High resolution spectral multi-photon imaging of *ex vivo* transected peripheral nerves treated with scaffold E, 2 weeks post injury at z = 0 (top) and z = 2 mm (bottom) away from the proximal point of transection. Red: banded collagen (collagen synthesized by cells), green: non-banded collagen (scaffold), yellow: glycoproteins (WGA staining), purple: phalloidin, blue-green: cell nuclei. N: nerve tissue. C: Capsule. S: Scaffold. Bars = 50 µm.

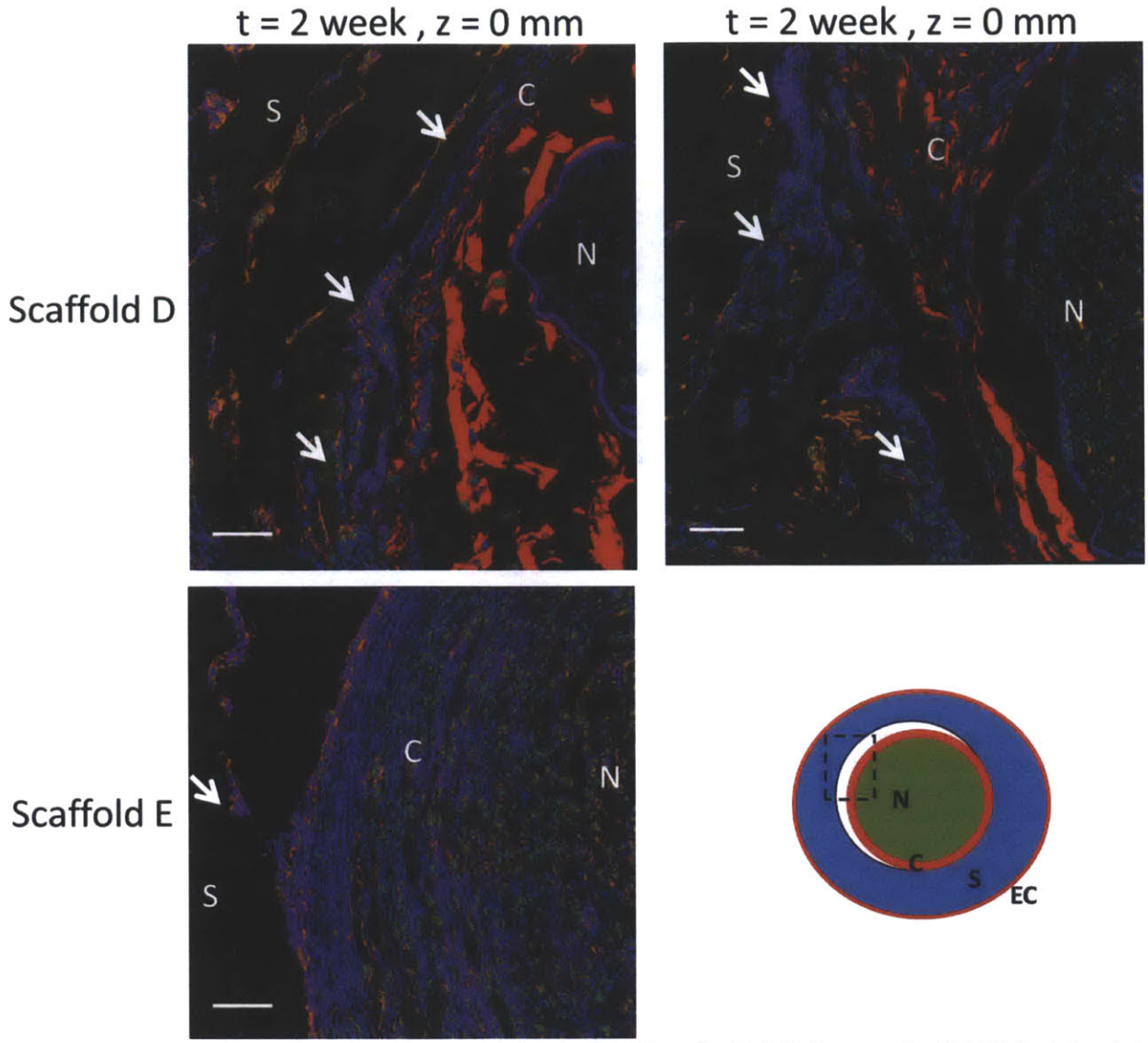


Figure 4.4.14: Adhesion pattern of capsule cells to the scaffold. **Top**: Scaffold D. **Bottom**: Scaffold E. Red: banded collagen (collagen synthesized by cells), green: non-banded collagen (scaffold), purple: phalloidin, blue-green: cell nuclei. Orange: αSMA (in the capsule) or schwann cells (nerve tissue). N: nerve tissue. C: Capsule. S: Scaffold. Bars = 50 μm. Arrows highlight regions of cell-matrix adhesion that survived the sample cutting procedure. Bars = 50 μm.

Capsule thickness [μm]		**t = 1 week**		**t = 2 weeks**		
		z = 0 mm	**z = 2 mm**	**z = 0 mm**	**z = 2 mm**	**z = 4 mm**
Scaffold D	μ	77.7	87.4	60.2	53.7	44.1
	σ	18.6	26.7	8.8	14.0	11.9
	N	3	3	3	3	3
	SE	10.8	15.4	5.1	8.1	6.9
Scaffold E	μ	133.8	176.9	118.3	79.1	No nerve tissue observed
	σ	29.6	97.8	29.4	23.1	
	N	3	4	4	3	
	SE	17.1	48.9	14.7	13.4	

Table 4.4-1: Statistics of mean thickness of the contractile capsule (in μm) measured around the nerve tissue as a function of the scaffold graft, the time of sacrifice, and the position z away from the point o transection. The data are plotted in Figure 4.4.15. μ: average, σ: standard deviation, N: sample size, SE: standard error of mean

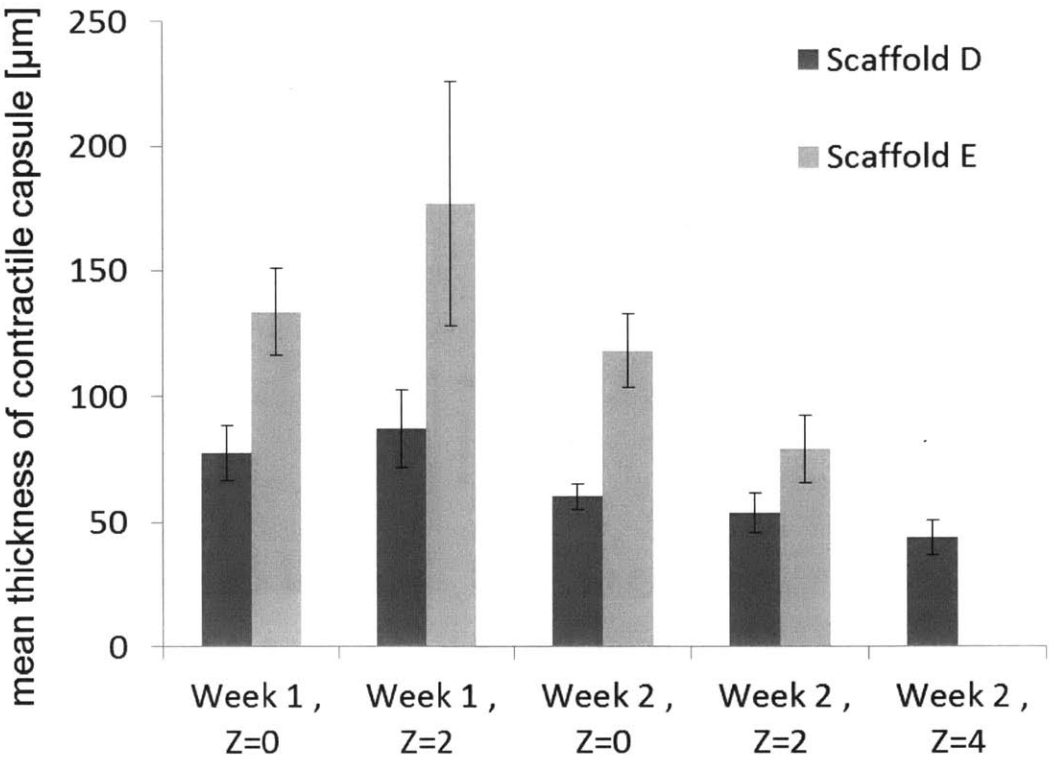


Figure 4.4.15: Mean thickness of the contractile cell capsule at various locations and time instances as a function of the scaffold used to treat the transected nerve. P-values are provided for one-way ANOVA analysis on the effect of the scaffold for each experimental condition. The p-value for pairwise one-side ANOVA tests that quantifies the effect of different locations equals p=0.633 (device D, t = 1 week), p=0.367 (device D, t = 2 weeks), p=0.577 (device D, t = 1 week), and p=0.372 (device E, t = 2 weeks). Multiple comparison tests using the Tukey-Kramer approximation suggest that the data sets of the five sets points for group "D" are not statistically different. Similarly, the four data sets for group "E" are not not statistically different at p=5% significance.

4.5 Discussion

4.5.1 Collagen Scaffolds Regulate the Formation of the Capsule

Figure 4.4.3 to Figure 4.4.13 show that already 1 and 2 weeks post injury the wound healing response of transected peripheral nerves treated with scaffolds D or E are significantly different.

Specifically, a non-neural tissue ("capsule") rich in contractile cells forms in the outer part of the nerve tissue. This capsule is imaged at 0, 2 and 4 mm away from the transection point. Tissue at 4mm was observed only in the animal group D, 2 weeks post-injury.

One week post-injury, the mean thickness of the capsule in animals treated with scaffold D is approximately 80μm. Statistics suggest that the capsule thickness at z=0 and z=2 mm are not statistically different. The mean thickness of the capsule formed in animal group D is less than the mean thickness of the capsule formed in animal group E (approximately 130 μm) (Figure 4.4.15), however the difference is statistically significant only at z=0. Two weeks post-injury, in both animal groups (D, E) the capsule measurements are slightly less compared to the measurements of 1 week, however this difference is not always statistically significant. Still, the measurements of mean capsule formed in animal group D are smaller than the ones of group E, however this is not always statistically significant. The thickness of the scaffold that forms within1 week post-injury could be related to the inner diameter of the lumen. Specifically, the

difference between the inner diameter of the scaffold (approximately 1.5 mm [Soller 2011]) and the outer diameter of the healthy rat sciatic nerve (normally 1 mm but nerves swell after transection [Harley et al. 2004]). The size of the gap generated between the nerve and the material may affect the formation of the cell-rich tissue that forms between them early in wound healing.

The capsule contains a large number of cells (evident in Figure 4.4.3 (z=2mm), Figure 4.4.4 (z=2mm), Figure 4.4.9 and Figure 4.4.10) that are already preferably circumferentially oriented. The vast majority of cells stain for WGA, many of them stain brightly for actin. The number of circumferential cells that stain positively for αSMA (excluding blood vessels) seems to be significantly more in samples grafted with scaffold "E" compared to samples grafted with scaffold "D" (Figure 4.4.7 and Figure 4.4.8). Capsule is abundant in blood vessels (circular structures that stain for phalloidin and αSMA) of approximately 10 – 40 μm diameter. Many blood vessels are observed in the outer part of the newly formed nerve tissue (Figure 4.4.12, Figure 4.4.13), in agreement with observations in nerve treated with silicone tubes [Schröder et al. 1993; Zochodne and Nguyen 1997].

At z=0 and z=2 mm away from the point of transection, the capsule is separated from neural tissue by a thin layer of highly elongated cells that stain strongly for phalloidin and a few layers of banded collagen. These structures resemble the structures of perineurium and epineurium found in normal PN tissue (Figure 4.4.2d) and possibly they are parts of the original PN tissue. It seems therefore reasonable to suggest that z=0 is not the point of transection any more.

The structure of the capsule is not homogenous. In the animal group D, close to point of transection the capsule consists of concentric layers of cell and ECM. The inner-most layers consist of cells attached to large circumferentially oriented collagen fibers (Figure 4.4.11 to Figure 4.4.11), possibly the collagen fibers of the old epineurium. The few outer-most layers consist of circumferentially oriented cells that stain for αSMA and are in contact with scaffold D. A similar structure of circumferentially oriented layers of cells and collagen appears near the nerve transection point in animal group E (Figure 4.4.6 top, Figure 4.4.13 top). In this case the thickness of the capsule is larger and the αSMA signal is present in a larger number of layers (still localized in the outer part of the capsule).

Two weeks post-injury, the morphology of the nerve and the capsule seems very different at locations away from the point of transection. In both animal groups, the nerve trunk looks much less organized and consists of a large number of blood vessels, single WGA-positive cells (in contrast to the honey-comb like staining pattern of WGA close to the transection point), axons, and Schwann cells (not shown here), see Figure 4.4.12 and Figure 4.4.13 bottom. At these locations there is no perineurial and epineurial-like structures and there is no clear distinction between the nerve tissue and the capsule. However, in the outer surface of the nerve trunk there is a thin region (2-3 cell layers) where cell density is higher compared to the core of the nerve trunk. These cells stain positively for actin and αSMA (Figure 4.4.7, Figure 4.4.8). It is possible that this circumferential structure of cells is the precursor of the thick capsules observed 9 weeks post-injury [Chamberlain et al. 2000; Soller et al. 2012]. The data shown in Figure 4.4.15 suggest the mean capsule thickness observed at z=4mm in group D is almost statistically significantly smaller the mean capsule thickness observed at z=2mm in group E ($p=0.08$, these two points are compared because the capsule surrounds a newly-formed tissue

and not the remaining of the old stump), however this is kind of misleading because the capsule in one of the three samples of the 2-week z=2 set of the E group is much larger compared to the rest. As shown in Figure 4.4.11 and Figure 4.4.13bottom, in the early steps of capsule formation around the nerve tissue the capsules in both groups seem similar in size. This suggests that a factor that may regulate their final growth it their interaction with the surrounding scaffold.

4.5.2 Cell-matrix Interactions and the Outcome PN Wound Healing

Figure 4.4.14 highlights difference in cell-matrix interactions between capsule cells and the scaffold, close to the point of transection. Nerve cross sections are prepared by fixing, embedding in OCT and cryo-sectioning the nerve samples after they are harvested. Unfortunately sectioning induces sectioning artifacts. These artifacts are tears, which form in the locus of least mechanical strength. For example, in the top left image of Figure 4.4.14 a tear separates collagen fibers located in the epineurium. In the top right image of Figure 4.4.14 a tear separates collagen perineurium from the epineurium. In none of these images (and in the vast majority of the cross sections of the D group) did the tear take place at the location where the capsule connects to the scaffold (highlighted using arrows). On the contrary, in the vast majority of the cross sections of the E group, sectioning induced tears exactly at the scaffold-capsule interface, just like the one shown in the bottom left image of Figure 4.4.14. Another related observation is that 9 weeks post-injury, in the D animal group the nerve trunk (nerve plus capsule) is always in touch with the scaffold which shirks and adapts its shape to the shape of the nerve trunk. On the contrary, 9 weeks post-injury in the E animal group the nerve trunk inside the gap usually does not make contact with the surrounding scaffold; its outer diameter is much smaller compared to the inner diameter of the scaffold. These observations resemble the nerve trunks formed inside silicone tubes [Williams et al. 1983, 1985; Williams 1987]. These observations suggest that the cell-matrix adhesion between capsule cells and the inner lumen of the scaffold is stronger in group D and much more fragile in group 3. In group E, it is possible that as the capsule cells attempt to apply more force to the scaffold, the cell-scaffold connections break because they are not strong enough, and eventually the nerve trunk separates from the scaffold. This observation is related to the findings of Chapter 5, which show that the density of ligands for the two major collagen-binding integrins $\alpha1\beta1$, $\alpha2\beta1$ is much smaller in scaffold E compared to D. Fewer ligand correspond to more fragile cell-scaffold interaction and more fragile capsule-scaffold association.

The previous paragraph suggests that one key difference between groups D and E (which also could be related to the regulation of the capsule thickness) is the strength of the bond between capsule cells and the scaffold. This difference means that capsule cells can apply much more force to scaffold D compared to capsule E. At the same time, 1,2 and 9-week data suggest that scaffold D leads to the formation of a thinner capsule compared to scaffold E. This pattern between scaffolds D and E (the ability to transmit force to scaffold is inversely related to the size of the capsule that forms) resembles results observed experimentally in independent studies of peripheral nerve regeneration that used materials with very different properties compared to the ones used in this study (Table 4.5-1):

- [Chamberlain et al. 1998] and [Kemp et al. 2009] observed that the PN wound healing outcome in transected nerves grafted with silicone conduits is much worse (larger capsule, more wound contraction) compared to the wound healing outcome in transected nerves

grafted with collagen conduits similar to the scaffold D of this study. In contrast to collagen that provides many ligands for adhesion, silicone does not provide ligands and cells cannot attach and apply forces to it.

- The wound healing response to silicone conduits changes by opening two large holes on the silicone conduit. This enables the capsule to connect through the holes with the external capsule that surrounds the scaffold (Figure 4.4.2). This configuration enables the contractile forces that are applied by the capsule cells to be mechanically transmitted to the silicone scaffold, even though cells do not directly bind to the scaffold (the two interconnected capsules act as a cuff). Opening holes in the silicone conduits results in a much thicker nerve trunk and in the formation of ECM inside the holes [Jenq 1985b]. The same result is observed when the two holes on the walls of the silicone conduit are covered by a cell-permeable 5 μm filter. However, when the holes are covered by a cell impermeable 1.2μm filter the response resembles the one of the silicone tube without holes [Jenq 1987].
- Transected nerves treated with PAN-PVC conduits (artificial polymer, no ligands) of smooth (cell impermeable) inner lumen surface result in the formation of a fibrin cable and then a nerve trunk similar to the one observed when using silicone conduits. In this case capsule cells cannot attach to the conduit inner lumen and apply forces to the conduit. However, PAN-PVC conduits (same material) of rough (cell permeable) inner lumen result in a completely different outcome: the fibrin matrix covers the whole gap volume and a very thin capsule formes. In this case perineurial cells infiltrate the PAN-PVC conduit. Even though individual cells cannot bind to the conduit, it is possible that arrays of interconected cells that penetrate the conduit pores could apply forces to the conduit [Aebischer 1990].
- Finally in transected nerves treated with scaffold D or scaffold E, 9-weeks post injury the resulting nerve tissue in the group D has much larger diameter and more axons compared to the transected nerves treated with scaffold E [Soller 2012]. Based on the findings on Chapter 5 of this thesis, scaffold D provides to cells much more ligands compared to scaffold E, making possible a stronger adhesion between capsule cells and the inner lumen of the scaffold (which is smooth as shown in [Soller 2011])

Reference	**Group 1**	**Group 2**
	Cells can apply limited force to scaffold	**Cells can apply force to scaffold**
	More wound contraction	**Less wound contraction**
Ikegutchi 2003	Silicone conduits	Silicone conduits with C^- anions
Jenq and Coggeshall 1985b, Jenq et al.1987	Silicone conduits with holes and cell-impermeable filter	Silicone conduits with holes Silicone conduits with holes and cell-permeable filter
Aebischer 1990	PMA-PVC smooth surface	PMA-PVC rough surface
Chamberlain et al. 1998, Kemp et al. 2009	Silicone conduits	Collagen conduits
Harley et al. 2004 Soller et al. 2012 This study	Collagen conduits E	Collagen conduits B, C, D

Table 4.5-1: Studies that provide that the ability of capsule cells to apply forces to the scaffold is related to the size of the capsule that forms and the resulting wound contraction.

The observations of Table 4.5-1 suggest a critical role for mechanics and mechano-biology in determining the wound healing outcome in transected nerves grafted with conduits. One week post-injury (when the capsule close to the point of transection has formed) the tissue surrounded by the capsule is expected to be more compliant compared to normal nerve tissue (due to remodeling of the initial fibrin clot). These findings agree with a recent mathematical model of the pressure cuff hypothesis, which suggests that a biomaterial can favor regeneration by delaying the formation of the capsule in order to prevent its forces from contracting the complaint nerve tissue present in the early phases of wound healing [Buydash 2013].

4.6 Summary and Future Work

This chapter describes an imaging study of peripheral nerve regeneration. Transected rat sciatic nerves were treated with either an "active" collagen scaffold (D) or an "inactive" collagen scaffold (E), sacrificed 1 or 2 weeks post-injury, and then imaged *ex vivo*. The two biomaterials are known to induce significantly different regeneration in peripheral nerve 9 weeks post injury [Soller et al. 2012]. Specifically, observation show that the quality of the wound healing outcome (area of new nerve tissue, number of axons observed in gap midpoint) is inversely related to the thickness of the contractile cell capsule that forms around the injured organ.

The results of this study attempt to trace the findings of [Soller et al. 2012] back in the early phases of wound healing after nerve transection and entubulation. Findings suggest that already 1 week post-injury the two scaffolds induce different wound healing response. Although a contractile cell capsule is observed in both animal groups 1 week post-injury, scaffold D results in a significantly thinner less-contractile capsule that is in more association with the inner lumen of the scaffold. The early capsules observed 2 weeks post-injury close to the nerve front are much smaller (50μm) and contain cells that are much less elongated compared to the capsuls observed 9 weeks-post injury.

Future experiments can aim in probing more the critical role of mechanics and in designing materials that induce better regeneration by regulating the growth of the capsule between 2 weeks and 9 weeks. Unfortunately, force measurements *in vivo* are yet impossible. Nevertheless, it is possible to design experiments where one could modify the maximum force that can be applied by cells to the scaffold (e.g. by modifying the scaffold surface chemistry) and evaluate tis effect on the wound healing outcome.

Up to now, the available experimental evidence provides a correlation between wound contraction and the outcome of regeneration. At the moment there is no study that attempted to design an experiment in such a way that could prove a direct causality relationship between wound contraction (expression of contractile phenotypes during wound healing) and the outcome of regeneration. Such experiments could be designed by using specific agents (e.g. shRNA lentiviruses) that can target and knock-down specific critical molecular components that mediate wound contraction (e.g. components of the contractile acto-myosin machinery). Such experiments could also be developed by modifying the chemical composition of biomaterials and doping them with components known to impede or accelerate contraction. One example of such an experiment could be the addition of EDA splice variant of fibronectin (a known mediator of myofibroblast differentiation) to "active" scaffolds known to be able to induce regeneration. In both cases those experiments would attempt to prove some sort of causality by using various

different ways to impede or accelerate wound contraction, measuring the wound healing contraction, the wound healing outcome, and examining the trend of the results.

Finally, the role of the scaffold in the formation of the contractile capsule can be studied in more detail by *in vivo* intravital imaging via multi-photon microscopy. In contrast to existing intravital imaging studies of nerve wound healing or the diseased nervous system [Misgeld and Kerschensteiner 2006], PN regeneration studies would require imaging of the injured nerve in the presence of a biomaterial. Combining transgenic animals, biomaterials and multi-photon imaging can be used to study the dynamic nature of contractile capsule formation and the results of various biomaterial designs in regulating capsule formation and eventually the wound healing outcome.

4.7 Acknowledgements

This study would not have been possible without the hard work of Eric Soller who did the animal experiment and prepared then nerve samples. This valuable assistance of Melissa Buydash and former UROP student Rebecca Lin in the staining and preparation of the nerve samples is also acknowledged.

4.8 Literature Cited

Abou Neel EA, Bozec L, Knowles JC, Syed O, Mudera V, Day R, and Hyun JK. Collagen--emerging collagen based therapies hit the patient. *Adv Drug Deliv Rev.* **65**(4): 429-56, 2013. (doi: 10.1016/j.addr.2012.08.010).

Aebischer P, Guénard V, and Brace S. Peripheral nerve regeneration through blind-ended semipermeable guidance channels: effect of the molecular weight cutoff. *J Neurosci.* **9**(10):3590-5, 1989.

Aebischer P, Guénard V, and Valentini RF. The morphology of regenerating peripheral nerves is modulated by the surface microgeometry of polymeric guidance channels. *Brain Res.* **531**(1-2): 211-8, 1990.

Akassoglou K, Yu WM, Akpinar P, and Strickland S. Fibrin inhibits peripheral nerve remyelination by regulating Schwann cell differentiation. *Neuron.* **33**(6): 861-75, 2002a.

Akassoglou K, and Strickland S. Nervous system pathology: the fibrin perspective. *Biol Chem.* **383**(1): 37-45, 2002b.

Akassoglou K, Akpinar P, Murray S, and Strickland S. Fibrin is a regulator of Schwann cell migration after sciatic nerve injury in mice. *Neurosci Lett.* **338**(3): 185-8, 2003.

Arbuthnott ER, Boyd IA, and Kalu KU. Ultrastructural dimensions of myelinated peripheral nerve fibres in the cat and their relation to conduction velocity. *J Physiol.* **308**: 125-57, 1980.

Badylak SF, Taylor D, and Uygun K. Whole-organ tissue engineering: decellularization and recellularization of three-dimensional matrix scaffolds. *Annu Rev Biomed Eng.* **13**: 27-53, 2011. (doi: 10.1146/annurev-bioeng-071910-124743).

Bozkurt A, Lassner F, O'Dey D, Deumens R, Böcker A, Schwendt T, Janzen C, Suschek CV, Tolba R, Kobayashi E, Sellhaus B, Tholl S, Eummelen L, Schügner F, Damink LO, Weis J, Brook GA, and Pallua N. The role of microstructured and interconnected pore channels in a collagen-based nerve guide on axonal regeneration in peripheral nerves. *Biomaterials*

33(5): 1363-75, 2012.

Buydash M. Mechanical Modeling of Tissue Response During Early Stage Entubulated Peripheral Nerve Regeneration, SM Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, June 2013.

Carbonetto S. The extracellular matrix of the nervous system. *Trends Neurosci.* **7**: 382-7, 1984.

Chamberlain LJ, Yannas IV, Arrizabalaga A, Hsu HP, Norregaard TV, and Spector M. Early peripheral nerve healing in collagen and silicone tube implants: myofibroblasts and the cellular response. *Biomaterials* **19**(15): 1393-403, 1998.

Chamberlain LJ, Yannas IV, Hsu HP, Strichartz G, and Spector M. Collagen-GAG substrate enhances the quality of nerve regeneration through collagen tubes up to level of autograft. *Exp Neurol.* **154**(2): 315-29, 1998b.

Chamberlain LJ, Yannas IV, Hsu HP, and Spector M. Connective tissue response to tubular implants for peripheral nerve regeneration: the role of myofibroblasts. *J Comp Neurol.* **417**(4):415-30, 2000.

Chang A, Yannas IV, Perutz S, Loree H, Sethi RR, Krarup C. et al. Electrophysiological study of recovery of peripheral nerves regenerated by a collagen–glycosaminoglycan copolymer matrix, *Progress in biomedical polymers*, Plenum, New York, 1990.

Chen X, Nadiarynkh O, Plotnikov S, Campagnola PJ. Second harmonic generation microscopy for quantitative analysis of collagen fibrillar structure. *Nat Protoc.***7**(4): 654-69, 2012.

Chen ZL, Yu WM, and Strickland S. Peripheral regeneration. *Annu Rev Neurosci.* **30**: 209-33, 2007.

Desmoulière A, Chaponnier C, Gabbiani G. Tissue repair, contraction, and the myofibroblast. *Wound Repair Regen.* **13**(1): 7-12, 2005.

Deumens R, Bozkurt A, Meek MF, Marcus MA, Joosten EA, Weis J, Brook GA. Repairing injured peripheral nerves: Bridging the gap. *Prog Neurobiol.* **92**(3): 245-76, 2010.

Flores AJ, Lavernia CJ, and Owens PW. Anatomy and physiology of peripheral nerve injury and repair. *Am J Orthop (Belle Mead NJ)* **29**(3):167-73, 2000.

Freyman TM, Yannas IV, Yokoo R, and Gibson LJ. Fibroblast contraction of a collagen-GAG matrix. *Biomaterials* **22**(21): 2883-91, 2001.

Friede RL, and Bischhausen R. The organization of endoneural collegen in peripheral nerves as revealed with the scanning electron microscope. *J Neurol Sci.* **38**(1):83-8, 1978.

Fu SY, and Gordon T. The cellular and molecular basis of peripheral nerve regeneration. *Mol Neurobiol.* **14**(1-2): 67-116, 1997.

Gordon T. The role of neurotrophic factors in nerve regeneration. *Neurosurg Focus* **26**(2): E3, 2009.

Gurtner GC, Werner S, Barrandon Y, Longaker MT. Wound repair and regeneration. *Nature* **453**(7193): 314-21, 2008. (doi: 10.1038/nature07039).

Harley BA, Spilker MH, Wu JW, Asano K, Hsu HP, Spector M, and Yannas IV. Optimal degradation rate for collagen chambers used for regeneration of peripheral nerves over long gaps. *Cells Tissues Organs* **176**(1-3):153-65, 2004.

Harley BA, Freyman TM, Wong MQ, and Gibson LJ. A new technique for calculating individual dermal fibroblast contractile forces generated within collagen-GAG scaffolds. *Biophys J.* **93**(8): 2911-22, 2007.

Hinz B. Formation and function of the myofibroblast during tissue repair. *J Invest Dermatol.* **127**(3): 526-37, 2007.

IJkema-Paassen J, Jansen K, Gramsbergen A, Meek MF. Transection of peripheral nerves, bridging strategies and effect evaluation. *Biomaterials* **25**(9): 1583-92, 2004.

Ikeguchi R, Kakinoki R, Matsumoto T, Tsuji H, Ishikawa J, Nakamura T. Rat nerve regeneration through a silicone chamber implanted with negative carbon ions. Brain Res Dev Brain Res. **140**(1):127-31, 2003.

Jenq CB, and Coggeshall RE. Long-term patterns of axon regeneration in the sciatic nerve and its tributaries. *Brain Res.* **345**(1): 34-44, 1985a.

Jenq CB, and Coggeshall RE. Nerve regeneration through holey silicone tubes. *Brain Res.* **361**(1-2): 233-41, 1985b.

Jenq CB, Jenq LL, and Coggeshall RE. Nerve regeneration changes with filters of different pore size. *Exp Neurol.* **97**(3): 662-71, 1987.

Joseph NM, Mukouyama YS, Mosher JT, Jaegle M, Crone SA, Dormand EL, Lee KF, Meijer D, Anderson DJ, Morrison SJ. Neural crest stem cells undergo multilineage differentiation in developing peripheral nerves to generate endoneurial fibroblasts in addition to Schwann cells. *Development* **131**(22):5599-612, 2004.

Kamoshida S, and Tsutsumi Y. Expression of MUC-1 glycoprotein in plasma cells, follicular dendritic cells, myofibroblasts and perineurial cells: immunohistochemical analysis using three monoclonal antibodies. *Pathol Int.* **48**(10): 776-85, 1998.

Kemp SW, Syed S, Walsh W, Zochodne DW, and Midha R. Collagen nerve conduits promote enhanced axonal regeneration, schwann cell association, and neovascularization compared to silicone conduits. *Tissue Eng Part A.* **15**(8): 1975-88, 2009.

Kerns JM. The microstructure of peripheral nerves. *Tech. Reg. Anesth. Pain Manag.* **12**: 127-133, 2008.

Liu HM. The role of extracellular matrix in peripheral nerve regeneration: a wound chamber study. *Acta Neuropathol.* **83**(5):469-74, 1992.

Longo FM, Hayman EG, Davis GE, Ruoslahti E, Engvall E, Manthorpe M, and Varon S. Neurite-promoting factors and extracellular matrix components accumulating in vivo within nerve regeneration chambers. *Brain Res.* **309**(1): 105-17, 1984.

Lundborg G, Gelberman RH, Longo FM, Powell HC, Varon S. In vivo regeneration of cut nerves encased in silicone tubes: growth across a six-millimeter gap. *J Neuropathol Exp Neurol.* **41**(4): 412-22, 1982a.

Lundborg G, Longo FM, Varon S. Nerve regeneration model and trophic factors in vivo. *Brain Res.* **232**(1):157-61, 1982b.

Lundborg G, Dahlin LB, Danielsen N, Gelberman RH, Longo FM, Powell HC, and Varon S. Nerve regeneration in silicone model chambers: influence of gap length and of distal stump components. *Exp Neurol* **76**(2):361-75, 1982c.

Madison RM, Archibald SJ, and Krarup C. in: Wound Healing, Cohen IK, Diegelmann RF, and Lindblad WJ (Eds). p. 450. Saunders, Philadelphia, 1992.

Misgeld T, and Kerschensteiner M. In vivo imaging of the diseased nervous system. *Nat Rev Neurosci.* 7(6): 449-63, 2006.

Mizisin AP, and Weerasuriya A. Homeostatic regulation of the endoneurial microenvironment during development, aging and in response to trauma, disease and toxic insult. *Acta Neuropathol.* **121**(3): 291-312, 2011.

Müller H, Williams LR, and Varon S. Nerve regeneration chamber: evaluation of exogenous agents applied by multiple injections. *Brain Res.* **413**(2): 320-6, 1987.

Murphy GF, Orgill DP, and Yannas IV. Partial dermal regeneration is induced by biodegradable collagen-glycosaminoglycan grafts. *Lab Invest.* **62**(3): 305-13, 1990.

Orlando G, Wood KJ, Stratta RJ, Yoo JJ, Atala A, and Soker S. Regenerative medicine and organ transplantation: past, present, and future. *Transplantation.* **91**(12): 1310-7, 2011. (doi: 10.1097/TP.0b013e318219ebb5).

Parmantier E, Lynn B, Lawson D, Turmaine M, Namini SS, Chakrabarti L, McMahon AP, Jessen KR, Mirsky R. Schwann cell-derived Desert hedgehog controls the development of peripheral nerve sheaths. *Neuron* **23**(4): 713-24, 1999.

Pomerantz J, and Blau HM. Nuclear reprogramming: a key to stem cell function in regenerative medicine. *Nat Cell Biol.* **6**(9):810-6, 2004.

Popović M, Bresjanac M, and Sketelj J. Regenerating axons enhance differentiation of perineurial-like cells involved in minifascicle formation in the injured peripheral nerve. *J Neuropathol Exp Neurol* **53**(6): 590-7, 1994.

Poss KD. Advances in understanding tissue regenerative capacity and mechanisms in animals. *Nat Rev Genet.* **11**(10): 710-22, 2010. (doi: 10.1038/nrg2879).

Scaravilli F. The influence of distal environment on peripheral nerve regeneration across a gap. *J. Neurocytol.* **13**(6): 1027-41, 1984.

Schmalbruch H. Fiber composition of the rat sciatic nerve. *Anat Rec.* **215**(1): 71-81, 1986.

Schmidt CE, and Leach JB. Neural tissue engineering: strategies for repair and regeneration. *Annu Rev Biomed Eng.* **5**: 293-347, 2003.

Schröder JM, May R, Weis J. Perineurial cells are the first to traverse gaps of peripheral nerves in silicone tubes. *Clin Neurol Neurosurg.* **95** Suppl:S78-83, 1993.

Soller EC. *Cell-mediated Contraction & Induced Regeneration of the Injured Peripheral Nerve*. PhD Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 2011.

Soller EC, Tzeranis DS, Miu K, So PT, Yannas IV. Common features of optimal collagen scaffolds that disrupt wound contraction and enhance regeneration both in peripheral nerves and in skin. *Biomaterials* **33**(19): 4783-91, 2012.

Sondell M, Lundborg G, and Kanje M. Vascular endothelial growth factor stimulates Schwann cell invasion and neovascularization of acellular nerve grafts. *Brain Res.* **846**(2): 219-28, 1999.

Sylvester MF, Yannas IV, Salzman EW, and Forbes MJ. Collagen banded fibril structure and

the collagen-platelet reaction. *Thromb Res.* **55**(1): 135-48, 1989.

Tanaka E, and Galliot B. Triggering the regeneration and tissue repair programs. *Development* **136**(3): 349-53, 2009. (doi: 10.1242/dev.031682).

Thomas PK. The connective tissue of peripheral nerve: an electron microscope study. *J Anat* **97**: 35-44, 1963.

Tomasek JJ, Gabbiani G, Hinz B, Chaponnier C, Brown RA. Myofibroblasts and mechano-regulation of connective tissue remodelling. *Nat Rev Mol Cell Biol.* **3**(5): 349-63, 2002.

Tona A, Perides G, Rahemtulla F, and Dahl D. Extracellular matrix in regenerating rat sciatic nerve: a comparative study on the localization of laminin, hyaluronic acid, and chondroitin sulfate proteoglycans, including versican. *J Histochem Cytochem.* **41**(4): 593-9, 1993.

Troxel KS. *Delay of skin wound contraction by porous collagen-GAG matrices*, PhD Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 1994.

Ushiki T, and Ide C. Three-dimensional organization of the collagen fibrils in the rat sciatic nerve as revealed by transmission- and scanning electron microscopy. *Cell Tissue Res* **260**(1):175-84, 1990.

Uygun BE, Soto-Gutierrez A, Yagi H, Izamis ML, Guzzardi MA, Shulman C, Milwid J, Kobayashi N, Tilles A, Berthiaume F, Hertl M, Nahmias Y, Yarmush ML, and Uygun K. Organ reengineering through development of a transplantable recellularized liver graft using decellularized liver matrix. *Nat Med.* **16**(7): 814-20, 2010. (doi: 10.1038/nm.2170).

Weis J, May R, and Schröder JM. Fine structural and immunohistochemical identification of perineurial cells connecting proximal and distal stumps of transected peripheral nerves at early stages of regeneration in silicone tubes. *Acta Neuropathol.* **88**(2):159-65, 1994.

White ES, Baralle FE, and Muro AF. New insights into form and function of fibronectin splice variants. *J Pathol.* **216**(1): 1-14, 2008. (doi: 10.1002/path.2388).

Williams LR. Exogenous fibrin matrix precursors stimulate the temporal progress of nerve regeneration within a silicone chamber. *Neurochem Res.* **12**(10): 851-60, 1987.

Williams LR, Longo FM, Powell HC, Lundborg G, and Varon S. Spatial-temporal progress of peripheral nerve regeneration within a silicone chamber: parameters for a bioassay. *J Comp Neurol.* **218**(4): 460-70, 1983.

Williams LR, Powell HC, Lundborg G, and Varon S. Competence of nerve tissue as distal insert promoting nerve regeneration in a silicone chamber. *Brain Res.* **293**(2): 201-11, 1984.

Williams LR, and Varon S. Modification of fibrin matrix formation in situ enhances nerve regeneration in silicone chambers. *J Comp Neurol.* **231**(2): 209-20, 1985.

Williams LR, Danielsen N, Müller H, and Varon S. Exogenous matrix precursors promote functional nerve regeneration across a 15-mm gap within a silicone chamber in the rat. *J Comp Neurol.* **264**(2): 284-90, 1987.

Yannas IV. *Tissue and organ regeneration in adults*, Springer, New York, 2001.

Yannas IV, and Tobolsky AV. Cross-linking of gelatine by dehydration. *Nature* **215**(5100): 509-10, 1967.

Yannas IV, Lee E, Orgill DP, Skrabut EM, and Murphy GF. Synthesis and Characterization of a model extracellular-matrix that induces partial regeneration of adult mammalian skin. *Proc*

Natl Acad Sci U.S.A. **86**: 933-7, 1989.

Yannas IV, Zhang M, and Spilker MH. Standardized criterion to analyze and directly compare various materials and models for peripheral nerve regeneration. *J Biomater Sci Polym Ed.* **18**(8): 943-66, 2007.

Yoshii S, Oka M, Shima M, Taniguchi A, and Akagi M. Bridging a 30-mm nerve defect using collagen filaments. *J Biomed Mater Res A.* **67**(2): 467-74, 2003.

Zhao Q, Dahlin LB, Kanje M, and Lundborg G. The formation of a 'pseudo-nerve' in silicone chambers in the absence of regenerating axons. *Brain Res.* **592**(1-2):106-14, 1992.

Zhu S, Wei W, and Ding S. Chemical strategies for stem cell biology and regenerative medicine. *Annu Rev Biomed Eng.***13**: 73-90, 2011. (doi: 10.1146/annurev-bioeng-071910-124715).

Zochodne DW, and Nguyen C. Angiogenesis at the site of neuroma formation in transected peripheral nerve. *J Anat.* **191**(Pt. 1):23-30, 1997.

Chapter 5: An Optical Method for the Quantification of the Surface Chemistry in 3D Matrices *in Situ*

5.1 Chapter Overview

This chapter describes a novel optical method for quantifying the surface chemistry of 3D matrices (extracellular matrix or biomaterials) *in situ*. The surface chemistry of a matrix is defined as the density of adhesion ligands for particular cell adhesion receptors available to cells for binding on the matrix surface. Although it is known that the surface chemistry of a matrix affects the micro-environment felt by cells and can regulate cell phenotypes, at this time there are very few methods that can quantify the surface chemistry of matrices *in situ*. This work fills part of this knowledge gap by providing a method than can quantify the surface density of adhesion ligands for particular cell adhesion receptors that are available on the surface of a matrix for cell binding.

The proposed method is an *in situ* binding assay that quantifies optically the surface density of fluorescent markers that bind on to the same adhesion ligands as the receptor of interest on the surface of the matrix. The assay is based on two key technologies. The first is the development of biomarkers that i) emulate the binding function of the adhesion receptor of interest, and ii) are fluorescent so that they can be detected optically. The second key technology is the use of spectral 3D multi-photon imaging (see Chapter 2) which i) utilizes optical sectioning to provide *in situ* measurements of the biomarkers bound on the complex surface of a matrix with 3D spatial resolution, and ii) utilizes spectral information to distinguish the signal of the markers from the signal emitted by the matrix itself.

The methodology that was developed has been implemented by quantifying the surface chemistry of porous collagen scaffolds, similar to the ones used clinically to induce regeneration in severely injured skin and peripheral nerves. Specificially, it is of interest to measure the density of ligands for the two major collagen-binding integrins (CBI) $\alpha_1\beta_1$, $\alpha_2\beta_1$. The study utilizes as markers of these receptors the I Domains of their α subunit. Experimental results provide the first estimates for the surface chemistry of these materials for ligands of $\alpha_1\beta_1$ and $\alpha_2\beta_1$, and demonstrate that different methods for cross-linking scaffolds can affect significantly their surface chemistry. These findings support recent evidence on the role of mechanics and kinetics on wound healing (Chapter 4) and provide new tools for designing and quantifying the chemical environment felt by cells either in tissue ECM or inside biomaterials.

5.1.1 Motivation

There is extensive evidence that the insoluble microenvironment around cells (the "matrix") is a critical modulator of cell phenotypes. The available experimental evidence includes the ability of cell-free biomaterials to induce regeneration in severely injured organs *in vivo*, and also includes **evidence in** thousands of published papers that describe how several properties of the matrix (chemical composition, stiffness, microstructure) affect various phenotypes of various kinds of

cells *in vitro*. At the moment the efforts of many research labs focus on the effects of the matrix: developing new materials for tissue engineering and regenerative applications [Maskarinec and Tirrell 2005; Nelson and Tien 2006; Sands and Mooney 2007; Soller et al., 2012], understanding the interplay between cancer cells and tumor ECM [Paszek et al. 2005; Naba et al. 2012], developing biomaterials that can guide the differentiation of stem cells [Huebsch et al. 2010], developing novel 3D printing tools that could generate the next generation of "smart" biomaterials [Derby 2012; Miller et al. 2012].

The insoluble microenvironment of cells is described in much more complex terms compared to the soluble microenvironment (e.g. growth factors, cytokines). The soluble microenvironment can be described well by the composition and the concentration of its constituents, parameters that can be quantified by several established proteomic methods. On the other hand, the insoluble microenvironment felt by cells is described by a larger number of properties including chemical composition, surface density, topology, stiffness, degradation rate, and stress field. The complex geometry of the matrix (ECM in tissues or biomaterials) makes it difficult to measure *in situ* many of these properties. Nevertheless, *in situ* measurements are important because they provide a direct description of the environment felt by cells, in contrast to average macroscopic properties (e.g. mass fraction, chemical composition) that are convenient to use and easier to measure, but are not entities directly sensed by individual cells.

One of the key properties of the matrix around cells is its surface chemistry, defined in this study as the identity and surface density of adhesion ligands available to cells for binding. The surface chemistry of a matrix defines which adhesion receptors can be utilized by cells for adhesion, therefore controls the perception of the cell about its insoluble environment and affects the types of signals sensed by cells and therefore the resulting cell response. Despite its importance, there are very few published methods for quantifying the surface density of matrices, and the latter can be applied only to a small class of artificial biomaterials (Section 5.2.7). This is of no surprise, since even quantifying the bulk chemical composition of a matrix (the identity, composition, and modifications) is by itself a challenging task [Naba et al. 2012].

This chapter attempts to fill part of this knowledge gap by providing a methodology that can quantify the surface chemistry of a 3D matrix. The method is demonstrated by quantifying the surface chemistry of collagen scaffolds, and specifically by quantifying the adhesion ligands for the major collagen-binding integrins ($\alpha_1\beta_1$, $\alpha_2\beta_1$). The methodology can be extended to characterize other kinds of 3D insoluble environments felt by cells inside tissues or biomaterials of various chemical composition (not just collagen).

5.1.2 Chapter Description

Section 5.2 provides background on subjects related to the developed methodology, such as cell adhesion to collagens via CBI, the effects of cell adhesion to collagen, and the current state of the art of methods that quantify the surface chemistry of biomaterials. Section 5.3 describes the various parts of the methodology and its application in quantifying the density of CBI ligands in collagen scaffolds. A big part of the methodology (Sections 5.3.2 to 5.3.5) describes the expression and extensive biochemical characterization of recombinant I Domains, utilized in this study as markers of CBI. The remainder of Section 5.3 describes the experimental procedure for quantifying and interpreting the density of adhesion ligands *in situ*. Section 5.4 provides experimental results, first about the characterization of recombinant I Domains, and then

presents measurements of the surface chemistry of three kinds of porous collagen scaffolds used in nerve regeneration studies. Section 5.5 discusses the methodology, its findings, and their implications. Section 5.6 suggests directions for future research.

5.2 Background

5.2.1 The Extracellular Matrix as a Critical Modulator of Cell Phenotypes

What is the Matrix? Insoluble Regulators of Cells

Cells perceive their environment using receptors on their plasma membrane that bind to molecules located in their immediate environment. Binding of a receptor to such a molecule induces a cascade of chemical events inside the cell cytoplasm, referred to as "cell signaling" [Gomperts et al. 2002]. The outcome of a cell signaling cascade can be either a fast short-term response (e.g. cytoskeleton polymerization), or a slow long-term response (e.g. gene expression, differentiation), that eventually regulate cell phenotypes. These molecules, which are present in the cell's immediate environment, and which bind to its receptors, are referred as "modulators". For simplicity of discussion, modulators can be classified as:

- *Soluble regulators*: these are molecules that exist in solution, and reach cell's receptors via diffusion. Examples of soluble regulators include cytokines, hormones, and growth factors. Each class of soluble regulators binds to a distinct class of receptors and induces a distinct set of intracellular signaling cascades [Lodish et al. 2007].
- *Insoluble regulators*: these are molecules that exist in the solid state, form solid structures (e.g. fibers, lamellae) where cells can apply forces and induce strain. Binding (adhesion) of cells to insoluble regulators via adhesion receptors is utilized by cells not just for cell signaling, but also as a means to exert forces to their environment.

The insoluble microenvironment of cells (matrix) can refer to two kinds of matrix:

- *Extracellular matrix* (ECM) refers to the native insoluble microenvironment of cells in tissues. In animals, organs are composed of tissues, which comprise tissue-specific cells surrounded by tissue-specific ECM. The appropriate combination of cells and ECM provides the necessary strength and functionality to tissues and organs. ECM is a complex inter-connected network of macromolecules including structural proteins (collagens, elastin), adhesion proteins (fibronectin, laminin), proteoglycans, glycosaminoglycans, and bound soluble factors [Bosman and Stamenkovic 2003; Vakonakis and Campbell 2007; Hynes 2009; Naba et al. 2012].
- *Biomaterials* refer to materials fabricated by humans that can interact with cells. Usually biomaterials are used as implants inside the body to support the function or structure of tissues and organs. Based on their chemical composition, they can be classified as "natural" (made of biopolymers found in tissues, e.g. proteins, polysaccharides) or "artificial" (made of artificial polymers not found in tissues) [Ratner and Bryant 2004].

Experimental Evidence for the Role of Insoluble Regulators: Wound Healing

Several pieces of experimental evidence suggest the critical importance of the matrix in regulating the phenotypes of the cells that interact with it.

The field of regenerative medicine provides three major experimental observations:

- After severe organ injury, the outcome of wound healing depends in the nature of the injured tissue, and changes significantly when injury involves ECM-rich tissues. The outcome of wound healing is classified as "regeneration" when it is structurally and functionally similar to or identical to the physiological organ; otherwise it is classified as "repair". Severely injured organs in adults do not regenerate spontaneously; instead they heal by eventually forming a scar. Skin is one example where the wound healing outcome is repair, with scar forming when ECM-rich tissues (stroma) become injured [Yannas 2001]. The cell-rich epidermis tissue can regenerate spontaneously, however the underlying ECM-rich dermis does not regenerate when injured. Similar results have been observed in other organs, including peripheral nerves (Chapter 4).
- At this time, clinically successful treatments for inducing regeneration in two kinds of injured organs (skin and peripheral nerves) are based on collagen-based biomaterials with particular properties. In the form of highly porous "scaffolds", these materials can change the kinetics of wound healing and induce partial regeneration in skin [Yannas et al. 1982] and peripheral nerves [Chamberlain et al. 1998]. Although the performance of these materials can be enhanced by the presence of cells or soluble regulators contained within, the presence of an insoluble biomaterial seems to be an absolute prerequisite [Yannas 2001]. Furthermore, animal experiments show that the performance of these materials depends significantly on several physicochemical properties of the biomaterial, such as mean pore diameter, degradation rate and chemical composition [Yannas et al. 1989; Soller et al. 2012]. It seems that these biomaterials contain the key "instructions" that guide cells towards inducing regeneration.
- Recently, several promising approaches attempt to induce regeneration using decellularized animal organs [Uygun et al. 2010; Badylak et al. 2011; Orlando et al. 2011]. This approach can be thought of as an extension of the use of biomaterials: instead of using a biomaterial of simple chemical composition, the motivation is to use a more complex scaffold that contains a larger variety of insoluble clues and whose properties vary over its volume. Since current knowledge cannot prescribe with certainty the optimal matrix parameters for all organs, and since current fabrication methods cannot fabricate scaffolds as complex as those currently used as decellularized matrices, it appears intuitive to use as scaffold the native ECM of animal organs, after removing cells or anything else that potentially can trigger an immune response. Again, the success of decellularized organs to guide the formation of new functional organs, suggests that the ECM contains the key "instructions" that guide cells towards inducing regeneration.

Experimental Evidence for the Role of Insoluble Regulators: Cancer

Cancer biology provides experimental observations that support the critical role of the matrix in regulating cells during pathological conditions. Tumor-associated stroma is associated with all steps of cancer: initiation of carcinogenesis, tumor progression, and metastasis [Allen and Jones 2011]. ECM synthesized by tumor cells differs from normal ECM both in terms of chemical composition and in terms of mechanical properties. Tumor-associated ECM is usually denser and stiffer compared to normal ECM [Paszek et al. 2005], a sign of abnormal ECM remodeling dynamics (synthesis versus degradation) [Lu et al. 2012], and eventually leads to hypoxia, which induces tumor cells to synthesize soluble factors that favor ECM remodeling and

angiogenesis [Carmeliet et al. 1998; Canning et al. 2001]. The increased stiffness of tumor ECM can be attributed to altered chemical composition, increased cross-linking between ECM fibers (e.g. by increased lysyl oxidase activity [Kagan and Li 2003]) or different structural organization of collagen fibers [Provenzano et al. 2006]. Tumor ECM composition contains a different profile of ECM proteins compared to the one observed in normal ECM [Allen and Jones 2011; Naba 2012]. The deposition of several collagens (including I, II, and III) increases during tumor formation [Zhu et al. 1995; Kauppila et al. 1998; Huijbers et al. 2010]. Tumor ECM often contains increased amounts of particular isoforms or splice variant of proteins (such as tenascin C and the EDA splice variant of fibronectin) whose expression is observed during development or wound healing [Kurose et al. 2001; White et al. 2008]. The combination of stiff ECM and EDA fibronectin resembles other pathologic cases such as fibrosis and abnormal scar formation in wound healing [Serini et al. 1998].

The altered tumor ECM is not believed to be a second order effect of cancer, but an integrated part of the disease that affects critically tumor initiation and progression. The abnormal ECM synthesized by cancer-associated fibroblasts (CAF) has been shown to impact the behavior of epithelia, and could affect cancer initiation in locations of epithelial abnormalities [Olumi et al. 1999]. The stiff environment of tumor ECM can facilitate several cancer phenotypes (metastasis, neutrophil infiltration) via altered integrin signaling, and may affect the kinetics of tumor progression by affecting the activation of growth factors that bind on the ECM and whose activation requires mechanical stress. Tumor-associated stroma has been demonstrated to induce cancer-like phenotypes to normal cells *in vitro* [Amatangelo et al. 2005].

Implications for Biomaterial Design

Additional examples of systems where the insoluble microenvironment of cells acts as a niche that affects critically cell phenotype include the interaction between stroma cells and normal ECM in tissues (tissue homeostasis), and interactions between stem cells and ECM. Stem cell binding to ECM is essential for maintaining stem cell properties and guides asymmetric stem cell division [Shen et al. 2008, Raymond et al. 2009]. At the moment the molecular details of how the ECM guides cells in physiological and pathological conditions are just beginning to unravel. Knowledge of the identity and spatial organization of the appropriate chemical clues (including adhesion ligands) could be utilized for the development of improved biomaterials that could guide cell phenotypes efficiently [Sands and Mooney 2007]. Biomaterial development based on reverse engineering of natural processes requires methods that can quantify the insoluble microenvironment felt by cells in ECM and inside biomaterials, such as the one described in this Chapter.

5.2.2 The Surface Chemistry of a Matrix

The microenvironment felt by cells of a particular type inside a matrix depends both on the set of adhesion receptors expressed by this cell type, and on the chemical entities present on the matrix where adhesion receptors can bind (adhesion ligands). In this thesis, the surface chemistry of a matrix is defined as the density of adhesion ligands for particular receptors available on the surface of a matrix for cells to bind. In contrast to the macroscopic chemical composition of the matrix, the concept of surface chemistry attempts to provide a description of the microenvironment as felt by cells (on the matrix surface cells sense and respond to adhesion ligands, not molecules).

Adhesion ligands provide cells with a means for binding to and perceiving their insoluble microenvironment; therefore these ligands encode part of the information provided by the matrix to cells, which as discussed in Section 5.2.1 can affect cell phenotypes significantly. The term 'adhesion ligand' in this study refers to the chemical entities recognized by cell adhesion receptors, located on the surface of a 'parent' molecule. This use of the term "ligand" is distinct from the use of the term "ligand" in biochemistry, where it describes the entire biomolecule [Rehm 2006]. Adhesion ligands are found in various ECM components. Adhesion ligands found in proteins (e.g. collagens, fibronectins) are small peptide motifs such as RGD and GFOGER. A list of known ligands of integrin receptors is provided in [Plow et al. 2000]. Adhesion ligands associated with polysaccharides (glycosaminoglycans, proteoglycans) consist of a small oligosaccharide sequence of particular chemical conformation [Varki 1994; Monsigny et al. 2000; Taylor and Drickamer 2007]. Many adhesion ligands are present in more than one ECM components (e.g. the RGD ligand). Furthermore, many ECM components contain more than one type of ligand or several copies of the same ligand (e.g. collagens) [Di Lullo 2002]. The functionality of adhesion ligands can be conformation-dependent, that is adhesion receptors can bind the ligand provided the parent molecule is folded correctly. In some cases, the number of exposed adhesion receptors that are available for cell binding depends on the stress state of the molecule and stretching the parent molecule exposes additional ligands (e.g. fibronectin) [Mao and Schwarzbauer 2005].

While ECM and "natural" biomaterials contain the adhesion ligands found in their constituent molecules, artificial biomaterials do not contain ligands and need to be functionalized by incorporating peptides or protein domains that contain adhesion ligands. At the moment the design of optimal artificial polymers by combining several building blocks, including blocks that contain adhesion ligands, is an area of active research [Nelson and Tien 2006; Sands and Mooney 2007].

There are several families of adhesion receptors in mammalian cells. The two most important are integrins (bind ECM proteins) and lectins (bind polysaccharides) [Hynes 2002; Sharon and Lis 2007]. Each adhesion receptor can bind to more than one type of ligand. The expression of each adhesion receptor is dependent on cell type. Alltogether, the same matrix will be perceived differently by different cell types since the latter express different sets of receptors. Therefore, it is expected that optimal biomaterial design depends strongly on the identity of the cells that will interact with the biomaterial.

5.2.3 Collagen and Collagen-Binding Integrins (CBI)

This thesis focuses on the surface chemistry of collagen-rich matrices, such as collagen biomaterials. The surface chemistry of such matrices is dominated by adhesion ligands of adhesion receptors that bind to collagen. This section describes in detail what is known about cellular adhesion to collagen.

Collagens

Collagens are a family of proteins found in the extracellular matrix (ECM) of tissues. Collagens are one of the most important components of ECM and provide both structural support and functionality to tissues and organs, where they are organized into tissue-specific super-molecular structures (fibers, fibrils, lamellae, meshes). At the moment 29 isoforms of collagen

have been identified [Ricard-Blum 2011]. The distribution of collagen isoforms is tissue-dependent [Gelse et al. 2003]. Different isoforms form different super-molecular structures: some collagen isoforms (types I, II, III, V, XI) form fibril structures, others (types IV, VIII, X) form network structures or beaded filaments (type VI), while still others (IX, XII, XIV, XIX, XX, XXI) associate with fibrils. Collagens bind to many other ECM components, such as adhesion proteins, and proteoglycans [Di Lullo et al. 2002; Sweeney et al. 2008].

Collagen I

Collagen type I is by far the most abundant isoform of collagen, expressed in multiple tissues (e.g. dermis, cornea, cervix, tendon, ligaments). Collagen I molecules are organized into super-molecular structures (fibrils, fibers) in a tissue-dependent way that is believed to be regulated by other ECM components [Gelse et al. 2003]. Specifically, five collagen molecules arrange themselves in space to form micro-fibrils, which are then organized into the much larger collagen fibrils. Fibril diameter ranges between 25 and 400 nm, depending on the tissue and the age of the animal. Collagen I fibrils have a semicrystalline structure that results in the formation of an axial 67nm periodicity called "D banding" [Gross and Schmitt 1948; Yannas 1972; Hulmes and Miller 1979; Orgel et al. 2006]. The process by which cells modulate collagen molecule organization into fibrils and fibers is an ongoing area of research [Canty et al. 2005].

Apart from being a major component of native ECM in tissues, collagen I purified from animal tissues is a natural polymer utilized in the development of *in vitro cell* assays, and in the fabrication of biomaterials, including the materials utilized in this thesis [Dagalakis et al. 1980; Yannas and Burke 1980; Yannas et al. 1980; O'Brien et al. 2004].

Families of Collagen Receptors

Mammalian cells use several families of adhesion receptors to bind to collagens: members of the integrin family, discoidin domain receptors (DDRs), glycoprotein VI (GPVI), leukocyte-associated IG-like receptor-1 (LAIR-1), and members of the mannose receptor family [Leitinger and Hohenester 2007; Heino et al. 2009; Leitinger 2011]. Cells can also bind denatured collagen through various RGD-binding integrins [Gullberg et al. 1992], or can bind collagen indirectly, by binding matrix proteins that associate with collagens (fibronectins, proteoglycans).

Integrins are the major collagen receptors [Leitinger 2011]. Of the twenty-four members of the integrin family, five ($\alpha_1\beta_1$, $\alpha_2\beta_1$, $\alpha_3\beta_1$, $\alpha_{10}\beta_1$, $\alpha_{11}\beta_1$) have been shown to bind at least one type of collagen [Heino 2000; Tulla et al. 2001]. Four integrins ($\alpha_1\beta_1$, $\alpha_2\beta_1$, $\alpha_{10}\beta_1$, $\alpha_{11}\beta_1$) comprise the collagen-binding integrins (CBI) [Hynes 2002], while $\alpha_3\beta_1$ acts probably as an assisting collagen receptor rather than a primary receptor [DiPersio et al. 1995]. The four CBIs have overlapping specificities for collagens and their expression in cells is tissue-dependent. Binding of CBIs to collagen has been shown to depend on divalent cations. Knocking down the major CBI results in severe birth defects [Hynes et al. 2002].

There are several types of non-integrin collagen receptors, reviewed in [Leitinger 2011]. The two DDR receptors (DDR1 and DDR2) are a subtype of receptor tyrosine kinases (RTK) with the unique feature that they bind to ECM molecules instead of growth factors. DDR receptors are expressed during development and bind specific types of collagen through a collagen-binding discoidin domain with affinity similar to integrins. DDR are also involved in the regulation of collagen fibrogenesis [Flynn et al. 2010] and CBI integrin signaling [Xu et al. 2012]. GPVI are

expessed in platelets. LAIR-1 is found in several immune cells. Both GPVI and LAIR-1 bind the collagen triple-helix with low affinity. Mannose receptors (e.g. Endo-180) bind both collagen and gelatin through their F2 domain and are involved in collagen/gelatin internalization by cells.

Collagen-Binding Integrins (CBI)

The expression of collagen-binding integrins ($\alpha_1\beta_1$, $\alpha_2\beta_1$, $\alpha_{10}\beta_1$, $\alpha_{11}\beta_1$) varies among different cell types: $\alpha_1\beta_1$ is abundant in smooth muscle cells, mesenchymal cells, hepatic stellate cells, pericytes, bone marrow stem cells, chondrocytes, and neuronal cells [Hamaia et al. 2012]. $\alpha_2\beta_1$ is the main collagen receptor in epithelia, lymphocytes and platelets. $\alpha_{10}\beta_1$ is present in chondrocytes. $\alpha_{11}\beta_1$ is expressed by fetal muscle cells, and by fibroblasts in regions of highly organized interstitial collagen networks [Carracedo et al. 2010]. Lately integrin $\alpha_{11}\beta_1$ has been found to be involved in myofibroblast differentaton, a critical event of the wound healing process [Carracedo et al. 2010; Talior-Volodarsky et al. 2012], see Chapter 4. Many cell types (fibroblasts, endothelia cells, lymphocytes) express more than one CBI [Heino 2000; Tiger et al. 2001; Tulla et al. 2001]. Table 5.2-1 provides a partial list of cell types that express both, one or none of $\alpha_1\beta_1$ and $\alpha_2\beta_1$, and have been used in CBI-related research.

Cell type	Express $\alpha_1\beta_1$	Express $\alpha_2\beta_1$	References
CHO	-	-	Jokinen et al. 2004
C2C12 (mouse myoblasts)	-	-	Hamaia et al. 2012
PC12 (rat adrenal pheochromocytoma)	+	-	Hamaia et al. 2012
Rugli (glioma) cells	+	-	Knight et al. 2000
Platelets	-	+	Knight et al. 1998, 2000; Onley et al. 2000; Raynal et al. 2006
HT 1080	-	+	Knight et al. 1998, 2000; Raynal et al. 2006
HGF (human gingivinal fibroblasts)	+	+	Jokinen et al. 2004
MC5 (human lung fibroblasts)*	+	+	Kim et al. 2005

Table 5.2-1: Examples of cell lines used in integrin-mediated collagen adhesion assays. Cell lines can express none, one, or both of the major collagen-binding integrins $\alpha_1\beta_1$, $\alpha_2\beta_1$.
* highlights comparable levels of $\alpha_1\beta_1$ and $\alpha_2\beta_1$ expression.

All integrins consist of two subunits, named α and β. Each subunit consists of four parts: head, leg, transmembrane section, and cytoplasmic domain, Figure 5.2.1. The two heads mediate and regulate binding to integrin ligands, while the remaining parts regulate integrin activity and transmit signals from and into the cell [Hynes 2002; Shimaoka et al. 2002; Luo et al. 2007]. The α subunits of CBI contain an additional domain close to their N terminus, which is called "I domain" **(or** sometimes "A domain") [Kamata and Takada 1994; Luo et al. 2007]. In this study, CBI I domains are referred to as α_1I, α_2I, α_{10}I, and α_{11}I respectively. In those integrins whose α subunit contains an I domain ($\alpha_1\beta_1$, $\alpha_2\beta_1$, $\alpha_{10}\beta_1$, $\alpha_{11}\beta_1$, $\alpha_L\beta_2$, $\alpha_M\beta_2$, $\alpha_X\beta_2$, $\alpha_D\beta_2$, $\alpha_E\beta7$), ligand binding is mediated entirely by this I domain [Hynes et al. 2002; Luo et al. 2007].

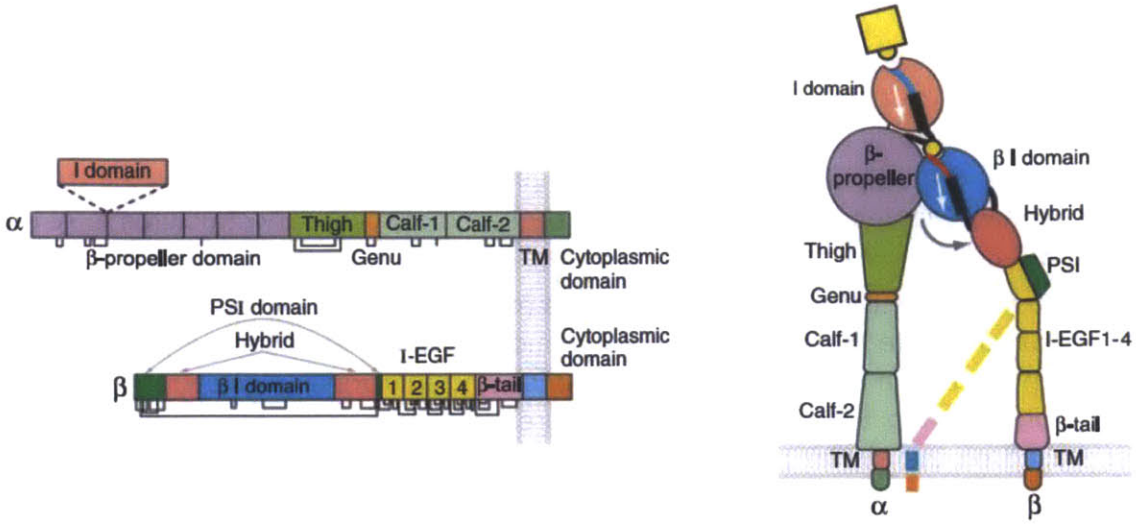


Figure 5.2.1: Schematic of collagen binding integrin structure. Left: Domains of α and β subunits. Right: schematic of an integrin-binding integrin bound to a ligand. Adopted from [Luo et al. 2007].

5.2.4 I Domains of Collagen-Binding Integrins

I domains of CBI have been shown to contain all the components necessary for binding the adhesion ligands of the parent integrin [Kamata and Takada 1994; Hynes 2002]. Similarly to the parent integrin, each type of I domain can bind multiple ligands with varying affinities [Calderwood et al. 1997; Hynes 2002]. I Domains are of critical importance to this thesis, because they are utilized as markers of the adhesion ligands of the parent integrin.

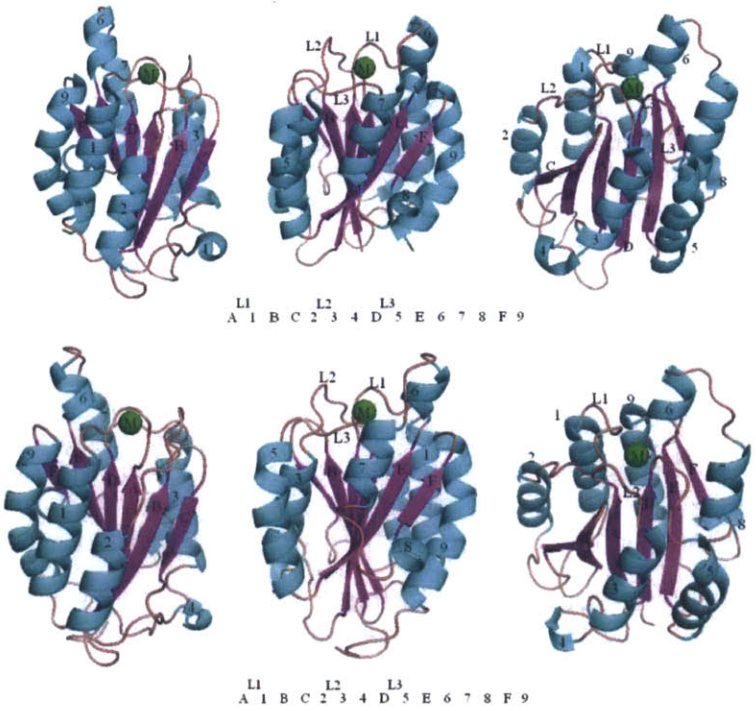


Figure 5.2.2: Crystal structure of I domains of the two major collagen-binding integrins. Top: α_1 I domain (structure 1QCY [Nymalm et al. 2004]). Bottom: α_2 I domain (structure 1AOX [Emsley et al. 1997]). Blue: α helix, magenta: β strand, grey: loop, M: Mg^{2+}. Structures are rendered using Pymol 1.0r1 using the crystal structures 1QCY, 1AOX that can be downloaded from the Protein Data Bank (PDB) database (http://www.rcsb.org).

I Domain Structure

The crystal structures of several I-domains in several conformation states have been obtained by x-ray crystallography (Table 5.2-2). Resolved structures show that I domains belong to the Rossmann fold superfamily [Emsley et al. 1997] They consist of a β sheet core containing six β strands (named A, B, … F) surrounded by nine amphipathic α helices (named 1, 2, … 9). All β strands are parallel with the exception of C. The sequence of secondary structures is A-1-B-C-2-3-4-D-5-E-6-7-8-F-9.

The four I domains of the collagen binding integrins (α_1I, α_2I, α_{10}I, α_{11}I) are highly homologous in terms of primary structure (Figure 5.2.3). The two most well-studied I domains (α_1I and α_2I) are also highly homologous in terms of secondary and tertiary structure as revealed by the resolved crystal structures (Figure 5.2.2). The x-ray structures for α_{10}I, α_{11}I are not currently available. However, based on the similar primary sequences (Figure 5.2.3), computational tools predict that α_{10}I, α_{11}I have secondary and tertiary structure similar to α_1I, and α_2I (Figure 5.2.4).

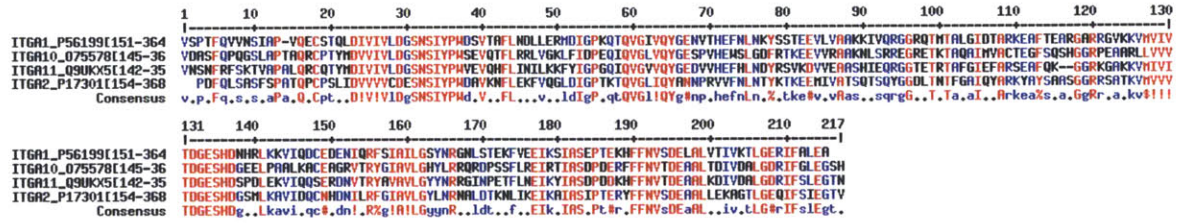


Figure 5.2.3: Multiple sequence alignment of the amino acid sequences of the I domains of the four collagen-binding integrins. α_1I: residues 151-364 of sequence P56199 (protein ITGA1). α_2I: residues 154-368 of sequence P17301 (protein ITGA2). α_{10}I: residues 145-361 of sequence O75578 (protein ITGA10). α_{11}I: residues 142-356 of sequence Q9UKX5 (protein ITGA11). Conserved residues are highlighted red, mostly conserved residues are highlighted blue. Multiple alignment was performed by the online tool MultAlin (http://multalin.toulouse.inra.fr/multalin/).

Figure 5.2.4: Secondary structure of the I domains of the four collagen-binding integrins. α_1I: residues 151-364 of sequence P56199 (protein ITGA1). α_2I: residues 154-368 of sequence P17301 (protein ITGA2). α_{10}I: residues 145-361 of sequence O75578 (protein ITGA10). α_{11}I: residues 142-356 of sequence Q9UKX5 (protein ITGA11). Blue: α helix, Green: β sheet, red: turn. Bold: highly-conserved residues. Secondary structures of α_1I and α_2I were based on the crystal structures 1QCY [Nymalm et al. 2004] and 1DZI [Emsley et al. 2000] respectively as annotated in the PDB

database (http://www.rcsb.org). Secondary structures of α_{10}I and α_{11}I were estimated based on the primary sequence of the proteins using the computational tool Jpred (http://www.compbio.dundee.ac.uk/www-jpred/).

I Domain States

Binding of I domains to their ligands is mediated by residues located in three different loops that form a metal ion-dependent adhesion site (MIDAS): loop L1 forms between β strand A and helix 1, loop L2 forms between helices 2 and 3, and loop L3 forms between β strand D and helix 5 (Figure 5.2.5). Binding of I domains to their ligands involves a divalent metal cation (Mn^{2+} or Mg^{2+} but not Ca^{2+}), water molecules, an acidic amino acid in I Domain loop L1, and an acidic amino acid in the ligand [Emsley et al. 2000].

Based on x-ray crystallography, I domains of CBI can exist in two distinct conformations, called "closed" and "open", which possess low and high affinity for ligand binding respectively (a list of solved crystal structures is provided in Table 5.2-2). Experimental evidence suggests that recombinant I domains in solution exist in both states, however equilibrium favors the "closed" conformation [Vorup-Jensen et al. 2007]. Conformation change is accompanied by lateral motion of the three loops (L1, L2, L3) that surround the MIDAS site, a change in the divalent cation coordination in the MIDAS site, and a large (10 Å) vertical motion of helix 9 at the C-terminus [Emsley et al. 2000]. The vertical motion of the C terminus α helix has been shown to affect the conformation state of I domains. Preventing the C-terminus helix motion locks the I domain in "closed" configuration. Interaction of the C terminus helix with the β propeller domain of the α subunit has been demonstrated to be the mechanism by which cells control the activity of integrins through "inside-out" signaling [Luo et al. 2007]. I domains can be engineered to remain locked in the "open" or "closed" conformation by appropriate point mutations [Lu et al. 2001; Shimaoka et al. 2001; Tulla et al. 2008]. The binding pattern of such engineered I domains is less specific compared to the wild-type I-domains. The I domain α_LI of the α_L integrin subunit has also been crystallized in a third distinct "intermediate" conformation [Shimaoka et al. 2003], the stability of which was confirmed computationally by molecular dynamics [Jin et al. 2004]. The same computational model predicted that intermediate conformations do not exist in α_1I and α_2I I domains.

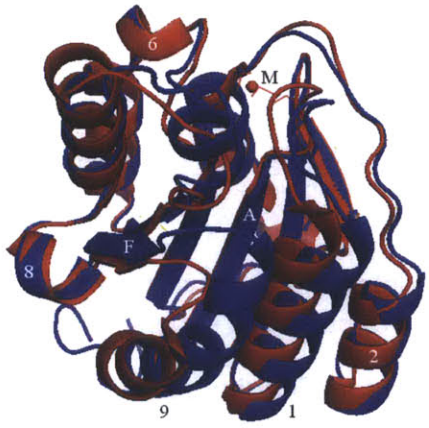


Figure 5.2.5: Crystal structures of the two conformations of the α_2 I domain. Blue: "closed" conformation (structure 1AOX [Emsley et al. 1997]). Red: "open" conformation (structure 1DZI [Emsley et al. 2000]). Superposition of the two structures highlights the large conformation change in the C-terminus α helix (9) and the MIDAS site loops. Structures are rendered using Pymol 1.0r1 using structures downloaded from the Protein Data Bank (PDB) database (http://www.rcsb.org).

BIACORE data suggest that 20% to 40% of native α_LI are in "open" configuration (estimate is based on the maximum binding values of wild type versus constitutively open α_LI) [Shimaoka et al. 2001; Vorup-Jensen et al. 2007]. One explanation is the exposure of a hydrophobic amino acid (F302) in the "open" configuration, which is hidden in the "closed' conformation. Recently, it was found that Glu336 in α_2I regulates switching of α_2I into the "active" configuration through interaction with the β_1 subunit in inside-out signaling [Connors et al. 2007].

The different affinity of the "closed" and "open" configurations of I domains for their ligands can be explained by the stabilization of the divalent cation in the MIDAS site. In the "closed" conformation, the divalent cation forms two water-mediated bonds with two serine residues located at loop L1, and interacts with two (negatively charged at neutral pH) aspartates, one at loop L1 and one at loop L3. The stabilization of the cation by two negatively charged aspartates of the I domain (without contribution from the ligand) is believed to be the cause of the low affinity of the "closed" configuration for its ligands. In the "open" conformation, due to the translocation of the cation and the three MIDAS loops, interaction with the L3 aspartate is lost, and the divalent cation can be stabilized by the aspartate of loop L1 plus an acidic amino acid (D or E) provided by the ligand [Lee et al. 1995; Emsley et al. 2000]. These findings agree with experimental results of an inverse relationship between the affinity of an I domain conformation for its ligands and its affinity for the divalent cation [Vorup-Jensen et al. 2007].

PDB file	Description	Reference
1JLM	α_MI (closed), Mn^{2+}	Lee et al. 1995
1IDO	α_MI (open), Mg^{2+}	Lee et al. 1995
1LFA	α_LI (closed), Mn^{2+}	Qu et al. 1995
1MJN	α_LI (intermediate), Mn^{2+}	Shimaoka et al. 2003
1MQ9	α_LI (open), Mn^{2+}	Shimaoka et al. 2003
1QC5	α_1I (closed), Mg^{2+}	Rich et al, 1999
1PT6	α_1I (closed), Mg^{2+}	Nymalm et al. 2004
1QCY	α_1I (closed), Mg^{2+}	Nymalm et al. 2004
1CK4	α_1I (closed)	Nolte et al. 1999
4A0Q	α_1I E317A (open)	Lahti et al. 2011
1AOX	α_2I (closed), Mg^{2+}	Emsley et al. 1997
1DZI	α_2I (open) + GFOGER- peptide, Co^{2+}	Emsley et al. 2000

Table 5.2-2: Available crystal structures of I domains in the RCSB Protein Data Bank (http://www.pdb.org).

5.2.5 Binding of CBI I Domains to Collagen

This section describes in more detail the mechanism utilized by CBI I domains to bind to collagen, and provides an overview of published experimental results.

Molecular Mechanism of CBI I Domain Binding to Collagen

CBI integrins bind to small peptide motifs ("adhesion ligands") on ECM proteins via the I domain of their α subunit. Each CBI I domain binds to several kinds of adhesion ligands found on a particular subset of collagens or other ECM molecules. The affinity and specificity of each CBI I domain for its ligands can be explained by the molecular details of I domain-ligand binding.

The crystal structure of α_2I bound to a triple-helical peptide that contains the GFOGER motif

(one of the major adhesion ligands for α_1I and α_2I) provides insight on how I domains bind their ligands [Emsley et al. 2000]. The acidic negatively-charged E residue of GFOGER protrudes into the MIDAS site of the I domain, where it coordinates a divalent cation (Co^{+2}) together with several residues of the MIDAS site. This coordination creates the main strength of the I domain - ligand bond. In addition, the F and R residues of the GFOGER motif participate in hydrophobic and ionic interactions with a nonpolar residue (Q215) and an acidic amino acid (D219) respectively on the surface of the I domain. Furthermore, several α_2I residues around the MIDAS site (N154, D219, E258, Y157) are complementary to the surface of collagen [Kamata et al. 1999; Emsley et al. 2000; Smith et al. 2000]. Although the MIDAS residues that coordinate the divalent cation are highly conserved among CBI I domains (Figure 5.2.3), the residues that lie on the I domain surface close to the MIDAS vary among CBI I domains. This variation is believed to control the specificity and affinity of each CBI I domain for its ligands [Emsley et al. 2000]. For example, when the negatively charged D219 of α_2I is substituted with the positively charged R (present in α_1I, and α_{10}I) the result has binding specificity to collagens similar to α_1I (binds stronger to collagens IV, VI compared to type I) [Tulla et al. 2001].

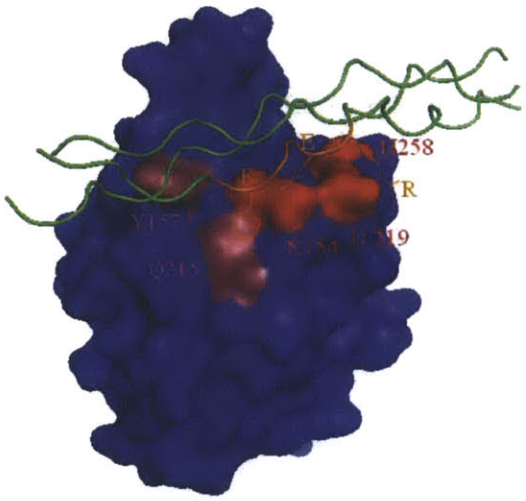


Figure 5.2.6: Residues of the α_2 I domain critical for high-affinity binding to GFOGER ligand in fibrilar collagens. The α_2 I domain (blue, red) binds a triple helical peptide (green, orange) that contains the GFOFER motif (orange). Based on structure 1DZI [Emsley et al. 2000] rendered using Pymol 1.0r1.

In some collagens (including types I, II) the GFOGER motif is present in two chains of the collagen triple helix. These two locations of GFOGER are situated close enough so that when an I-domain binds one GFOGER motif, two residues (F, R) of the neighboring GFOGER interact with residues around the MIDAS site [Emsley et al. 2000]. The residues of helix 6 are not crucial for ligand binding specificity and affinity as it was originally proposed [Kamata et al. 1999]. I domain – collagen binding can also be regulated by collagen post-translational modification. Specifically, is has been shown that prolyl hydroxylation in the hydroxyproline residue of the GFOGER motif is required for α_1I but not α_2I binding to collagen [Perret et al. 2003]. Apart from collagen, CBI can bind other ECM proteins. α_1I, α_2I, and α_{10}I bind laminin-1, while α_2I binds tenascin. α_1I and α_2I can discriminate different laminin isoforms.However CBI cannot bind fibronectin or fibrinogen [Tuckwell et al. 1995; Calderwood et al. 1997; Tulla et al. 2001; Tulla et al. 2008].

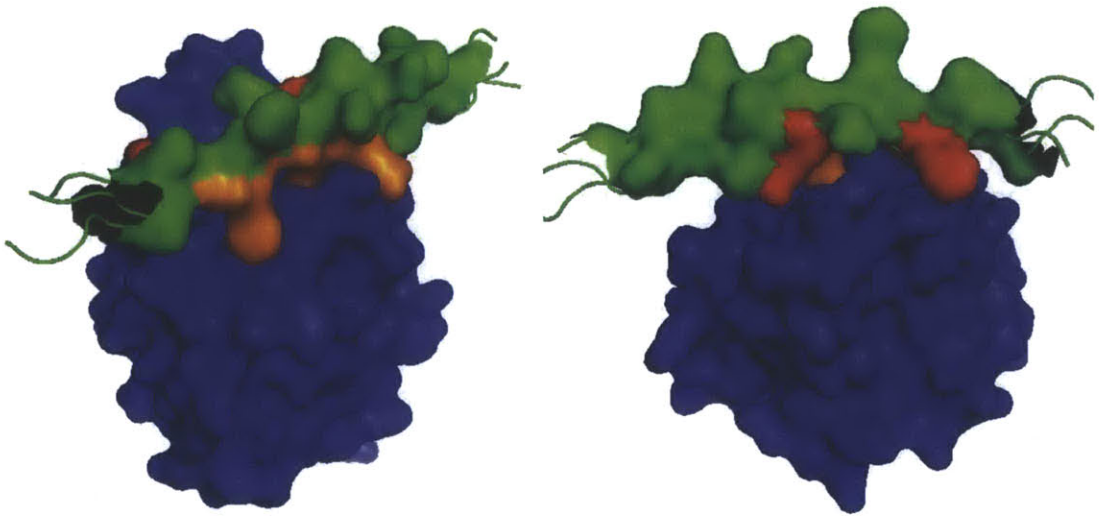


Figure 5.2.7: Surface complementarity between the α_2I domain (blue) and a triple helical peptide (green, red, orange) whose chains contain the GFOGER ligand. The GFOGER motif where α_2I binds is shown in orange, the residues (F, R) of the neighboring GFOGER motif that participate in surface complementarity are shown in red. Based on structure 1DZI [Emsley et al. 2000] rendered using Pymol 1.0r1.

Ligands of CBI I Domains in Collagens

The GFOGER motif was the first ligand of α_1I, α_2I, identified in collagen I [Knight et al. 1998, 2000]. Later it was found that GFOGER is also a ligand for α_{11}I [Zhang et al. 2003; Siljander et al. 2004]. Rotary shadowing experiments revealed that in tropocollagen molecules there are multiple ligands. CBI I Domain ligands in collagen I are located mostly **in** three sections (N-terminus, C-terminus, middle) of the ~300nm long collagen molecule [Xu et al. 2000]. In tropocollagen III the ligand close to the C terminus dominates [Kim et al. 2005]. Additional research revealed several additional ligands of CBI I domains. Particular substitutions of the F residue of motifs GLOGER and GROGER have been identified as ligands of both α_1I and α_2I [Xu et al. 2000; Kim et al. 2005; Raynal et al. 2006]. The motif GASGER, found in human collagens I, II and III is also a ligand for integrins $\alpha_1\beta_1$ and $\alpha_2\beta_1$, however of lower affinity than GxOGER ligands [Xu et al. 2000]. The proposed motif GAOGER was found to bind α_1I and α_2I with very low affinity [Kim et al. 2005]. The motif GLOGEN found in human collagen III binds both $\alpha_1\beta_1$ and $\alpha_2\beta_1$ and has been found to be a better ligand for $\alpha_1\beta_1$ compared to GFOGER [Raynal et al. 2006; Hamaia et al. 2012]. Finally the motif GVOGEA has been identified as a specific ligand for $\alpha_1\beta_1$ [Hamaya et al. 2012]. Binding of recombinant α_1I, α_2I, and α_{11}I to ligand-containing peptides revealed the following affinity strength sequence: GFOGER>GLOGER> GLSGER>GMOGER> GAOGER≈GASGER [Siljander et al. 2004]. Interestingly, mutant α_2I locked in the "open" configuration have similar affinity for peptides containing GFOGER, GLOGER, GLSGER, or GMOGER ligands, still significantly larger than GAOGER and GASGER ligands. Competitive binding experiments verified that CBI I domains bind to common ligands in collagens. For example the GFOGER and GLOGER motifs are ligands of α_1I, α_2I and α_{11}I [Knight et al. 1998, 2000; Xu et al. 2000; Zhang et al. 2003; Siljander et al. 2004].

CBI I Domain ligands are present in many collagen isoforms (Figure 5.2.8). GFOGER is present in collagens I, II, IV, V, VII and XI, GLOGER is present in collagens I, II, III, VII, and VIII, while GROGER is present in collagens I, III, VII, and X [Xu et al. 2000; Siljander et al. 2004; Kim et al. 2005]. Furthermore, each collagen isoform contains multiple copies of different ligands. For

example, each collagen I molecule contains 2 GFOGER, 4 GLOGER, 2 GMOGER, and 2 GASGER ligands, while collagen III contains 3 GMOGER, 3 GLSGER, 3 GMOGER, and 6 GAOGER ligands [Siljander et al. 2004].

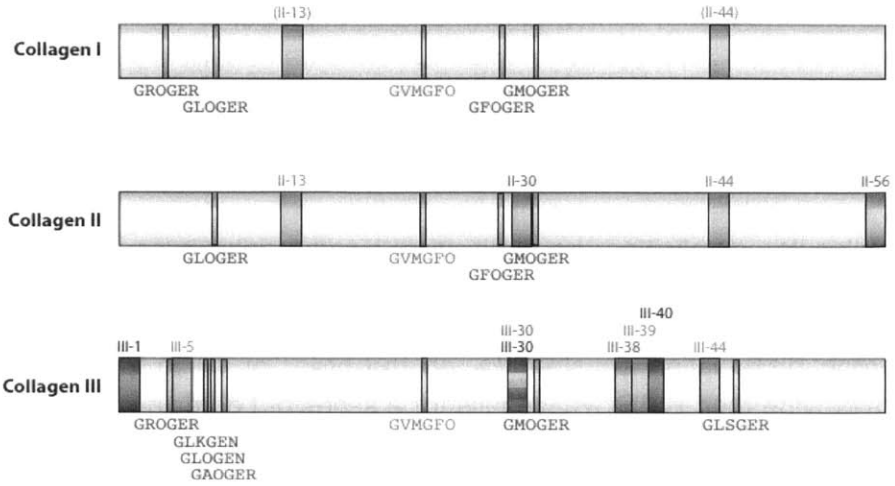


Figure 5.2.8: Ligands of collagen-binding integrins (highlighted blue) and DDR family receptors (highlighted orange) in collagens I, II and III [Leitinger 2011].

Specificity and Affinity of CBI I Domain Binding to Collagens

Studies on the binding specificity of CBI I domains to ECM molecules rely mostly on two kinds of binding assays: i) solid phase binding assay (SPBA), and ii) BIACORE binding assay. The vast majority of published studies focus on interactions of soluble recombinant α_1I and α_2I with tropocollagen molecules (usually type I collagen). Few studies consider α_{10}I and α_{11}I. A single study focused on binding of I domains to collagen I molecules in fiber form. No study so far has quantified the binding of CBI to ECM tissue or biomaterials.

Although there is great variation in experimental findings, most publications suggest that α_1I bind network-forming collagens more strongly than fibril-forming collagens, while α_2I bind fibril-forming collagens more strongly than network-forming collagens [Tuckwell et al. 1995; Calderwood et al. 1997; Dickeson et al. 1999; Tulla et al. 2001, 2008; Jokinen et al. 2004; Shi et. 2012]. This pattern agrees with observations of cells expressing $\alpha_1\beta_1$ versus cells expressing $\alpha_2\beta_1$ [Dickeson et al. 1999; Nykvist et al. 2000]. α_{10}I bind network-forming collagens slightly better than fibril forming collagens and resemble slightly the pattern of α_1I [Tulla et al. 2001]. α_{11}I bind preferably to fibril-forming collagens in a similar way to α_2I [Zhang et al. 2003]. I domains that have been engineered to stay in "open" conformation show much less specificity (i.e. bind to a wider range of ECM molecules) and greater avidity (bind to more binding sites with somewhat tighter dissociation constants) [Tulla et al.2008].

Although studies of CBI I domain binding to collagen use tropocollagen I molecules, in tissues collagen molecules form super-molecular structures such as fibrils, fibers and mesh-like structures [Gelse et al. 2000]. The single study that has focused so far on the effects of fibril formation on I domain adhesion to collagen, revealed that fibril formation reduced the affinity of α_1I and α_2I binding to collagen and reduced the number of binding sites (ligands) available for binding [Jokinen et al. 2004].

Published estimates of the dissociation constant k_D for the affinity of CBI I domain to collagen vary widely, see Table 5.2-3. Possible reasons that can explain this variance are: i) different experimental methods (SPR, SPBA), ii) different I domains expressed and purified in different ways, iii) collagen state (amount, conformation), and iv) improper analysis and interpretation of experimental data, see also the discussion on BIACORE data nanalysis (Section 5.3.5), and the discussion on binding assays on surfaces (Section 5.3.8). Few papers have highlighted the difficulty in interpreting these data. Among them, one study suggests that results from CBI I domain – collagen binding experiments do not agree with 1:1 binding models or cooperative binding models, and that the presence of multiple adhesion ligands for each CBI I domain on each collagen molecule complicates the analysis of binding data, because I domains bind to each ligand with different affinity [Rich et al. 1999]. Most SPBA assays suggest k_D on the order or 200 nM. Most BIACORE assays fit the curves using a bi-exponential model, which usually provides one strong dissociation constant k_D on the order of 10 to 200 nM. However few published papers provide their BIACORE results for a series of I domain concentrations [Rich et al. 1999]. Estimations of k_D based on kinetic analysis need to agree reasonably with steady-state binding results from titration experiments. However, most published papers provide few curves which cannot support dissociation constants on the order of 100 nM [Calderwood et al. 1997; Xu et al. 2000].

I-domain	Collagen isoform (source)	k_D [nM]	k_{on} [$M^{-1}s^{-1}$]	k_{off} [10^{-5} s^{-1}]	Assay	Reference
α_1I	I (rat)	24 23	4.1E3 -	8.2	SPR (k) SPBA	Calderwood et al. 1997
α_1I	I (bovine)	90*, 7000			SPR (eq)	Rich et al. 1999
α_1I	I (bovine)	110			SPBA	Xu et al. 2000
α_1I	I (chic)	260,13900			SPR (eq)	
α_1I	I (rat)	160			SPBA	Tulla et al. 2001
α_1I	I (rec. human)	320 5500			SPR (eq)	Kim et al. 2005
GST-α_1I	I (bovine)	30			SPBA	Jokinen et al. 2004
GST-α_1I	I fibrils (bovine)	250			SPBA	
GST-α_1I	I (rat)	268			SPBA	Tulla et al. 2008
α_1I	II (rec. human)	150, 7280			SPR (eq)	
α_1I	III (rec. human)	190, 6150			SPR (eq)	Kim et al. 2005
α_1I	IV (human)	6 34	12E3	7.3	SPR (k) SPBA	Calderwood et al. 1997
α_1I	IV (mouse)	65			SPBA	Tulla et al. 2001
α_1I	VI (human)	200			SPBA	
α_2I	I (rat)	180 54	67E3	120	SPR (k) SPBA	Calderwood et al. 1997
α_2I	I (bovine)	10700 5800			SPR (eq) SPBA	Xu et al. 2000
α_2I	I (chicken)	8800			SPR (eq)	
α_2I	I (rat)	20			SPBA	Tulla et al. 2001

α_2I	I (rec. hum)	260, 3990			SPR (eq)	Kim et al. 2005
GST-α_2I	I	500*			SPBA	Kamata and Takada 1994
GST-α_2I	I (bovine)	10			SPBA	Jokinen et al. 2004
GST-α_2I	I fibrils (bovine)	100			SPBA	
GST-α_2I	I (rat)	31			SPBA	Tulla et al. 2008
α_2I	II (rec. human)	1750,16500			SPR (eq)	
α_2I	III (rec. human)	330 14500			SPR (eq)	Kim et al. 2005
α_2I	IV (human)	115			SPBA	Calderwood et al. 1997
α_2I	IV (mouse)	140			SPBA	
α_2I	VI (human)	-			SPBA	Tulla et al. 2001
α_{10}I	I (rat)	350			SPBA	
α_{10}I	IV (mouse)	300			SPBA	
α_{10}I	VI (human)	350			SPBA	

Table 5.2-3: Experimental results for binding parameters reported in the literature for the binding of recombinant I domains (with or without a GST fusion) to collagen molecules (type I, II, III, IV or VI). The estimated equilibrium parameters are the dissociation constant k_D and the number of binding sites n per molecule. The estimated kinetic parameters are the on rate k_{on} and the off rate k_{off} of the binding reaction. Binding parameters are estimated from solid phase binding assays (SPBA) or surface plasmon resonance (SPR) assays. SPR is implemented either in kinetics mode (SPR(k)) or in equilibrium mode (SPR(eq)). *: question results due to large errors between best fit and data.

The Effect of Collagen Conformation on CBI I Domain binding

The molecular mechanism of I domain-ligand binding suggests that proper collagen conformation (triple helix) is required for strong I domain-ligand binding. This has been validated experimentally. α_2I cannot bind heat denatured (gelatinized) fragments of collagen or cleaved collagen molecules [Tuckwell et al. 1995; Messent et al. 1998]. α_1I and α_2I bind collagen with approximately 10 times higher affinity than gelatin [Tuckwell et al. 1995; Calderwood et al. 1997]. α_1I and α_2I bind to peptides that contain the GFOGER ligand as long as they form triple helical quaternary structure [Knight et al. 2000]. Collagen cleavage *in vivo* by MMPs has been proposed as a signaling mechanism because it changes the set of integrins utilized by cells to bind to their environment (cells bind gelatinized collagen by a distinct set of adhesion receptors, e.g. $\alpha_v\beta_{v3}$ [Messent et al. 1998]). For example, collagen degradation in the early steps of wound healing has been proposed as a way to enhance cell motility by preventing the formation of extensive interactions through $\alpha_2\beta_1$ integrins [Ruoslahti 1992; Pilcher et al. 1997; Messent et al. 1998].

Divalence Cation Dependence of CBI I Domain Binding to Collagens

The affinity of α_1I, α_2I, and α_{10}I binding to their ligands is dependent on divalent cations [Tuckwell et al. 1995; Calderwood et al. 1997; Tulla et al. 2001]. Experimental evidence from BIACORE experiments, suggests that divalent cation (Mg^{+2} or Mn^{+2}) binding to I domain is a prerequisite for I domain binding to its ligand. [Rich et al. 1999] suggest that excluding Mg^{+2} reduces I domain binding to collagen significantly. The presence of the chelating agent EDTA reduces binding of α_1I and α_2I to collagen by 50% to 90% [Estavillo et al. 1999; Dickeson et al. 1999; Tulla et al. 2008; Hamaia et al. 2012; Shi et al. 2012]. Some papers claim that EDTA

blocks completely the binding of α_1I and α_2I to collagen [Calderwood et al. 1997; Messent et al. 1998], or to GROGER peptides [Kim et al. 2005]. However, the effect of EDTA on cell adhesion is more severe compared to its effect on I Domain binding. EDTA blocks almost completely several cell adhesion assays [Siljander et al. 2004; Hamaia et al. 2012].

On the other hand, although Ca^{+2} affects integrin-mediated adhesion, there is conflicting evidence whether it affects I Domain binding to collagen [Tuckwell et al. 1995; Estavillo et al. 1999; Onley et al. 2000] or to GFOGER peptides [Onley et al. 2000].

It is proposed that the affinity of the I Domain MIDAS site for the divalent cation is inversely related to the addinity of I Domains for their adhesion ligands. The reported dissociation constant of α_1I for Mg^{+2} is 10 μM [Rich et al. 1999]. The dissociation constants of wild-type α_LI for divalent cations depend on the cation type (3.16 μM for Mn^{+2}, 19.42 μM for Mg^{+2}, 384 μM for Ca^{+2}). This theory is supported by the finding that constitutively "active" I domains bind ligands with 2-3 orders of magnitude increased affinity and bind cations with 1 order of magnitude reduced affinity compared to "closed" wild type I domains [Vorup-Jensen et al. 2007]. In the same study, the number of binding sites per collagen molecule in BIACORE experiments was shown to depend on the concentration of both Mg^{+2} and I domains.

5.2.6 Cellular Effects of Adhesion via Collagen-Binding Integrins

General Effects of Integrin-Mediated Adhesion to Collagen

Even though integrins $\alpha_1\beta_1$ and $\alpha_2\beta_1$ are structurally similar, they have been found to induce different effects to cells. $\alpha_1\beta_1$ has been found to promote cell proliferation and to impede collagen synthesis, while $\alpha_2\beta_1$ inhibits the growth of some cell types and increases ECM production and remodeling [Jokinen et al. 2004; Shi et al. 2012].

Cell contraction has been generally associated with integrin $\alpha_2\beta_1$. CHO cells (normally expressing neither $\alpha_1\beta_1$ nor $\alpha_2\beta_1$) transfected stably with $\alpha_2\beta_1$ contract floating collagen matrices more than CHO cells transfected with $\alpha_1\beta_1$ [Jokinen et al. 2004]. Furthermore $\alpha_2\beta_1$ has been shown to promote (but not to be necessary for) the formation of characteristic long protrusions in cells cultured on fibrillar collagen (but not on tropocollagen) [Jokinen et al. 2004]. It is obvious that the molecular organization of collagen affects cell phenotypes, perhaps by affecting the ability of different integrins to bind collagen.

Adhesion via CBI Affects Several Cell Phenotypes Related to Wound Healing or Cancer

A large number of studies provide evidence of a correlation between the amount and type of integrin-mediated adhesion to the surrounding matrix of a cell, and the expression of particular cell phenotypes (adhesion, ECM remodeling, cell contraction, proliferation, migration) related to important biological processes such as wound healing and cancer.

During wound healing, growth factor patterns which include PDGF and TGFβ change the expression pattern of CBI in fibroblasts: $\alpha_1\beta_1$ is down-regulated and $\alpha_2\beta_1$ is up-regulated [Heino 2000]. In general, there is evidence that $\alpha_1\beta_1$-mediated adhesion to collagen promotes cell growth and suppresses ECM remodeling [Shi et al. 2012], while $\alpha_2\beta_1$-mediated adhesion stimulates ECM remodeling by up-regulating synthesis of collagen I and MMPs [Riikonen et al. 1995; Heino 2000; Tulla et al. 2001]. $\alpha_1\beta_1$ and $\alpha_2\beta_1$ seem to compete with each other [Riikonen et al. 1995; Heino 2000] in order to fine-tune ECM remodeling and contraction.

In vitro studies of collagen gel contraction by fibroblasts show that gel contraction is mediated entirely by collagen-binding integrins ($\alpha_x\beta_1$ integrins excluding $\alpha_3\beta_1$ and $\alpha_5\beta_1$) and that blocking $\alpha_2\beta_1$ through anti-α_2 antibodies results in 50% less contraction [Schiro et al. 1991]. The same study shows that, even though $\alpha_2\beta_1$ adhesion could be mediated by α_2 subunits lacking a cytosolic domain, contraction required the cytosolic domain of α_2 subunits, i.e. required interaction of α_2 with cytosolic proteins. Even though evidence suggests that $\alpha_2\beta_1$ seems is a key feature of ECM contraction, it is not the only CBI that mediates contraction [Schiro et al. 1991; Heino 2000]. A different study showed that stellate liver cells expressed $\alpha_1\beta_1$ and not $\alpha_2\beta_1$ before and after (up to 6 days) full-thickness injury [Racine-Sampson et al. 1997]. Adhesion and contraction of stellate cells harvested 6 days post-injury to collagen I and IV gels was mediated by $\alpha_1\beta_1$.

Although the $\alpha_2\beta_1$ integrin is the major integrin found in epithelial cells, re-epithelization of full-thickness wounds in α_2-null mice is not delayed, even though harvested keratinocytes from such animals failed to adhere in collagen I gels [Grenache et al. 2007]. In the same animal model, the mRNA expression of several matrix metalloproteinase (MMP) genes is upregulated in α_2-null mice compared to wild-type mice.

The upregulation of matrix remodeling genes by $\alpha_2\beta_1$ makes $\alpha_2\beta_1$ a protein that could promote invasive cancer cell phenotypes, therefore a potential target for anti-cancer studies [Riikonen et al. 1995]. A 3D *in vitro* model of pancreatic cancer cell adhesion and proliferation inside collagen-GAG scaffolds (CGS, similar to the ones used in this thesis) showed that $\alpha_2\beta_1$ was necessary for pancreatic cancer cell adhesion and subsequent proliferation [Grzesiak and Bouvet 2007].

The vast majority of studies on the effects of CBI on wound healing have focused on integrins $\alpha_1\beta_1$ and $\alpha_2\beta_1$. Table 5.2-1 provides a partial list of cell lines used in experiments of adhesion via CBI. Different cell lines have different patterns of CBI expression. Apart from cells that naturally do not express some CBI, it is possible to use cells obtained from knock-out mice that lack ITGA1 or ITGA2 genes [Shi et al. 2012]. The effects of adhesion via the remaining two CBI ($\alpha_{10}\beta_1$ and $\alpha_{11}\beta_1$) are much less known.

Cell Signaling Induced by CBI-Mediated Adhesion

Integrins are birectional signaling machines that transmit chemical information in both "outside-in" and "inside-out" ways [Hynes 2002]. In "outside-in" signaling, binding of an integrin to an adhesion ligand in the ECM changes the activation state of the integrin, which is transmitted to the integrin cytosolic domain through conformational changes. Homologous subunits of activated integrins cluster [Li et al. 2003] and interact with cytosolic signaling proteins to induce intracellular signal transduction pathways that inform the cell about its environment. In "inside-out" signaling, integrins interact with cytosolic proteins that regulate their activation state and therefore the type and amount of cell-matrix interactions [Hynes 2002].

The large conformation changes that occur during I domain activation are involved in both "outside-in" and "inside-out" integrin signaling [Lee et al 1995; Emsley et al. 2000; Hynes 2002]. Binding of an I domain to its ligand favors transition into the "open" conformation, which is propagated to the rest of the integrin through the motion of the I Domain C-terminus helix, and eventually activates the integrin by separating its two subunits. It is believed that integrin activation (separation of its subunits) leads to conformational changes that lead to vertical

displacement of the C helix, which favors the "open" conformation of the I domain (inside-out signaling).

Cell adhesion through integrins has been shown to be very important for cell survival and proliferation. Many kinds of cells survive only when they adhere to an insoluble matrix. This requirement has been shown to be implemented in two ways [Gomperts et al. 2002]. The first is through protein kinase B (PKB/Akt): Activated integrins recruit focal adhesion kinase (FAK) that autophosphorylates, and activates PKB through phosphatidylinositol-3 kinase (PI-3). PKB phosphorylates and inactivates caspaces and several transcription factors related to apoptosis. The second process involves the mitogen-activated protein kinase (MAPK) pathway: Experiments in cells grown inside 3D collagen matrices show that $\alpha_2\beta_1$ integrins excite the p38 MAPK pathway through the Cdc42 GTPase and kinases MEK3/4 [Heino 2000]. These events take place only inside 3D collagen matrices and are not observed when the cells are cultured in monolayers. The ERK/MAPK signaling has been also involved in the downregulation of collagen IV expression induced by adhesion to collagen IV through $\alpha1\beta1$ [Shi et al. 2012]. Finaly, $\alpha2\beta1$ has been also shown to induce MMP1 through PKC and NF-κB pathways [Heino 2000].

Activated integrins have been shown to interact with several GTPases by first binding and activating FAK, and then recruiting adaptor proteins (e.g. SOS, GRB2) that interact and activate GTPases. Ras GTPase is involved in the MAPK pathways described above. Rho family GTPases (Rac, cdc42, Rho) are involved in regulating cell cytoskeleton (actin and microtubule polymerization) and cell contraction [Lodish et al. 2007]. *In vitro* experiments on cells cultured on 2D glass surfaces suggest that each kind of Rho GTPase favors the formation of different actin structures: Rac regulates lamellipodia formation, cdc42 regulates filopodia formation and Rho regulates stress fiber formation [Lodish et al. 2007]. Finally, Rho GTPase participates in cell contraction by directly phosphorylating (and therefore inducing contraction) the myosin light chain, or by activating the MCI-17 inhibitor of myosin phosphatase.

Integrins are expected to interact with multiple pathways. It was shown above that integrins excite the MAPK pathway and therefore should cross-talk with other receptors that excite the MAPK pathway such as receptor tyrosine kinases (RTK, ligands include the NGF, PDGF, FGF, EGF growth factors) or cytokine receptors (ligands include interferons). MAPK pathway is involved in activating several transcription factors that are related to cell proliferation [Lodish et al. 2007], as well as in the control of cyclins, proteins that control cell cycle progression [Gomperts et al. 2002]. Integrins are also expected to cross-talk with other receptors that excite Rho family GTPases, such as RTK and G-protein coupled receptors (GPCR). Furthermore, both the MAPK pathway and Rho family GTPases have been shown to affect the TGβ signal transduction pathway.

Even though the binding patterns of the four collagen-binding integrins (CBI) overlap, it has been shown that each one of the CBI has different effects to the cell, therefore it is possible that each CBI triggers different signaling events [Heino 2000]. However, the available information about the signaling events that are triggered by each one of the four CBI is limited. Furthermore, most of the available information is based on *in vitro* experiments in cells cultured on stiff 2D surfaces. Less information is available about CBI-induced signaling in cells cultured inside more physiologically-relevant 3D microenvironments such as collagen-GAG scaffolds.

5.2.7 Methods for Quantifying the Surface Chemistry of Biomaterials

Current Practice of Reporting the Surface Chemistry of Matrices

The majority of published studies on cell–matrix interactions were carried out on planar surfaces (petri dish, flasks) where interactions with matrix biomolecules (such as peptides or ECM molecules) take place by adsorbing (or less often crosslinking) these molecules on the surface. Published papers hardly ever report the surface chemistry of the corresponding insoluble environment; instead they provide estimates of the amount of biomolecule adsorbed on the surface, expressed in units of biomolecule mass/area [Holub et al. 2003; Jirouskova et al. 2007]. Few researchers have measured the mass of the biomolecule that is adsorbed or cross-linked on a surface. This was accomplished by using appropriately labeled biomolecules, and detecting the signal emitted by the biomolecule after adsorption/crosslinking, which is then converted into biomolecule surface density (mass/area) using a calibration curve. Examples include radiolabelled fibronectin [Maheshwari et al. 2000; Valenick and Schwarzbauer 2006] and fluorescently labeled collagen [Engler et al. 2004]. Nevertheless, a measurement of biomolecule surface density is not necessarily proportional to the density of adhesion ligands available to cells for binding, therefore these methods do not describe accurately the surface density felt by cells.

Over the past ten years, an increasing number of researches utilize 3D in vitro cell culture systems, such as gels or porous biomaterials [Griffith and Swartz 2006]. Again, none of these studies reports the surface chemistry of the corresponding matrix, and instead describe the chemistry of the matrix felt by cells using bulk properties such as mass fraction and chemical composition.

Methods for Quantifying Adhesion Ligands in Biomaterials

Two recently developed methods measure quantities that are related to the density of adhesion ligands in biomaterials. Both of them are applied in artificial biomaterials that contain only RGD ligands.

The first method was developed for quantifying the density of RGD ligands in artificial biomaterials made by functionalizing biologically inert surfaces with RGD peptides [Barber et al. 2005; Harbers et al. 2005]. This method works by attaching the RGD entity to a fluorophore through a linker that contains a protease recognition site. Treatment of the material with the protease chymotrypsin releases the fluorophores. The detected fluorescence signal emitted by the released fluorophores is converted to RGD density via a calibration curve.

The second method quantifies the number of bound RGD ligands on cell adhesion receptors [Kong et al. 2006; Huebsch & Mooney 2007; Hsiong et al. 2008]. In this method cells, whose membranes are stained with fluorescein, are incubated inside a gel of alginate molecules attached to peptides that contain RGD ligands next to a rhodamine dye. The number of cell adhesion receptors bound to fluorescently labeled RGD peptides is estimated by detecting the amount of Förster resonance energy transfer (FRET) between fluorescein and rhodamine. FRET takes place when the distance between the two fluorophores is less than approximately 8 nm, a condition that occurs in the proximity of an RGD ligand bound to a cell adhesion receptor. The FRET measurement is converted into number of RGD peptides bound per cell through a calibration curve. Such measurements can take place in high-magnification optical configurations in order to quantify cell adhesion to RGD in single cell level [Kong et al. 2006].

The discussion of this section suggests that there is a lack for methods that could quantify the surface density of adhesion ligands of particular cell adhesion receptors in a wider range of biomaterials or tissue ECM.

The need for methods to quantify the surface chemistry of a matrix

At the moment, there is considerable effort to study the extracellular matrix of cells using methods of biochemistry and proteomics (e.g. mass spectroscopy). This is not an easy task, due to several biochemical properties of ECM macromolecules (large size, extensive inter-molecular cross-linking between ECM molecules, poor solubility, disulfide bridges) [Naba et al. 2012]. Furthermore, studying bio-molecules in the insoluble state is much more complex compared to studying bio-molecules in solution. In addition to the chemical composition and its modifications, the insoluble state is described by a larger number of parameters including its stiffness and stress tensors, and surface chemistry.

At the moment there exist several techniques that can characterize the chemistry on the surface of biomaterials or protein-coated surfaces [Ma et al. 2007; Kingshott et al. 2011]. Several spectroscopic techniques such as Attenuated Total Reflectance Fourier transform spectroscopy (ATR-FTIR), X-ray Photoelectron Spectroscopy (XPS) and time-of-flight secondary ion mass spectroscopy (TOF-SIMS) can quantify the chemical structure on the surface of a biomaterial. ATR-FTIR is an alternative configuration of IR spectroscopy used in monomolecular layer analysis. XPS measures elemental composition, empirical formula and chemical state. TOF-SIMS is mass spectroscopy technique that analyzes the elemental and chemical composition of the outermost molecular or atomic layer of a solid surface. Several colorimetric methods measure the surface density of particular chemical groups such as carboxylic groups, primary amines (Rhodamine 6G, methyl orange), or the density of proteins on biomaterial surfaces (Ninhydrin, bicinchoninic acid). Ellipsometry is utilized for measuring the thickness of protein coatings. Some of these methods have been applied not just in characterizing functionalized surfaces, but also in characterizing the ECM of decellularized organs [Brown et al. 2010], an actual 3D matrix of significant biological activity.

Although the presence of particular chemical groups on the surface of a material has been found to affect cells *in vitro* [Keselowsky et al. 2005], the biological relevance of the methods described in the previous paragraph is limited by their inability to quantify the surface chemistry from the point of view of cell biology, where what matters is not the presence of particular elements or functional groups, but instead the presence of chemical entities (adhesion ligands)

recognized by cell adhesion receptors. Specifically, there is a severe shortage of methods that can quantify the presence and density of the elementary chemical clues that cells recognize and bind to *in situ* on the surface of actual 3D matrices. Such methods could have significant impact in quantifying the stimuli provided by the matrix to cells, and in characterizing and designing new biomaterials.

This chapter describes a method that combines fluorescent protein technology and spectral multi-photon microscopy to measure *in situ* the density of ligands for particular adhesion receptors in three-dimensional matrices (such as the connective tissue of organs or biomaterials). The method attempts to quantify the surface chemistry of a matrix as detected by cells, and fills a critical gap in cell–matrix interaction studies and biomaterial development. Although the results presented in this study focus on quantifying the density of ligands for the two major collagen-binding receptors (integrins $\alpha_1\beta_1$ and $\alpha_2\beta_1$) on collagen-based biomaterials (similar to collagen-based scaffolds used in [Soller et al. 2012]), the proposed method can be appropriately generalized to quantify the density of ligands for several types of adhesion receptors in a wide variety of three-dimensional matrices.

5.3 Methods

5.3.1 Overview of the Methodology of Surface Chemistry Quantification

Key features

The proposed methodology for quantifying the surface chemistry of a matrix is an *in situ* binding assay, where 3D fluorescence microscopy is utilized to measure the signal emitted by fluorescent adhesion biomarkers bound on adhesion ligands on the surface of the biomaterial. This subsection summarizes two key features of this methodology (adhesion markers, 3D imaging):

The methodology quantifies the surface chemistry of a matrix by utilizing adhesion markers, soluble molecules that have similar binding properties (specificity and affinity) as the adhesion receptor of interest. Possible adhesion markers include soluble recombinant domains of adhesion receptors that mediate receptor adhesion to its ligands (provided it is feasible to identify and express these domains). Such soluble markers recognize and bind to ligands present on the surface of a matrix. For each adhesion receptor of interest, two kinds of adhesion markers are necessary:

- A fluorescent adhesion marker (denoted as I^*). The density of ligands for an adhesion receptor is quantified by measuring the fluorescence intensity emitted by soluble fluorescent adhesion markers bound to their ligands on the matrix surface.
- A non-fluorescent adhesion marker (denoted as I), which ideally is identical to the fluorescent adhesion marker but **is** non-fluorescent. Non-fluorescent markers are used in combination with fluorescent markers in competitive binding assays in order to evaluate the specificity and affinity of the fluorescent marker.

This study focuses on quantifying the surface chemistry of collagen biomaterials. This is implemented by quantifying the surface density of adhesion ligands for the two major collagen-binding integrins (CBI) $\alpha_1\beta_1$ and $\alpha_2\beta_1$. Adhesion markers are based on recombinant I domains of integrin subunits α_1 and α_2 (Section 5.2.4). The fluorescent adhesion marker of each CBI is the I

domain of the corresponding α subunit (ITGA1 or ITGA2) fused to a small tetracysteine tag (TC-α_1I, TC-α_2I). TC-tagged I domains are not intrinsically fluorescent, however they become fluorescent after a small biarsenical dye binds to the TC tag (Section 5.3.6). The non-fluorescent adhesion marker of each CBI is simply the native I domain (α_1I, α_2I) of the corresponding α subunit (Section 5.3.2).

Quantifying the signal emitted by a fluorescent molecule located on the surface of a biomaterial or ECM is a challenging problem due to the complex 3D geometry of the matrix. Biomaterials and ECM consist of super-molecular structures (fibers, struts, and lamellae) whose dimensions vary, and whose orientation can be random. In order to measure the surface chemistry of such a matrix *in situ* with minimal matrix perturbation, it is necessary to use a 3D imaging modality that can image deep inside the matrix and can resolve the matrix volume from its surroundings. In this thesis this is achieved using spectral multi-photon microscopy (see Chapter 2). Multi-photon microscopy provides the optical sectioning capability required for imaging complex 3D geometries inside thick matrix samples [Denk et al. 1990]. The spectral detection capability of the MPM (see Section 2.3.1) enables simultaneous imaging of the emissions generated by fluorescent components in the matrix and by the fluorescent I Domains (Chapter 2). In this thesis, spectral imaging is exploited to image the weak blue-green intrinsic fluorescence emission of collagen and the green fluorescent emission of the FlAsH biarsenical dye attached to TC-tagged I domains. Collagen emission is utilized to identify the matrix volume and guide the interpretation of the fluorescent I Domain emission.

Experimental Protocol Summary

Each experiment for the measurement of the surface density of adhesion ligands of a particular adhesion receptor on a matrix consists of the following steps:

- I Domain staining: fluorescently stain TC-tagged I Domains by treating them first with 1 mM TCEP reducing agent (to reduce the cysteines of the TC tag) and then with a slight excess (10%) of FlAsH biarsenical dye. Remove the excess unbound FlAsH dye by gel filtration.
- Matrix blocking: matrix samples are treated with a blocking agent (e.g. BSA) in order to reduce non-specific binding of adhesion markers to the matrix.
- Binding of I Domains to the matrix: remove the blocking buffer and add the fluorescently-labeled I domain solution in appropriate buffer (with or without Mg^{+2} ions). Incubate matrix samples overnight at 4°C to let the I Domains bind their adhesion ligands on the matrix.
- Imaging: image the matrix sample in binding equilibrium with I domains using a spectral multi-photon microscope. Acquire a 3D image, by imaging consecutive planes 10-30 μm away from the material surface at 1 μm increments.
- Image Processing: Apply the image processing algorithms described in Chapter 2 in order to segment the image into pixels representing the scaffold and pixels representing I Domain solution. The same algorithms also provide images of the fluorescent emission of collagen and the fluorescent emission of fluorescently-labeled I domains. Use the fluorescent emission of collagen to identify the matrix volume. Quantify the emission intensity of fluorescently-labeled I domains at the scaffold surface at appropriate locations. Transform the intensity measurements of the emission of fluorescently-labeled I Domains into surface density of fluorescently-labeled I Domains using a calibration curve. Derive the mean and standard deviation over six measurements for each experimental setup.

- Result Interpretation: combine results for the density of adhesion ligands obtained by treating the same matrix with different concentrations of fluorescently-labeled I domains and various concentrations of non-fluorescent I Domains. Use a 1:1 binding model to estimate the number of binding sites on the collagenous matrix and verify that the estimated dissociation rate k_D makes physical sense.

5.3.2 Recombinant Expression and Purification of CBI I domains

Four kinds of recombinant I domains were expressed in BL21 *E.coli* as GST fusion proteins and purified by GST affinity chromatography [Wilson and Walker 2000]. After purifying the fusion protein, the GST entity was removed by thrombin cleavage, I domains were reconstituted in appropriate buffer, and stored at 4°C until use. The purity and concentration of purified I domain solutions was evaluated via SDS-PAGE electrophoresis and 280nm light absorption spectroscopy (Section 5.3.3). Advanced characterization of purified I Domains by circular dichroism (CD) spectroscopy and BIACORE binding assay is described in Sections 5.3.4 and 5.3.5.

Preparation of I Domain cDNA Plasmids

I domains from the α subunits of the two major collagen-binding integrins (CBI) $\alpha_1\beta_1$, $\alpha_2\beta_1$ were expressed. The cDNA of each CBI I domain (from the α_1 and α_2 integrin subunits) was was amplified from the cDNA of corresponding *Homo Sapiens* α subunit gene (ITGA1 and ITGA2), and inserted in a pGEX4T3 vector (General Electric Healthcare, NY), appropriate for expressing GST fusion proteins. For plasmid amplification, plasmids were transfected in DH5α e-coli, and glycerol stocks of bacteria were stored for long-term at -80°C. More information about the genes and plasmids of each I domain cDNA is provided in Table 5.3-1.

Two I domains are non-tagged (α_1I, α_2I), and two I domains (TC-α_1I, TC-α_2I) are tagged with a teracysteine (TC) tag that enables their specific fluorescent labeling using biarsenical dyes [Adams et al. 2002], see Section 5.3.6. The tagged I domains are identical to the non-tagged, apart from the presence of the tetracystine tag WDCCPGCCK in the N terminus of the I domains.

I domain name	Gene	Accession	Vector	Restriction enzymes	TC-tagged
α_1I	ITGA1	P56199 (147-360)	pGEX4T3	BamHI … SalI	-
TC-α_1I	ITGA1	P56199 (147-360)	pGEX4T3	BamHI … SalI	YES
α_2I	ITGA2	P17301 (188-378)	pGEX4T3	BamHI … EcoRI	-
TC-α_2I	ITGA2	P17301 (188-378)	pGEX4T3	BamHI … EcoRI	YES

Table 5.3-1: Information oabout the four I domains expressed and purified in this thesis.

Plasmids containing the cDNA of I domains (α_1I, α_2I) of integrin subunits α_1 and α_2 were a kind gift of Sue Craig (Martin Humphries Lab, Faculty of Life Sciences, University of Manchester, UK). Plasmids were transformed into DH5α e-coli by Dr. Amit Roy.

The cDNA of TC-taged I domains was constructed by amplifying the purified α_1I, α_2I plasmids by PCR using appropriate primers designed using the "oligocalc" software tool (http://www.basic.northwestern.edu/biotools/OligoCalc.html). The cDNA of TC-α_1I was amplified from α_1I cDNA using 5-ATAT - GGATCC - TGG GAT TGT TGT CCC GGG TGT TGT AAG -

GTC AGC CCC ACA TTT CAA GTC[7]-3 (N-primer; T_m = 54-61°C), and 5-ATAT - GTCGAC - TCA GGC TTC CAG GGC AAA TAT-3 (C-primer; T_m = 52-60°C). The cDNA of TC-α_2I was amplified from α_2I cDNA using 5-ATAT - GGATCC - TGG GAT TGT TGT CCC GGG TGT TGT AAG - TGC CCT TCC CTC ATA GAT GTT-3 (N-primer; T_m = 52-60°C) and 5-GTGT - GAATTC - TCA AAC AGT ACC TTC AAT GCT GA-3 (C-primer; T_m = 52-60°C). Both N primers contain in frame the BamHI recognition site GGATCC and the cDNA that encodes for the tetracysteine tag WDCCPGCCK. Both C-primers contain in-frame a restriction recognition site (SalI and EcoRI) just after the TGA stop codon. All primers were purchased from IDT (Integrated DNA Technologies Inc, Coralville, IA). PCR reactions used the PCR master mix (M7505, Promega) and took place in the Dianne K. Newman laboratory (Dept. of Biology, MIT; current address: California Institute of Technology, Pasadena, CA) under the supervision of Dr. Lars Dietrich (current address: Assistant Professor, Department of Biological Sciences, Columbia University, NY). Successful PCR amplification was verified by 2% agarose gel DNA electrophoresis and ethidium bromide staining (protocol I.2).

The amplified cDNA of each TC-tagged I domain was inserted into a pGEX4T-3 vector. Linearized pGEX4T-3 vectors were obtained by cleaving purified α_1I plasmids with BamHI and SalI (101228-740 and 101228-752, New England Biolabs), separating cleavage products by 2% agarose gel DNA electrophoresis, and purifying the cleaved pGEX4T-3 vectors from the gel using the Wizard Plus SV Gel kit (A9281, Promega, Madison, WI). TC-α_1I and TC-α_2I cDNA was ligated in the cleaved pGEX4T-3 vectors using T4 DNA ligase (101228-180, New England Biolabs, Ipswich, MA) overnight at 14°C in a tropicooler device (Dedon Lab, MIT). The ligase reaction buffer was then used to transform competent DH5α e-coli by thermal shock (protocol H.8). Transformed cells were selected by spreading 50 µl transformed cell suspension on LB-agar-amp plates and incubating overnight in a 37°C incubator (Dedon Lab, MIT). Upon successful ligation, several (5-50) DH5α colonies formed. Five colonies of each kind are inoculated and amplified in 5 ml LB-amp medium overnight in a 37°C shaking incubator (Dedon Lab, MIT). Purified plasmids from 3 ml of overnight culture of each colony were sequenced in the MIT biopolymers facility, using the 5-TATAGCATGGCCTTTGCAGGG-3 "pGEX primer". Those colonies that sequencing indicated successful transformation were stored long-term at -80°C after adding glycerol to 10%.

While DH5α are efficient for cDNA amplification and storage, they are not appropriate for recombinant protein expression. On the other hand, the BL21 *E.coli* strain is more appropriate for recombinant protein expression due its deficient protease machinery. BL21 *E.coli* were transformed with I domain plasmids purified from 5ml DH5α overnight cultures. Competent BL21 cells (Sigma Aldrich) were transformed by thermal shock (protocol H.8). Transfected cells were selected by spreading 50 µl transformed cell suspension on LB-agar-amp plates and incubating overnight in a 37°C incubator (Dedon Lab, MIT). Upon successful transformation, several e-coli colonies form. Five colonies from each plate were inoculated and amplified in 5 ml LB-amp medium overnight in a 37°C shaking incubator (Dedon Lab, MIT). Purified plasmids from 3 ml overnight culture were sequenced in the MIT biopolymers facility using the pGEX primer.

[7] Underlined text in PCR primers corresponds to the part that will hybridize with the plasmid to be amplified

Colonies for which sequencing indicated successful transformation were stored long-term at -80°C after adding glycerol to 10%, and later used for I domain expression.

Expression and Purification of Recombinant I Domains

All four I domains variant are recombinantly expressed in BL21 e-coli as GST fusion proteins. After ***E.coli*** lysis, fusion proteins are batch purified via affinity chromatography. Then purified fusion proteins are treated with thrombin to cleave and remove the GST domain. The cleaved GST domains and the uncleaved fusion proteins are removed via affinity chromatography, and the purified I domains are stored at 4oC until use. The procedure for the expression and purification follows established practices for the expression of GST fusion protein using the pGEX system [Harper and Speicher 2008; GST Gene Fusion System Handbook 2009]. The purification procedure consists of the following paragraphs:

- Protein Expression: After striking BL21 glycerol stocks (stored at -80°C) in LB-agar-amp plates, and overnight culture at 37°C, single colonies are inoculated into 5 ml LB-amp and incubated overnight at 37°C with 250 rpm shaking. The next day, the saturated bacteria culture is transferred into 200 ml fresh LB-amp and incubated at 37°C with 250 rpm shaking. The 595nm O.D. is monitored in a spectrophotometer (Weiss Lab, Department of Biological Engineering, MIT). When the culture O.D. reaches 0.6, the production of GST-I Domain fusions is induced by adding 400 µl 0.2M IPTG (I5502, Sigma Aldrich, St Louis, MO). Bacteria are cultured 3h more at 37°C with 250 rpm to express the GST-I Domain fusion protein. The bacteria suspension is then poured in clean 500 ml centrifuge tubes, and bacteria are pelleted by centrifugation (6500×g, 10 min, room temperature). After disposing the supernatant (medium), the bacteria pellet is reconstituted in 50 ml PBS in clean 50 ml centrifuge tubes. Bacteria are pelleted by centrifugation (6500×g, 10 min, room temperature), the supernatant PBS is disposed, and the bacteria pellets are stored in a -80°C freezer. For troubleshooting, 500 µl aliquots of the bacteria culture are removed before and after the 3h protein expression incubation.
- Cell lysis: The pellets of GST-fusion expressing bacteria stored at -80 °C are thawn on ice for 30', reconstituted in 5.6 ml lysis buffer (PBS^-, protease inhibitors, 1 mM DTT, 2.5 mM $MgCl_2$, 0.5 mg/ml lysozyme, 10 µg/ml DNAseI, 150U benzonase) and incubated 1h on ice. Bacteria are lysed by sonification (4 pulse trains, each 10 sec long, let suspension cool 1 min between pulse trains). After adding 60µl 0.5M EDTA (to final 5 mM) and 600 µl 10% tritonX100 (to final 1%), the suspension is incubated 15 min on ice, and centrifuged 14000×g for 60 minutes at 4°C. The supernatant protein extract is carefully transferred into ice-cold test tubes. No solid pellet needs to be transferred. If necessary repeat the 14000×g centrifugation once more.
- GST fusion protein purification: Add 500 µl washed glutathione agarose beads (15160 thermo scientific, Waltham, MA) to the protein extract, and incubate the bead suspension 3h at 4°C so that beads capture the GST fusion proteins. The beads are then pelleted by centrifugation (280×g, 3 min, 4°C), washed three times with bead wash buffer (PBS, 1 mM DTT, 0.1% TritonX100, 2.5 mM $MgCl_2$, 1/10×protease inhibitors), incubated 20 min in room temperature in dnaK buffer (bead wash buffer, 2.5 mM ATP) to wash away e-coli chaperones [Rial and Ceccarelli 2002], and washed three times more in PBS. Then the GST-I Domainfusion proteins are eluted by incubating in elution buffer (50 mM Tris pH8, 10

mM reduced glutathione) for 1h at 4°C. The eluted proteins are dialyzed in PBS using a 3.5K MWCO dialysis cassette (66330 Thermo scientific, Waltham, MA) and 0.2 μm filter sterilized.

- Thrombin cleavage and removal of GST domains: Add 10 μl 100× cleavage buffer (200 mM $MgCl_2$, 0.2M TCEP, 2% Tween 20 in PBS) and 25μl thrombin stock solution (0.5 units/μl in PBS; 27-0846-01, G.E. healthcare) per 1 ml protein extract. Incubate 36 hours at 4°C. Catch the cleaved GST domains and any uncleaved GST-I Domain fusions by adding 500 μl washed glutathione agarose beads, incubating the bead suspension 3h at 4°C, and pelleting the beads by centrifugation (280×g, 3 min, 4°C). Keep the supernatant. Quantify the concentration and purity of I Domain solutions via A280 absoption and SDS-PAGE correspondingly (Protocol J.4).
- I Domain solution concentration: This step is necessary only if the concentration of the purified I domain solution is not high enough (i.e. 15 μM for TC-tagged I domains, 25 μM for non TC-tagged I domains). A dense enough I domain solution is necessary for the experimental implementation of binding experiments (ligand density measurements, BIACORE assay). Concentration of I Domain solutions is implemented by filter centrifugation using 4ml Amicon 4 kDa units (Millipore Corporation, Ballerica MA).

Handling I domain solutions

Several research groups that purified I domains for experimental applications, reported that several issues need to be considered during I recombinant expression and purification. The most importnat issue is that purified I domains aggregate and precipitate over time. The rate of I domain precipitation can be significantly reduced by adding a reducing agent [Xu et al. 2000]. Some studies suggest that purified I domains need to be used within 2 weeks after purification [Xu et al. 2000]. Several studies identified the formation of disulfide bonds between the GST domain and the I domain, making necessary to treat I Domains with a reducing agent in the end of the thrombin cleavage step so that the GST can be released and removed [Tuckwell et al. 1995]. A few researchers suggested avoiding sonification and using chemical lysis bacteria methods, however most published papers use sonification [Estavillo et al. 1999; Smith et al. 2000; Krahn et al. 2006].

Troubleshooting I domain expression and purification revealed the following:

- The yield of purified α1 I domains (α_1I, TC-α_1I) was larger than the yield of α2 I domains (α_2I, TC- α_2I). The yield of purified tag-free I domains (α_1I, α_2I) was also larger than the yield of TC-tagged I domains (TC- α_1I, TC- α_2I). A typical purification batch, produced 2ml 1.31 mg/ml (54.9 μM) α_1I, 2ml 0.64 mg/ml (25.8 μM) TC-α_1I, 750 μl 0.97 mg/ml (41.2 μM) α_2I, and 500 μl 0.39 mg/ml (15.9 μM) TC-α_2I.
- I domain solubility is dramatically reduced after GST removal. While it was possible to purify >2 mg/ml of any GST-I Domain fusion by 1L e-coli cultures, thrombin cleavage induced massive protein precipitation. To improve solubility, 0.02% tween 20 detergent, and 2 mM $MgCl_2$ is added to the thrombin cleavage buffer. The chosen concentration of tween20 detergent is low enough so that it does not interfere with the Bradford protein assay (Section 5.3.3), and 3 times less than its critical micelle concentration (0.07% w/v or 60 μM [Bhairi 2001]) to avoid detergent concentration during centrifugal filtering.
- Several studies suggest to add 1mM βme reducing agent in the purified I domain solutions to

avoid protein precipitation. This was not implemented in this study as βme interferes with the staining of TC-tagged I domains by biarsenical dyes [Adams et al. 2002], see Section 5.3.6. Instead 2 mM of the alternative reducing agent TCEP is added to purified I domain solutions. TCEP is known not to affect the reaction between a TC tag and a biarsenical dye, however TCEP is not particularly stable in phosphate buffers. Nevertheless, no protein aggregation was observed 1 month after purification and addition of TCEP.

- Thrombin cleavage of I domains is not very efficient. Cleavage of the thrombin recognition site (TRS) located between the GST tag and the I domain may be impeded sterically by disulfide bonds that form between the GST and the I domain [Tuckwell et al. 1995]. This may explain why thrombin is less effective in cleaving TC-tagged I domains, because the cysteines of the TC tag are located next to the TRS. Aaddition of reducing agents (DTT or TCEP) did not improve cleavage efficiency significantly. Nevertheless, it is necessary to add 2mM TCEP in the cleavage buffer during the last hour of the thrombin reaction to release the cleaved GST .
- Freezing I domains in PBS buffer caused protein aggregation. It was not possible therefore to store the purified proteins long-term. Snap-freezing in liquid nitrogen did not avoid aggregation. Optimization of the freezing buffer (e.g. add glycerol and store at -20°C) was not attempted.
- Protein concentration of I domains by filter centrifugation needs to be done carefully in order to avoid protein precipitation. Even though the specifications of AMICON centrifugal devices suggest concentrating protein solutions by spinning up to 4000×g for 30-90 min, in this study centrifugation was implemented in short centrifugation steps (15 min each) by applying no more than 2200×g. After each centrifugation step the retained I domain solution was mixed using a pipette to avoid high protein concentrations in the proximity of the filter, which can lead to precipitation.

5.3.3 Characterization of CBl I domains I: Protein Concentration and Purity

Measuring Protein Concentration (Protein Assay)

After protein purification, a critical step in characterizing the purified I domain solutions is to estimate their concentration. Accurate protein estimation is critical because the design of protein binding experiments (adhesion ligand surface density assay, BIACORE assay) and the interpretation of the results depend on the concentration of the binding partners. Estimating the concentration of purified I domain solutions was done using two methods:

The Bradford assay provides a quick and coarse estimation of protein concentration after various steps of the purification process (Protocol J.3) [Rehm 2006]. Although it is known that it is affected by the presence of detergents, the concentration (0.02%) of the tween20 at the I domain buffer is less than the minimum concentration (0.06%) that interferes with the Bradford assay [Pierce protein assay selection guide, Thermo scientific]. It was possible to generate reasonably linear protein concentration calibration standard curves by diluting 2 mg/ml BSA standards in PBS+3 buffer (PBS, 2 mM $MgCl_2$, 2 mM TCEP, 0.02% tween 20) as low as 0.015 mg/ml. The Bradford assay requires 3 µl protein solution, and measurements took place in a nanodrop 2000 spectrophotometer (Weiss Lab, MIT).

More accurate concentration measurements of I domain solutions were conducted using the 280nm absorption method (Protocol J.1).This method requires that the protein solution contains

a single protein [Wilson and Walker 2000], whose molar absorption coefficient ε_{280} can be estimated based on the protein amino acid primary sequence [Pace et al. 1995]. This method is not affected by the presence of detergents or reducing agents in the protein buffer.

Quantifying Purity

The purity of the purified I domain solutions is evaluated via SDS-PAGE electrophoresis and Coomassie Blue staining (Protocol J.4). Loading 4-5 μg purified protein solution per gel lane is enough to get bright protein bands.

Coomassie Blue staining is a generic protein staining method. However, it is possible to exploit the high-affinity of the biarsenical dye-tetracysteine interaction in order to specifically stain only tetracysteine-tagged proteins in a multiple-protein solution [Adams et al. 2002]. The method, "SDS-PAGE-FlAsH" (Protocol J.5) is a modification of the SDS-PAGE method where protein solutions are incubated in reducing Laemmli buffer (need to contain TCEP instead of βme) in the presence of 5% excess FlAsH dye for 60 min before boiling and gel loading. After running the gel, the gel is imaged in a fluorescent gel imager that contains a fluorescein filter set (Dedon Lab, MIT) in order to detect only TC-tagged proteins. The gel is subsequently stained with coomasie blue in order to visualize all proteins.

5.3.4 Characterization of CBI I domains II: Circular Dichroism Spectroscopy

After expressing and purifying the recombinant I Domains, the secondary structure of the purified proteins was evaluated by circular dichroism (CD) spectroscopy. The objectives of CD experiments were i) obtain evidence that the purified proteins were folded correctly, and ii) obtain evidence that the addition of the tetracysteine tag did not affect folding significantly.

Overview of the Circular Dichroism Assay

Circular dichroism is an optical spectroscopic method that can be used to estimate the secondary structure of proteins in solution. In contrast to other methods of structural biology (X-ray crystallography or NMR) far-UV CD cannot provide structural information of proteins at atomic level of resolution. Nevertheless, CD is a much less demanding method (in terms of complexity and sample requirements) that can provide information on the structure of a protein within a few hours [Kelly et al. 2005; Greenfield 2006]. CD can be applied to estimate the unknown secondary structure of a protein, or to monitor changes in the conformation of a protein due to mutations/edits, or changes in the protein environment [Johnson 1990; Pain 2004; Kelly et al. 2005]. As such, it is of interest to quantify whether the addition of a tetracysteine tag in the N terminus of an I domain affects its conformation.

Circular dichroism is based on the property of asymmetric molecules to absorb differently right-handed and left-handed circularly polarized light. When circularly polarized light passes through a solution that contains such molecules, the output light has elliptical polarization. The two requirements for a molecule to exhibit CD at some region of the EM spectrum are that the molecule absorbs in that particular EM region, and that the molecule is associated with an optically asymmetric environment [Pain 2004]. There exist several EM regions where proteins absorb light. The region of interest for protein conformation studies is the far UV region (180-260 nm), where light absorption by amide bonds between residues depends on the protein secondary structure [Pain 2004]. Specifically, the presence of particular secondary structures

(α-helix, anti-parallel β sheets, β turn, triple helix) corresponds to particular CD signatures, see Figure 5.3.1.

The unknown secondary structure of a protein in solution can be estimated by measuring the resulting CD after light passes through a known solution length d of the protein solution. CD measurements are conducted over a range of far UV light wavelengths, λ, usually between 180 nm and 250 nm. CD instruments report either the absorbance difference ΔE (difference of right handed minus left-handed circular polization), or equivalently the ellipticity $\theta(\lambda)$ (in degrees). The mean residue ellipticity $\theta_{mrw}(\lambda)$ (in deg·cm^2·dmol^{-1}; independent of the protein's molecular weight) is calculated as:

$$\boldsymbol{\theta_{mrw}(\lambda) = \frac{1}{10 \cdot n \cdot C \cdot d} \theta(\lambda)} \qquad \textbf{[5.3-1]}$$

where n is the number of residues of the protein, C is the protein concentration, and d is the light path length through the solution.

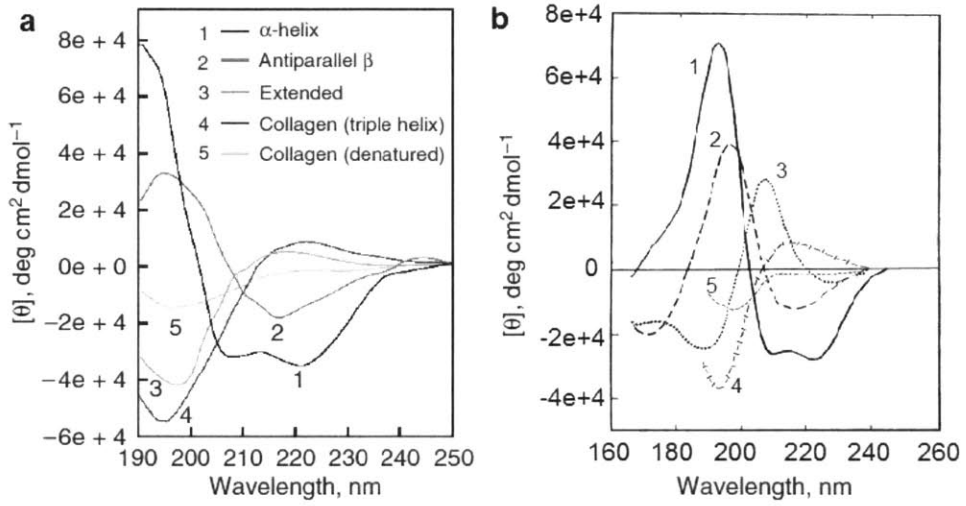


Figure 5.3.1: Reference far-UV CD spectra of protein with known secondary structure. a) CD spectrum of poly-L-lysine and collagen at different conformations. 1: poly-L-lysine as α-helix, 2: poly-L-lysine as anti-parallel β sheet, 3: poly-L-lysine extended (denatured), 4: collagen triple helix, 5: denatured collagen [Greenfield 2006]. b) Spectra of various secondary structures. 1. α-helix, 2: anti-parallel β-sheet, 3: type I β-turn, 4: triple helical (poly-pro), 5: irregular structure [Kelly et al. 2005].

Experimental Design

The quality of the secondary structure estimation depends on several parameters. CD data that contain less noise and sample a wider range of far-UV wavelengths (e.g. 180-250nm) can provide better estimates [Johnson 1990; Greenfield 2006]. Including measurements at wavelengths below 200nm can improve significantly the structure estimation [Johnson 1990; Kelly et al. 2005]. However measurements below 200nm contain more noise due to increased light absorption by particular buffer components (e.g. salts) or by the cuvette glass. Therefore, it is necessary to pick the protein buffer carefully, and avoid components that absorb significantly in this region [Aviv Biomedical Technical Notes]. In order to prevent significant light absorption it is also advisable to avoid concentrated protein solutions. The literature suggests that the 280nm absorption of the protein solution should be A_{280} <1, however depending on the instrument and cuvette, it may be necessary to use even more dilute samples. The protein solution should be

pure enough (>95%) and its concentration C should be accurately determined [Kelly et al. 2006; Greenfield 2006]. Finally, it is necessary to correct the experimental data for the baseline CD signal, which is the CD detected using the same cuvette and the same buffer (without protein content) [Pain 2004]. In practice, the baseline signal $\theta(\lambda)$ of the buffer sets the low wavelength limit where low-noise CD measurements are feasible.

The CD spectra of purified I domain solutions (α_1I, TC-α_1I, α_2I, TC-α_2I) after GST cleavage and removal were measured in an Aviv Model 202 CD spectrometer (Aviv Biomedical, Lakewood, NJ) located at the MIT Biophysical Instrumentation Facility (room 68-470), led by Mrs Debby Pheasant. The buffer of all purified I domain solutions was PBS+3 (PBS, 2 mM $MgCl_2$, 2 mM TCEP, 0.02% tween 20). Although PBS is a worse buffer for CD compared to other buffers (e.g. 10 mM Phosphate, [Greenfield 2006]) it was chosen because this is the buffer used in other binding experiments that utilized I domains (Sections 5.3.5 and 5.3.7). Control experiments (Section 5.4.2) showed that the presence of 2 mM TCEP, 2 mM $MgCl_2$ and 0.02% tween20 did not have a significant effect on the baseline CD signal of the buffer compared to plain PBS. The concentration of all I domain solutions was measured by 280nm absorption spectroscopy (Section 5.3.3). Stock I domain solutions were diluted inn PBS+3 so that the 280nm absorption of the samples measured by CD was approximately $A_{280} = 0.17$. Experiments were conducted using a Far-UV Quartz cuvette (NE-21CD-Q-1, New Era enterprises Inc., Vineland, NJ), which was cleaned between subsequent measurements using one 5% soap rinse, ten deionized water rinses, followed by air drying. All CD wavelength scans were conducted using the following parameters: 4°C temperature, 195-249 nm wavelength range, 1 nm wavelength sampling step, 1 nm bandwidth, 5 sec averaging time, and 0.333 settling time. All samples were equilibrated at 4°C for 5 minutes inside the CD instrument before data acquisition.

Computational Analysis

The measured spectrum $\theta_{mrw}(\lambda)$ can be analyzed by several available software tools (see below). Each tool estimates the percent contribution of several possible secondary structures (αhelix, β strand, turn, random coil) in a protein, based on the data $\theta_{mrw}(\lambda)$ and a reference set of CD spectra obtained from proteins of known secondary structure [Greenfield 2006]. Since different algorithms use a different approach to generate and process the reference set, it is suggested that data are analyzed by more than one software tools.

Experimental data (expressed in ellipticity θ) were pre-processed by subtracting the baseline signal of the PBS+3 buffer, and then calculating the mean residue ellipticity $\theta_{mrw}(\lambda)$ using Equation 5.3-1. The estimation of the contributions of the various secondary structures to the measured CD spectra was implemented using three different CD analysis tools (SELCON3, CONTINLL, CDSSTR), available as parts of the CDPro software package (http://lamar.colostate.edu/~sreeram/CDPro/main.html, Colorado State University, CO). CD spectra processing followed the published instructions at the CDPro web site. The three software tools differ in the way they generate the reference set and the mathematical methods used to deconvolve the CD spectra (more details are available in [Greenfield 2006]). All three methods are known to provide good results for globular proteins (I domains are globular proteins). The known secondary structure for the non-tagged I domains (α_1I, α_2I) was obtained from the annotation of published crystal structures [Emsley et al. 1997; Nymalm et al. 2004].

5.3.5 Characterization of CBI I domains III: BIACORE Binding Assay

The BIACORE binding assay is a method used to quantify binding interactions between two different molecules. Here, it is utilized to characterize the interactions between the purified recombinant I Domains and collagen. The BIACORE assay therefore is used here as a functional assay to verify that the purified I Domains function as expected (an indirect evidence that I Domains are well folded), and to describe how TC-tagging an I Domain affects its binding to collagen.

BIACORE Assay Description and Underlying Physics

The BIACORE assay is a high-tech label-free method for studying binding interactions between two biomolecules. Although the method is based on the optical phenomenon of surface plasmon resonance (SPR), the term BIACORE is used due to the name of the company (now part of General Electric Healthcare) that pioneered the commercialization of SPR instruments in the early 1990's. The BIACORE assay is applied for characterizing and quantifying (affinity, kinetics, binding mechanism, biological activity) binding interactions between proteins, peptides, nucleic acids, lipids, and small molecules [Schuck et al. 1999; *Biacore Sensor Surface Handbook*].

In the BIACORE assay, one of the two binding partners (called "ligand"8) is immobilized on the surface of a gold-coated glass surface. The second binding partner (called "analyte") flows in solution inside a microfluidic channel that is in contact with the ligand-gold coated surface (FIG). In the other side of the glass surface, a polarized infrared (IR) light beam is incident on appropriate incident angle so that total internal reflection (TIR) takes place. This causes an evanescent EM wave to propagate normal to the glass plane towards the gold-film and the microfluidic channel [Sjölander and Urbaniczky 1991].

SPR is an electromagnetic resonance phenomenon that occurs in thin conducting films (such as the gold film used in BIACORE) placed between media of different refractive indices [Jason-Moller et al. 2006]. At some particular incident angle (called SPR angle), the amount of reflected light is reduced due to SPR at the gold film that enhances the electric field of the evanescent wave, and increases the amount of incident light that "leaks" into the evanescent wave (Figure 5.3.2). SPR formation in the gold film is affected by the nearby environment (within a distance $h_i \approx 100$ nm of on the order of tenths of nanometers), causing the SPR angle to depend on the mass bound on the chip surface. Any variation of the mass of the binding partners close to the gold film (caused by "ligand" molecules that are immobilized on the gold film, or when "analyte" molecules associate or dissociate with the immobilized ligand molecules) affects the evanescent wave, which is detected by measuring the SPR angle.

[8] : The use of the term "ligand" in this section (and the BIACORE literature in general) should not be confused with the meaning described in Section 5.2.2 and used in the rest of the thesis, were "ligand" refers to the small chemical motif that mediates binding.

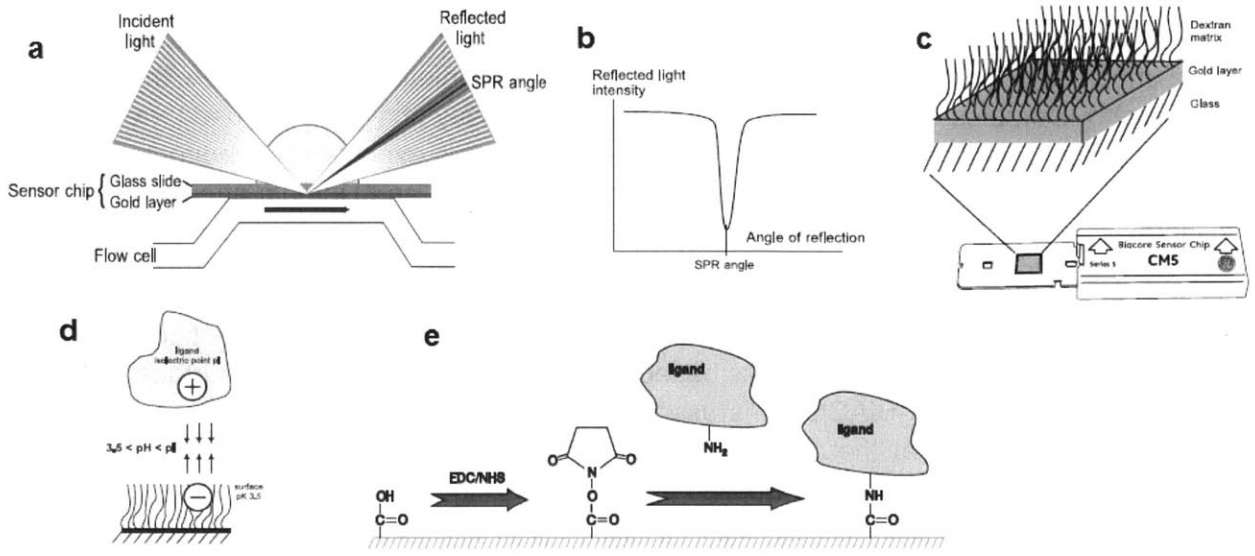


Figure 5.3.2: Schematics that describe the operation of the BIACORE assay. a) schematic of the BIACORE instrument demonstrating its key components: flow cell, gold surface, incident and reflected light, SPR angle. b) detection of the reflected light profile reveals the SPR angle. c) schematic of the CM5 sensor chip surface. d) Schematic of the electrostatic interactions during ligand immobilization. Ligands are concentrated on the dextran surface through electrostatic interactions induced by appropriate choice of the buffer pH. e) Amine coupling of ligands on the CM5 chip surface using EDC-NHS cross-linking reagents [BIACORE sensor surface handbook].

The output of each BIACORE experiment $R(t)$ (called "sensorgram") is a real-time measurement of the deflection of the reflected IR light beam as different kinds of buffers run through the microfluidic channel and interact with the gold-coated surface. The measurement $R(t)$ is expressed in resonance units (RU), where 1 RU corresponds to a 10^{-4} degree deflection of the reflected beam, caused by varying the mass on the surface of the sensor chip by 1pg/mm^2 (for a 150kDa protein) [Biacore Concentration Analysis Handbook 2008]. The instrument response R due to the presence of molecules of molecular weight M (g/mole) and surface density ρ_M (moles/mm^2) within the detection distance h_i away from the gold film equals:

$$R = \rho_M \cdot M \cdot G \qquad [5.3\text{-}2]$$

where $G = 10^{18}$ RU·m^2/g = 1 RU·mm^2/pg.

BIACORE experiments take place in specialized instruments that control the flow of buffers using elegant microfluidics, conduct SPR measurements, and display results. Experiments utilize "sensor chips" that contain four gold-coated surfaces of appropriate surface chemistry that enables immobilization of ligand molecules on the gold film. BIACORE provides several kinds of sensor chips, appropriate for different applications [BIACORE sensor surface handbook]. Each sensor chip contains four chambers (flow cells), which are connected via microfluidic chambers and can be controlled and measured independently.

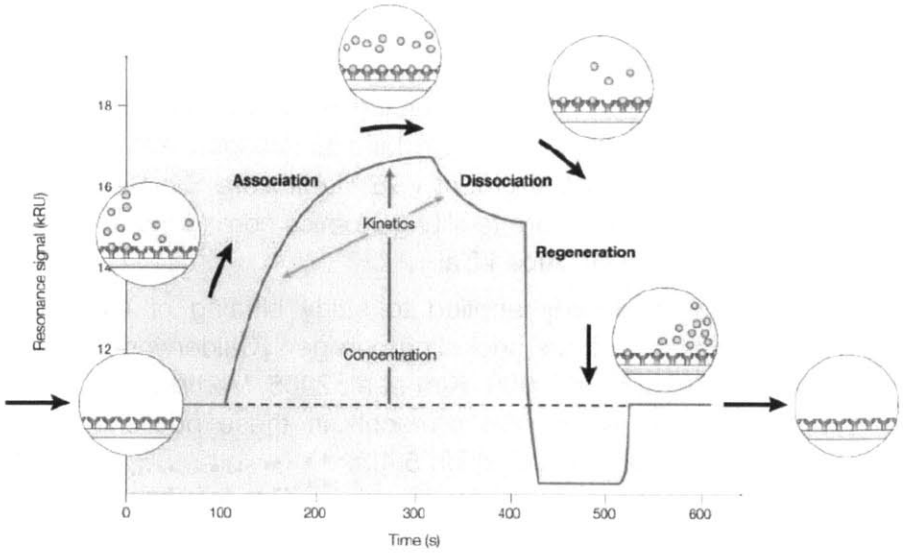


Figure 5.3.3: Schematic of the three steps of a typical analyte-ligand binding run: association, dissociation and regeneration. The plot shows the temporal response of the sensorgram, analogous to the amount of analyte bound to the immobilized ligand [Cooper 2002].

Experimental Design

Once the sensor chip is placed inside the BIACORE instrument, there is always flow of buffer through the four flow cells (chambers) of the sensor chip. Unless a particular buffer is run (see underneath), a running buffer runs through all four sensor chip chambers at all times.

BIACORE experiments consist of two parts:

- Ligand immobilization. In this part, an appropriate amount of ligand molecules are immobilized on the gold surface of each flow chamber of the sensor chip. There are several immobilization methods, depending on the surface chemistry of the sensor chip and the chemistry of the ligand. Immobilization is conducted once per chamber. It is possible to immobilize a different ligand in each flow chamber, although for experimental reasons it is better not to immobilize ligand in one of the chambers (blank) in order to be able to measure the binding of the analyte to the chip surface.
- Analyte-ligand binding. In this part, appropriate buffers that (may) contain analyte molecules flow through the chambers using the BIACORE microfluidic system. A typical binding experiment consists of four sequential steps: wash, association, dissociation, and regeneration (Figure 5.3.3). In the wash step there is flow of running buffers through all flow cells. In the association step there is flow of an analyte solution through the chambers; As analyte molecules bind to the immobilized ligand molecules, the measured response $R(t)$ increases. In the dissociation step again running buffer runs through the channels. As the previously bound analyte molecules dissociate from the ligand molecules, the response $R(t)$ decreases. In the regeneration step, an appropriate regeneration buffer runs through the chambers that forces the dissociation of the remaining analyte molecules bound on the immobilized ligands. Regeneration conditions need to be mild enough so that the immobilized ligand molecules do not dissociate or denature. The regeneration step acts as a "reset" that sets the chambers ready for the next binding run. After ligand immobilization, the same sensor chip can be used to run multiple binding runs that do not depend on each

other, provided that the regeneration step manages to dissociate all bound analyte without damaging the immobilized ligand.

In this study, BIACORE experiments were conducted to study how different I domains of collagen binding integrins (α_1I, α_2I, TC-α_1I, TC-α_2I) bind to collagen I molecules. Experiments were conducted in a BIACORE T100 instrument (GE Healthcare Bio-Sciences AB, Uppsala, Sweden) located at the Tufts University core facility genomics core (Jaharis 523A, 150 Harrison Ave, Boston MA 02111), directed by Dr. Albert Tai.

The BIACORE assay has been already applied to study binding of integrins or integrin I domains (α_1I, α_2I) to their binding partners, including collagen [Calderwood et al. 1997; Rich et al. 1999a; Estavillo et al. 1999; Xu et al. 2000; Kim et al. 2005; Vorup-Jensen et al. 2007]. The protocol described below follows closely the protocols in these published studies. A more detailed discussion of the results follows in Section 5.4.3.

The BIACORE experiments conducted in this thesis involved two kinds of ligands (collagen I, gelatin) and four kinds of analytes: two native non-tagged I domains (α_1I, α_2I), and two TC-tagged I domains (TC-α_1I, TC-α_2I) see Section 5.3.2). Ligands were covalently immobilized on the surface of CM5 sensor chips (GE Healthcare, Piscataway, NJ) via amine coupling. The gold layer of CM5 chips is covered with a 100 nm thick dextran layer that provides carboxyl groups for covalently immobilizing ligands (Figure 5.3.2c). Collagen I from tail tendon (tropocollagen) was purchased as a 3mg/ml solution in 0.012N HCl (354231 BD biosciences, San Jose, CA). Gelatin (denatured collagen) solution was prepared by heating the collagen I solution at 65°C for 30 minutes. Collagen was chosen as the ligand to be immobilized on the chip (instead of I domains) due to its much larger tendency for non-specific binding (caused in part by its much larger size: the molecular weight (MW) of each tropocollagen molecule is approximately 300 kDa, while the MW of I domains are around 24 kDa) [Schuck et al. 1999].

Collagen I, the main ligand of interest, was immobilized in chambers 2 and 4 of each sensor chip. Gelatin (denatured collagen I) was immobilized in chamber 3, while no ligand was immobilized in chamber 1 (blank). The BIACORE microfluidic system forces buffer to run sequentially through chambers 1, 2, 3 and then 4. The difference between the sensorgrams of chamber 4 (collagen) versus chamber 3 (gelatin) $R_{43}(t) = R_4(t) - R_3(t)$ probes the specificity of I domain binding to collagen I because it is known that denaturing collagen destroys the 3D conformation of the small motifs (binding ligands, e.g. GFOGER) necessary for integrin binding, see Section 5.2.3. The difference between the sensorgrams of chamber 2 (collagen) versus chamber 1 (blank) $R_{21}(t) = R_2(t) - R_1(t)$ is used to correct for various artifacts (see following sub-section) for example nonspecific binding of analytes to the dextran matrix of the CM5 sensor chip.

Ligand immobilization on the CM5 chip surface was implemented using the amine-coupling method, as described in the literature [Jason-Moller et al. 2006; Murphy et al. 2006] and the manufacturer's instructions (GE Healthcare, Piscataway, NJ). All chambers are first treated with a mixture of EDC and NHS solutions, which activate carboxyl groups on the dextran chains and make them reactive to primary amines. Then, 30 µg/ml solutions of collagen and gelatin in 10 mM sodium acetate pH 4.5. flow through chambers 2, 4 (collagen) and 3 (gelatin) respectively, so that activated carboxyl groups on the dextran chains form covalent bonds with primary amines at the ligand surface (Figure 5.3.2e). The pH=4.5 is chosen so that during

immobilization the positively charged ligand molecules (pH 4.5 is lower than the isoelectric point of collagen and gelatin) are attracted to the negative charges on the dextran layer (pH 4.5 is higher than the pKa of the carboxyl groups found on the dextran layer) causing local ligand concentration that favors ligand immobilization (Figure 5.3.2d). Finally, a wash with 1M ethanolamine quenches the remaining activated carboxyl groups that did not react with ligands. Since running buffer flows between consecutive crosslinking steps (EDC-NHS, ligand, ethanolamine), it is important to use a running buffer that does not contain primary amines. Ligand immobilization is implemented using an automatic script of the T100 instrument, so that eventually 1200 RU of collagen/gelatin are immobilized on the CM5 surface. This corresponds to a surface density of approximately 1.2 ng immobilized ligand per mm^2, or equivalently 2400 collagen molecules per μm^2. Given that the height of the dextran layer is 100 nm [BIACORE sensor surface handbook], this corresponds to a collagen volume fraction of 24% (assuming the dimensions of each collagen molecule are 400×5×5 nm). After immobilization, chambers were washed by 50 mM NaOH.

The binding interaction of each analyte with the ECM is studied by flowing solutions of increasing analyte concentration (0, 0.05, 0.13, 0.339, 0.830, 2.3 and 6 μM) and recording the binding of analytes to the immobilized ligands. Each binding run took place at room temperature using as running buffer HBS-P+ (10 mM HEPES, 150 mM NaCl, 0.05% v/v surfactant P20; GE Healthcare, Piscataway, NJ) supplemented with 1 mM $MgCl_2$. The necessary analyte concentrations were implemented by diluting stock analyte solutions (approximately 50 μM α_1I, 25 μM TC-α_1I, 40 μM α_2I, 15 μM TC-α_2I) in PBS+3 (PBS, 2 mM TCEP, 2 mM $MgCl_2$, 0.02% tween 20) with running buffer. Each association step lasted 3 minutes. The flow rate through the chambers during the association phase was 40 μl/minute. The dissociation step lasted 12 minutes. The flow rate during the dissociation phase was 40 μl/min. Regeneration was implemented by applying 20 μl/min regeneration buffer (HBS-P+, 10 mM EDTA, 0.01% v/v SDS) for 1 minute. All buffers (running, regeneration) were degassed for 2 hours prior to use. Each experimental condition was performed in duplicates. The total duration for each run (running buffer wash, association, dissociation, regeneration) was approximately 20 minutes.

Data Analysis I: Signal Conditioning

Every BIACORE experiment (run) provides a raw sensorgram $R_i^{raw}(t)$ i=1,2,3,4 for each one of four flow cells of the CM5 chip. Although ideally variations in the detected signal $R_i^{raw}(t)$ should be causwed only by analyte association/dissociation with the immobilized ligands, there exist several kinds of artifacts that affect the measured signal $R_i^{raw}(t)$ and complicate signal interpretation [Karlsson and Fält 1997; Myszka 1997; Schuck et al. 1999]:

- Baseline difference: Baseline is the (approximate constant) raw signal $R_i^{raw}(t)$ detected at a channel when running buffer (no analyte) runs through the flow cells, see Figure 5.3.2a. Each channel has a different baseline, due to small variations in the optical and mechanical parameters of each channel that define the baseline SPR angle, or due to small differences in the amount of immobilized ligand on the gold surface of each flow chamber.
- Instrument drift: refers to slow drift of the baseline signal caused by slow variation in the instrument's optical/mechanical systems. The observed drift was more significant in older instruments (e.g. BIACORE 2000, Center for biomedical engineering, MIT) compared to the

newer T100 used in this study. The drift appears to be approximately common in all four chamber signals of the T100 instrument.

- Instrument noise: noise inherent in the detection system. In the T100 instrument used in this study, instrument noise can be calculated from a drift-free part of the baseline signal as a white and Gaussian signal of standard deviation σ_R=0.11 RU, common to all four channels. This noise magnitude is significantly smaller than the maximum signal induced by analyte binding (20 to 40 RU in the samples of largest analyte concentration).
- Refractive index changes: A major source of artifacts are signal jumps caused by changes of the buffer's refractive index (RI). Since the detected signal depends on the local environment closed to the gold film, measurements are affected by the composition of the buffer that runs inside the microfluidic chamber. Every time a different buffer is applied through the microfluidic chamber, the SPR angle will change suddenly. Such a jump takes place at the beginning of the association step as the buffer switches from running buffer to analyte buffer. A jump of approximate the same magnitude but opposite sign should take place in the beginning of the dissociation step as the buffer switches back to running buffer.
- Matrix effects: the gold surface of the CM5 chip is functionalized with negatively charged dextran molecules (called the dextran "matrix"). The ionic strength of the buffer that runs through the flow chamber affects the distance between adjacent dextran molecules, therefore affects the mass content in the proximity of the gold film that is detected by SPR. When different buffers are applied through the flow chambers, the measured signal jumps not just due to RI changes, but also due to changes in the conformation of the dextran matrix. Matrix effects are affected significantly by the identity and the amount of immobilized ligand [Karlsson and Fält 1997].
- Non-specific binding: Refers to non-specific binding of analyte either in the dextran matrix, or in the immobilized ligand.
- Mass transfer: Affects kinetic measurements, where it is assumed that the con centration of analyte throughout the chamber is constant and analytes are not depleted ($I = I_0$). In practice, when the concentration of immobilized ligands is high (larger than 50) and when the flow rates are not fast enough (less than 30 µl/min), there is a gradient in the concentration of analyte close to the immobilized ligands. Mass slows both the association (creation of a analyte depletion zone) and dissociation kinetics (creation of a retention zone), and therefore can induce significant deviation from the ideal response in binding systems of strong affinity (low k_D) [Myszka 1997; Myszka et al. 1998; Schuck et al. 1999].
- Aggregate binding The presence of analyte aggregates induces slow accumulation of analyte in the sensor surface during the association step, and very slow dissociation of analyte during the dissociation step. Aggregate binding is caused by either the presence of analyte olimers in the analyte solution, or by local crowding effects in the proximity of the immobilized ligands (particularly in the case of large immobilized ligand density). Aggregate binding also induce significant deviation from the ideal response, in binding systems of poor affinity (high k_D) [Myszka 1997; Myszka et al. 1998; Schuck et al. 1999].

Signal conditioning refers to processing steps that remove some of the major artifacts from the raw detected signal $R_i^{raw}(t)$ and provide a signal $R_i(t)$, which can be fitted into a particular binding model. Some artifacts can be either estimated and compensated or incorporated into the mathematic model that describes the analyte-ligand binding response. However, some

artifacts (mass transport, matrix effects and aggregate binding) are hard to eliminate or incorporate in the mathematical model. Signal conditioning consists of three steps.

The first step is to calculate the difference $R_{21}^{raw}(t) = R_2^{raw}(t) - R_1^{raw}(t)$ (collagen surface – blank surface) and $R_{43}^{raw}(t) = R_4^{raw}(t) - R_3^{raw}(t)$ (collagen surface – gelatin surface) in a way that incorporates the small time delay necessary for the fluid front to travel between subsequent chambers. In modern BIACORE instruments (such as the T100) this is done automatically by the instrument acquisition software. Since the surface of channel 1 contains no immobilized ligand (reference surface) subtracting $R_1^{raw}(t)$ from $R_2^{raw}(t)$ should eliminate several artifacts such as baseline drift, non-specific binding on the dextran matrix and RI induced jumps [Myszka 1997]. Similarly, the signal $R_{43}^{raw}(t) = R_4^{raw}(t) - R_3^{raw}(t)$ should eliminate several artifacts such as baseline drift, non-specific binding on the dextran matrix and the gelatin molecule and RI induced jumps. Nevertheless the signals $R_{21}^{raw}(t)$ and $R_{43}^{raw}(t)$ still contain several important artifacts. One of them is that signals for different analyte concentrations have different baselines (shown in Figure 5.3.4a).

The second signal conditioning step corrects this baseline difference by subtracting the $R_{21}^{raw}(t)$ signal for a particular analyte concentration from the $R_{21}^{raw}(t)$ signal corresponding to zero analyte concentration, and then shifting the result so that the signal at the wash step is zero. At this step, results that correspond to the same analyte concentration are combined by taking their mean, in order to calculate the baseline-corrected sensorgrams $R_{21}^{bs}(t)$, $R_{43}^{bs}(t)$ for each analyte concentration (Figure 5.3.4b). The baseline-corrected sensorgrams are also calculated automatically by the BIAeval software (provided by the BIACORE manufacturer) and then fitted into a particular binding model.

As evident from Figure 5.3.4b the baseline-corrected sensorgrams can contain significant signal jumps in the beginning of the association and dissociation steps. As described previously, these jumps can be caused either by RI changes (not canceled completely by subtracting the signal of the reference blank channel) or by matrix effects. Furthermore, the steady-state portion of the dissociation curve can be non-zero due to matrix effects or aggregate binding. Since these jumps are on the same order of magnitude as the signal induced by analyte binding, it is necessary to either subtract these jumps before further processing the data, or incorporate these jumps in the models used to fit the data. Otherwise, artifact signal jumps can affect dramatically the estimated kinetic rates and steady state values of binding. The jumps observed in the baseline-corrected sensorgrams are larger in samples of high analyte concentration, which contain a larger fraction of the analyte stock solution buffer and less running buffer. In this study analyte solutions were prepared by diluting the stock analyte solutions using running buffer. Since the RI and the ionic strength of the running buffer (10 mM HEPES, 150 mM NaCl, 0.05% v/v surfactant P20) are different than the ones of the analyte stock solution buffer (137 mM NaCl, 2.7 mM KCl. 10 mM Na_2HPO_4, 1.8 mM KH_2PO_4, 2 mM TCEP, 2 mM $MgCl_2$, 0.02% tween 20), the observed jumps could be attributed to matrix effects.

The third signal condition step focuses on removing signal artifacts caused by signal jumps. Let δR_a and δR_d denote the magnitude of the signal jump at the beginning of the association and dissociation steps respectively. The duration of each jump equals the time required for the buffer front to move through the length of each flow cell and diffuse through the 100nm thick dextran matrix; in this study, the 40 μl/min flow rate corresponds to jumps that last 7 time

samples (0.7 sec). Also, let δR_{ss} denote the signal in the steady state of the dissociation step. Experimental observations suggest that $\delta R_{ss} \approx \delta R_a - \delta R_d$. Matrix effects are complex to include in a mathematical model solved by global optimization, therefore it was chosen to manually estimate the magnitude of δR_a and δR_d, and then remove the signal contribution of the jumps from the baseline-corrected sensorgrams. This correction provides the final corrected sensorgrams $R_{21}(t)$, $R_{43}(t)$ (Figure 5.3.4c). These corrected sensorgrams $R_{21}(t)$, $R_{43}(t)$ are then globally fitted into a particular binding model.

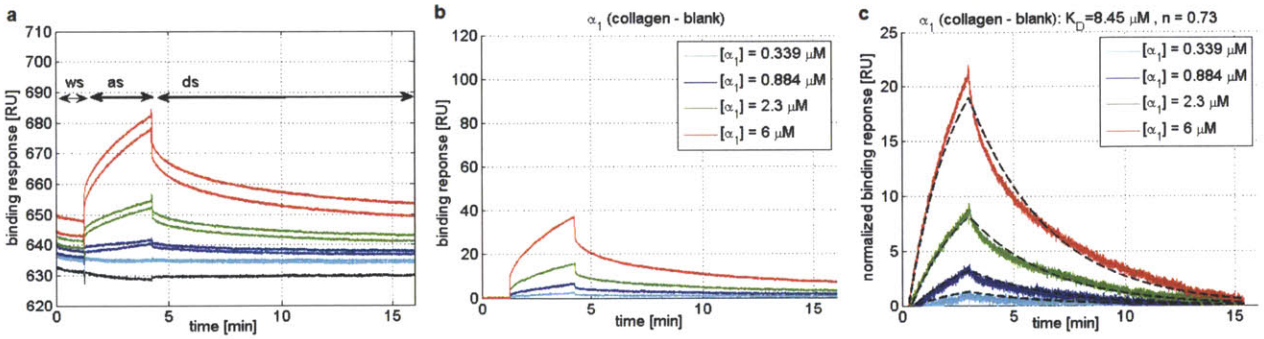


Figure 5.3.4: Signal processing steps for interpreting and fitting raw BIACORE data. a: Raw sensorgram data $\boldsymbol{R_{21}^{raw}}$ of α_1I (signal for α_1I binding to a collagen surface minus signal for α_1I binding to a blank surface). The top-most eight curves correspond to duplicate runs of 6, 2.3, 0.884 or 0.339 μM analyte [α_1I]. The bottom-most (black) is the signal obtained by running buffer (no analyte). The three main steps of each run are highlighted as "ws" (wash buffer), "as" (association step), and "ds" (dissociation step). The regeneration step is omitted for clarity. b: The resulting baseline-corrected sensorgrams $\boldsymbol{R_{21}^{bs}(t)}$ for each analyte α_1I concentration. Each baseline-corrected sensorgram is calculated by first subtracting from the signal of each analyte run the signal of the running buffer run, then calculating the mean of the two runs for each analyte concentration, and finally subtracting the constant signal of the "ws" step. This baseline-corrected sensorgram is used by the BiaEval for model fitting c: Corrected sensorgrams $\boldsymbol{R_{21}(t)}$ and their global fit to a 1:1 binding model (dashed lines). Corrected sensorgrams are calculated by first calculating manually the magnitude of the signal jumps $\boldsymbol{\delta R_a}$, $\boldsymbol{\delta R_d}$ in the beginning of the association and dissociation steps, and subtracting the jump contributions from the signal. The corrected sensorgram $\boldsymbol{R_{21}(t)}$ for each analyte concentration are then globally fitted by a 1:1 binding model (Eq. B2) to calculate the parameters $\boldsymbol{k_{on}}$, $\boldsymbol{k_{off}}$, and $\boldsymbol{n}$.

Data Analysis II: Binding Model

Binding of analytes (I Domains) in solution to their adhesion ligands present in collagen immobilized on a sensor chip surface can be modeled by a simple 1:1 binding model. This model is described by the following equation for the rate of formation of ligand-analyte complexes [Rich et al. 1999b]:

$$\frac{d\rho_B}{dt} = k_{on} \cdot I \cdot \rho_L - k_{off} \cdot \rho_B \qquad [5.3\text{-}3]$$

where ρ_B and ρ_L are the surface density of bound and unbound adhesion ligands in collagen molecules respectively (unit: moles/area), k_{on} and k_{off} are the association and dissociation rates of the analyte-ligand binding reaction, and I is the concentration (unit: M) of I domains in the proximity of the sensor chip surface. For more details on binding assays on a surface see Section 0. Assuming mass transfer effects are negligible (see discussion below) the concentration of analyte inside the microfluidic chamber is considered constant and equal to the nominal concentration of I domains applied by the user $I \approx I_0$. The density of unbound adhesion ligands on collagen molecules equals

$$\rho_L = \rho_{L0} - \rho_B = n \cdot \rho_{C0} - \rho_B \qquad [5.3\text{-}4]$$

where $\rho_{L0} = n \cdot \rho_{C0}$ is the total density of adhesion ligands for I domains, n is the number of I domain adhesion ligands per collagen molecule and ρ_{C0} is the surface density of collagen molecules (unit: moles/area). Combining Eq. 5.3-3 and 5.3-4, provides a linear ordinary differential equation that describes the dynamics of the detected signal $R(t)$ (the subscript i from the signal $R_i(t)$ of channel i is removed for simplicity):

$$\frac{dR}{dt} + \left(k_{on} \cdot I_0 + k_{off}\right) \cdot R = k_{on} \cdot I_0 \cdot R_{max}$$

$$R_{max} = R_C \cdot \frac{M_I}{M_C} \cdot n$$

where M_I and M_C are the molecular weights of I domain and collagen respectively, R_C is the amount of immobilized ligand (in RU), and R_{max} is the theoretically maximum signal $R(t)$ achieved in steady state when all adhesion ligands are bound by analyte. Assuming that during the association step ($0 \le t \le t_{on}$) I_0 remains constant (no analyte depletion), and that during the dissociation step ($t_{on} < t \le t_{on} + t_{off}$) the buffer flow succeeds in removing dissociated analyte without re-binding ($I_0 \approx 0$), then the solution of the differential equation provides the time response of the measured signal $R(t)$:

$$R(t) = \begin{cases} \frac{I_0}{I_0 + k_D} \cdot \rho_{C0} \cdot \frac{M_I}{M_C} \cdot n \cdot \left(1 - e^{-k_{off}\left(1 + \frac{I_0}{k_D}\right) \cdot t}\right) & , \ 0 \le t \le t_{on} \\ R(t_{on}) \cdot e^{-k_{off} \cdot (t - t_{on})} & , \ t_{on} \le t \le t_{on} + t_{off} \end{cases} \qquad [5.3\text{-}5]$$

where $k_D = k_{off}/k_{on}$ is the dissociation constant of the binding reaction, and $R(t_{on})$ is the signal at the end of the association step. While the time constant during the dissociation phase remains a constant $\tau = k_{off}^{-1}$, the time constant during the association phase $\tau = k_{off}^{-1} \frac{k_D}{(k_D + I_0)}$ decreases as the analyte concentration I_0 increases. When $I_0 \gg k_D$ and $\frac{k_D}{(k_D + I_0) k_{off}} < t < t_{on}$ then in steady state all ligand binding sites are bound to analyte $R = R_{max}$

This 1:1 binding model does not seem suitable to describe the binding of CBI I domains to their ligands in the collagen molecule. As discussed in Section 5.2.3 and 5.2.4, each CBI recognizes multiple kinds of ligands in each collagen molecule (some of which are present in more than one copies) with different affinities. Nevertheless, trying to incorporate more than one kinds of binding sites (of distinct dissociation constant k_D) in a binding model and fitting BIACORE kinetic data results in a very ill-behaved problem that provides unreliable results. Instead it was preferred to describe the I Domain-collagen binding system using multiple sites of a single average dissociation constant.

Data Analysis III: Global Fitting

Each binding model is described by a set of parameters $\boldsymbol{x}$. For the 1:1 binding model, $\boldsymbol{x}$ contains three parameters:

$$\boldsymbol{x} = [k_{on} \quad k_{off} \quad n]^T$$

The parameter vector $\boldsymbol{x}$ of a binding model can be estimated from a set of experimental BIACORE data based on a least-squares method. Let $R[k]$, k=1,2,...N denote the experimental data to be fitted. The index k refers to the time point and N to the number of time samples. The kind of data utilized for fitting depends on the algorithm used. BIAeval software uses as $R[k]$ the baseline-corrected sensorgrams $R_{ij}^{bs}(t)$ and therefore needs to augment the parameter vector $\boldsymbol{x}$

with the unknown signal jumps. Straightforward global optimization in MATLAB utilizes as $R[k]$ the corrected sensorgrams $R_{ij}(t)$.

Let $\widetilde{R}[k,\boldsymbol{x}]$, k =1,2,...,N be the simulated sensorgram response calculated using the model of interest given a parameter vector $\boldsymbol{x}$. The least-squares estimator $\hat{\boldsymbol{x}}$ is the parameter vector that minimizes the corresponding χ square metric:

$$\hat{\boldsymbol{x}} = \text{argmin}_{\boldsymbol{x}}\{\sum_{k=1}^{N}(\widetilde{R}[k,\boldsymbol{x}] - R[k])^2\}$$

Global fitting corresponds to calculating the least-squares estimator $\hat{\boldsymbol{x}}$ by incorporating results from several BIACORE response curves $R[k]$, each of which corresponding to a different analyte concentration I_0 (as opposed to calculating $\hat{\boldsymbol{x}}$ for the signal of each analyte concentration separately).

One way to estimate the parameters of the 1:1 binding system is to compute the corrected sensorgrams $R_{ij}(t)$ for the M analyte concentrations (as described in a previous subsection) and then globally fit them into a 1:1 binding model. This corresponds to solving the following optimization problem

$$\hat{\boldsymbol{x}} = \text{argmin}_{\boldsymbol{x}}\left\{\sum_{i=1}^{M}\sum_{k=1}^{N}(\widetilde{R}_i[k,I_0(i),\boldsymbol{x}] - R_i[k])^2\right\}$$

where the response $\widetilde{R}_i[k,I_0(i),\boldsymbol{x}]$ is calculated from Eq. 5.3-5 assuming the analyte concentrations $I_0(i)$ used in the experiments. This χ square metric incorporates data from M curves, each of which corresponds to a particular concentration I_0 of analyte and has N data time points. Since the metric to be minimized is a nonlinear function of parameters $\boldsymbol{x}$, this optimization problem can be solved numerically using a non-linear least squares solver (e.g. the Levenberg–Marquardt algorithm) or a generic nonlinear optimization algorithm [Nocedal and Wright 2006]. Here, numerical optimization was implemented using the functions lsqnonlin and fmincon of the MATLAB numerical computing environment (Mathworks, Natick, MA) [*Optimization Toolbox™ User's Guide*].

An alternative way to fit the baseline-corrected sensorgrams $R_{ij}^{bs}(t)$ signals into a 1:1 binding using the BIAeval software, provided by the manufacturer (BIACORE, Uppsala, Sweden). BIAeval calculates the baseline-corrected sensorgrams $R_{ij}^{bs}(t)$ automatically, and then fits them into an 1:1 binding model by including signal jumps in the beginning of the association and dissociation steps as additional model parameters. Results are provided for both methods.

Challenges in modeling and fitting BIACORE data

Accurate mathematical modeling of collagen binding to its binding partners is a hard problem due to the complex nature of the binding interactions. It has been demonstrated that each collagen molecule contains multiple distinct motifs where a particular CBI I domain can bind ("adhesion ligands" such as GFOGER, GLOGER), each one of which has a different binding affinity for that I domain (Section 5.2.3 and 5.2.4). Due to the long size and the structural stiffness of the collagen molecule, it is expected that binding of I domains to these adhesion ligands can be treated as independent (non-cooperative). By treating the binding of I domains to each ligand as an independent 1:1 binding event, the detected binding response $R(t)$ equals the sum of the responses caused by binding of analyte to each kind of adhesion ligand, each of which is described by an equation identical to Eq. B2 (assuming no analyte depletion).

Another source of difficulty lies in the fact that each I domain can exist in two conformation states ("open" and "closed", see Section 5.2.4), that have significantly different binding affinities for their adhesion ligands. Practically only the "open" state can bind to ligands on collagen. This causes further deviation from a simple 1:1 binding between I domains and collagen. Furthermore, there are no estimates for the reaction rates that control the equilibrium between the two I domain conformation states.

Based on the above two paragraphs, an accurate model that describes integrin binding to collagen would be quite complex and would contain many parameters that need to be estimated by fitting the experimental data. Although global curve fitting approaches improve the accuracy of parameter estimation, the resulting problem of fitting multiple curves to more than 1 exponentials is notoriously hard [Myszka 1997; Schuck 1997; Schuck et al. 1999; Goodrich and Kugel 2007]. Simulation results show that the presence of measurement noise and pipetting errors during sample preparation can induce huge estimation errors. Furthermore, the estimation of many parameters requires data from a large number of experiments so that independent variables are varied enough to "capture" all aspects of the model. This is usually not possible due to high experimental cost or other limitations (e.g. biochemical handling of the analyte solutions).

In summary, the literature suggests that fitting BIACORE data to complex models is very tricky, error-prone, and can lead to wrong conclusions (especially when a good fit is used as a way to support the validity of a complex model) [Schuck 1997; Schuck et al. 1999]. Since in this study the number of experiments for each analyte-ligand pair is limited (limited to a titration of the analyte concentration), and possibly prone to various artifacts (protein aggregation, mass transport), this study will not pursue anything more than describing each analyte-ligand pair with a simple 1:1 binding model, which will be attempted to capture the key properties of such interaction, and provide approximate answers to the following questions:

- Do the purified I domains of α_1 and α_2 integrin subunits bind specifically to collagen?
- Does the addition of the tetracysteine (TC) tag affect the affinity of I domains for collagen?

5.3.6 Fluorescent Staining of I Domains Using Biarsenical Dyes

Overview: Fluorescent Staining of Proteins

Fluorescent protein labeling is a key enabling technology for studying protein function, structure, and localization in vivo with high spatial and temporal resolution via microscopy [Giepmans et al. 2006]. Unfortunately, the vast majority of proteins do not emit fluorescence when excited by standard laser lines used in cell and tissue imaging (either UVA or visible) [Lakowicz 2006]. Fluorescence emitted by protein residues that contain aromatic side chains (tryptophan, tyrosine and phenylalanine) is excited by far UV light, which impedes practical cell and tissue imaging. Excluding indirect antibody labeling (which usually requires sample fixation and therefore cannot be applied *in vivo*), selective fluorescent labeling of a particular protein can be accomplished by fusing the protein of interest with an appropriate protein tag or a peptide tag [Giepmans et al. 2006; Wombacher and Cornish 2011; Crivat and Taraska 2012].

The most widespread way to fluorescently label a protein *in vivo* is to express the protein fused with a fluorescent protein (FP). While FP fusions offer choices of different photophysical properties (emission spectrum, stability, pH sensitivity), their relatively large size (27 kDa) may

affect the localization and function of the protein [Zhang et al. 2002; Giepmans et al. 2006; Lippincott-Schwartz 2011; Crivat and Taraska 2012]. The second class of protein tags are self-labeling enzymes, which are not fluorescent by themselves, however they can attach to a specific small fluorescent molecule and make the fusion protein fluorescent. This family includes enzymes that either covalently attach (e.g. SNAP tag, CLIP-tag, Halo-tag) or non-covalently bind (e.g. Edhfp, FKBP) to specific small fluorescent molecules [Wombacher and Cornish 2011; Crivat and Taraska 2012]. Such self-labeling proteins enable several imaging experiments (e.g. pulse and chase experiments), however their size is still significant (12-22 kDa), and their practice requires washing away the unreacted small fluorescent molecule [Wombacher and Cornish 2011].

Peptide tags are much smaller (6 - 30 amino acids) compared to protein tags. An ideal peptide tag would be as small as possible (to minimally disturb the protein function) and would interact with a particular fluorescent molecule with high specificity and affinity [Wombacher and Cornish 2011]. The most successful peptide tag is the tetracysteine tag (TC) [Griffin 1998], which reacts and covalently attaches to a small biarsenical dye. Upon binding to the TC tag, the biarsenical dye becomes 1000× more fluorescent. Due to the significantly smaller size of the TC tag-biarsenical dye complex compared to the size of a FP, this method has much less potential to interfere with the activity and function of the labeled protein [Griffin 1998; Zhang et al. 2002; Giepmans et al. 2006].

Tetracysteine Tag – Biarsenical Dye System: Description

The tetracysteine tag - biarsenical dye system is a method for fluorescently labeling proteins using a peptide tag and a small molecule [Griffin et al. 1998]. The method exploits the fragile covalent bonds that form between a trivalent arsenic (in organoarsenic compounds) and a pair of thiols located in two neighboring cysteines. The protein of interest is tagged with a small peptide (10-12 amino acids) that contains four cysteine residues organized in two pairs separated by two residues (CCxxCC motif). Biarsenical dyes (BD) contain two arsenic atoms separated by an appropriate distance, so that each arsenic atom can bond with each of the cysteine pairs of the tetracysteine motif. Although several BD have been developed, the two most widely used are FlAsH (4,5-bis(1,3,2-dithiarsolan-2-yl) fluorescein) and ReAsH (4,5-bis(1,3,2-dithiarsolan-2-yl) resorufin) [Griffin 1998; Adams et al. 2002; Spagnuolo et al. 2006; Adams and Tsien 2008; Pomorski and Krężel 2011]. The tetracysteine-biarsenical method has been applied to label proteins in many kinds of live cells (mammalian, plant, yeast, e-coli), as well as in FRET-based assays, single-molecule studies, protein research and biochemistry, material characterization, and in the control of protein activity, as reviewed in [Luedtke et al. 2007; Pomorski and Krężel 2011].

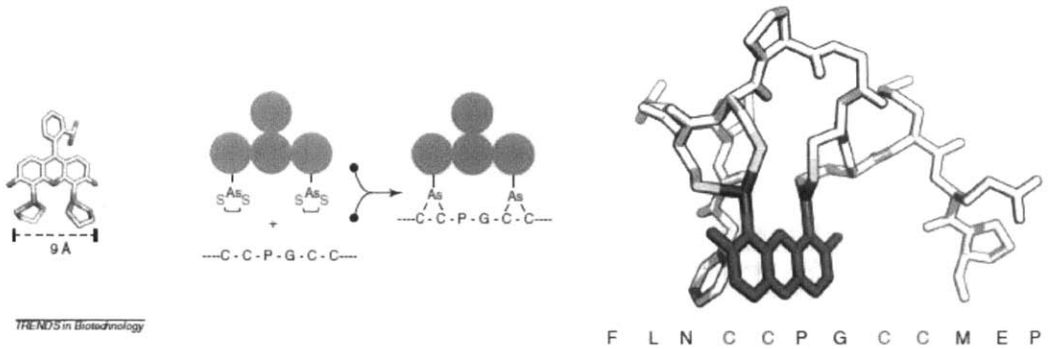


Figure 5.3.5: Molecular structure of FlAsH staining of tetracysteine-tagged proteins. Left: Chemical diagram of FlAsH-EDT2 complex. Middle: Chemical reaction of FlAsH-EDT2 staining of a CCPGCC tetracycteine tagged protein. Right: Model of the optimal FLNCCPGCCMEP motif bound to FlAsH (based on the NMR structure described in [Madani et al. 2009]. Red: FlAsH, purple: As atoms, yellow: thiol groups (in EDT or cysteines) [Crivat and Taraska 2012].

Tetracysteine Tag – Biarsenical Dye System: Chemistry

Biarsenical dyes (BD) can bind to cysteine-rich motifs that contain 4 reduced thiols in close proximity and appropriate orientation, such as the tetracycteine (TC) tag, and certain cysteine-rich proteins. BD also bind small vicinal dithiols such as 1,2-ethanedithiol (EDT) and 2,3-dimercaptopropanol (BAL). In tetracysteine labeling protocols, biarsenical dyes are bound to two EDT molecules [Adams et al. 2002; Hoffmann et al. 2010] that protect the BD and quench its fluorescence. The biarsenical dye becomes 1000× more fluorescent once it binds the TC motif. Due to the presence of the α-helix-breaking PG residues between the two cysteine pairs, TC motifs have hairpin 3D structure [Madani et al. 2009].

In practice BD are used with their As atoms bound to two small vicinal dithiols (e.g. EDT) that protect the two As atoms. During TC staining, the two As-EDT bonds are broken, and new bonds form between the two As and the cysteines of the TC tag. Fluorescent labeling of a TC-tagged protein (TC) by a BD (e.g. FlAsH) is described by the following equilibrium:

$$BD - EDT_2 + TC \leftrightarrow TC - BD + 2EDT$$

The kinetics of the tetracysteine tag-BD complex ($TC - BD$) formation depend significantly on i) the amino acid composition of the tetracysteine tag, ii) the biarsenical dye, iii) the presence and concentration on thiols in the reaction buffer [Adams et al. 2002; Cavit and Taraska 2012].

After the discovery of the first tetracysteine motif WEAAAREACCRECCARA [Griffin et al. 1998], additional screening revealed the much smaller tag CCPGCC that forms a more stable complex with biarsenical dyes [Adams et al. 2002]. Additional optimization of the flanking residues provided the optimal TC tags FLNCCPGCCMEP and HRWCCPGCCKTF for mammalian cell imaging [Martin et al. 2005]. Larger motifs provide better kinetics and offer the chance for improved contrast imaging, however they may interfere with the function of the tagged protein.

The presence of thiol-containing reducing agents affects significantly both the kinetics and the dissociation constant of the TC-BD binding. Small vicinal dithiols (EDT and BAL) bind to BD with significant affinity and compete with TC tags for BD. For the short CCPGCC motif, the IC_{50} value of BAL and EDT are 0.7 mM and 1.3 mM respectively [Hoffmann et al. 2010; Zurn et al. 2010]. For the optimized FLNCCPGCCMEP motif, the IC_{50} value of BAL is 5.6 μM. This affinity is utilized to reduce non-specific BD staining in live cells (see below). Small vicinal dithiols also reduce the apparent affinity of a TC motif for a BD. For example, the apparent dissociation

constant of FlAsH-CCPGCC formation in the presence of 200μM EDT is 0.24 μM, and this apparent dissociation constant depends on the square of EDT concentration [Chen et al. 2007]. Small monothiols (DTT, β-mercaptoethanol, MES) do not form stable bonds with the biarsenical dye, however they show some affinity for arsenoxides, and have been shown to accelerate TC-BD formation and slightly reduce its apparent dissociation constant [Griffin et al. 2000; Adams et al. 2002; Hoffmann et al. 2010]. Addition of thiols (DTT, β-mercaptoethanol) in BD solutions also results in some background fluorescence even in the absence of any TC tags [Griffin et al. 2000; Chen et al. 2007].

Tetracysteine Tag – Biarsenical Dye Staining Protocols

The teracysteine tag - biarsenical dye system has been applied mostly for intracellular protein imaging in mammalian cells, bacteria or yeast [Pomorski and Krezel 2011]. BD staining protocols aim to provide specific maximum-contrast staining of TC-tagged proteins by suppressing background staining, the major drawback of BD staining. BD background staining has two major origins: i) BD that binds cysteine-rich proteins (e.g. SlyD chaperones in bacteria [Wang et al. 2007]). This thiol-dependent background source can be reduced by adding small concentrations (<100μM) of small dithiols (EDT and BAL) [Griffin et al. 20000; Hoffmann et al. 2010] that compete for BD and reduce BD binding to cysteine-rich proteins. ii) BD that binds hydrophobic pockets in proteins (e.g. in bovine serum albumin). This is a thiol-independent source of background that can be reduced by adding non-fluorescent uncharged dyes (e.g. disperse blue, patent blue) in the staining solution [Griffin et al. 2000; Machleidt et al. 2007].

Most BD protocols consist of three steps: preparation, staining and washing. In the preparation step, samples are treated with a reducing agent (usually TCEP) in order to reduce the thiols of TC cysteines. At the same time BD is treated with EDT to get all As atoms bound to a EDT molecule. In the staining step, cells are treated with a BD solution in the presence of small amounts of excess EDT and an uncharged dye. In the washing step, cells are treated with a dilute BAL solution to reduce thiol-dependent background staining [Griffin et al. 2000; Machleidt et al. 2007; Hoffmann et al. 2010]. The BAL wash needs to be optimized in order to remove nonspecific signal without diminishing the specific signal emitted from TC-BD complexes.

Proposed Staining Protocol

In biochemical applications of BD staining of TC-tagged proteins, staining may be simplified compared to BD staining in live cell imaging. The reason is that since the identity of molecules present in biochemical experiments is usually well known, their interactions with BD can be tested. Under appropriate conditions the tedious washing steps with BAL could be omitted. One such example is staining of purified recombinant TC-tagged calmodulin with FlAsH described in [Park et al. 2004]. This study follows this approach for staining the TC-tagged I Domains:

- Reduce the cysteines of the tetracysteine tag in TC-tagged I Domains by treating them with 1 mM TCEP for 2h at 4°C.
- Treat the reduced TC-tagged I Domains with 5% excess FlAsH (previously treated with 100 μM EDT in order to get the cysteines of the TC tag react with EDT_2) in the presence of 1 mM βmercaptoethanol (βme) and 1 mM TCEP for 3h at 4°C.

- Remove unbound FlAsH-EDT_2, βme, TCEP, and EDT present in the FlAsH staining reaction via gel filtration using Biorad micro bio-spin 6 equilibrated in binding buffer (either PBS with 2 mM $MgCl_2$, or PBS with 10 mM EDTA).
    - It is necessary to remove unbound FlAsH-EDT_2 in order to prevent background staining with the blocking agent (e.g. BSA, casein, superblock) used in binding experiments.
    - It is necessary to remove the excess EDT and βme due to their nasty smell and because they increase the apparent dissociation constant of FlAsH for the TC tag from 10 pM (absence of EDT) to 240 nM (in the presence of 200 μM EDT) [Chen et al. 2007]. The presence of EDT and βme favors the dissociation of FlAsH from the TC tag during the binding experiment.

After gel filtration, the obtained I Domain solution is ready to be used in the binding experiments described in the following section.

5.3.7 Ligand Density Assay

The ligand density assay is an *in situ* binding assay, which provides the spatial distribution of adhesion ligands for particular adhesion receptors present in the surface of the matrix. Adhesion ligands are labeled using the FlAsH-stained TC-tagged I domains described in the previous sections of this Chapter). Then, by imaging the I Domain-treated matrix using a 3D microscope (here a spectral multi-photon microscope) it is possible to quantify the density of I Domains bound to their ligands.

Porous Collagen Scaffolds

The method developed is utilized to quantify the density of adhesion ligands for the two major collagen-binding integrins ($\alpha_1\beta_1$, $\alpha_2\beta_1$) on the surface of three kinds of porous collagen biomaterials (scaffolds). Type "D" collagen scaffold is the baseline scaffold. It is very similar to biomaterials applied clinically to induce regeneration in severely injured skin and peripheral nerves [Yannas 2001], made by freeze-drying micro-fibrillar collagen I (0.5% mass fraction) and then cross-linking it via dehydro-thermal treatment (DHT), see Appendix F.3. Scaffold "A" is identical to scaffold "D", however it is not cross-linked. Scaffold "E" differs from scaffold "D" in that it is cross-linked via chemical agents (EDC-NHS) instead of DHT. The three biomaterials are members of a degradation rate scaffold library (they are supposed to be identical in all aspects apart from cross-linking density thereby differing only in *in vivo* degradation half life), and they are identical to the scaffolds utilized in an animal study that provides the results described in [Soller et al. 2012] as well as the nerve samples shown in Chapter 4 of this thesis. The main difference between the materials used for ligand density experiments and the scaffolds utilized in [Soller et al. 2012] is the scaffold solid mass fraction (0.5% here, 5% in the animal study).

Binding Assay Implementation

Each binding assay consists of two experiments. In the "binding" experiment, the binding buffer of I Domains consists of PBS supplemented with 2 mM $MgCl_2$. Since divalent cations such as Mg^{+2} and Mn^{+2} are necessary for I Domain binding to collagen (Section 5.2.4), these conditions should enable binding of I Domains to collagen. In the "control" experiment, the binding buffer of

I Domains consists of PBS supplemented with 10 mM EDTA. EDTA is a chelator of Mg^{+2} and Mn^{+2}, therefore EDTA should prevent the binding of I Domains to their ligands.

The *in situ* binding assay consists of the following steps:

- Scaffold blocking: circular scaffold samples (aprorimately 3mm diameter, 3mm thick) are transferred inside PCR tubes, and hydrated in 50 µl PBS for 10 minutes. The excess medium is aspirated (each scaffold holds approximately 7µl of liquid) and the scaffolds are treated with 100µl superblock in PBS (37580, Thermo scientific) for three hours at 4°C.
- I Domain binding: The binding buffer is removed and 18 µl of I Domain working solution is added. I Domain working solutions are prepared by dilution the outcome of gel-filtrated stained I Domains (see Section 5.3.6) and supplementing it with superblock (to 1:10 final dilution) and in some cases non-tagged I Domains (competitive binding experiments). Scaffolds treated with I Domains are incubated overnight at 4°C to let I Domains bind their adhesion ligands on the matrix. After 8 hous the sample is assumed to have reached equilibrium and it is ready for imaging.
- Since the outcome of the gel filtration step contains trace amounts of FlAsH, and since FlAsH binds tightly to the TC tag and does not dissociate from the TC-tagged I Domains within the duration of the binding experiment and imaging, there is little concern that free unbound FlAsH binds non-specifically to the blocking agent (Super-block) present in the buffer. Instead over the duration of the experiment, practically all FlAsH will remain bound to TC-tagged I Domains. Therefore, there is no need to apply BAL washes to remove non-specifically bound FlAsH before imaging.
- Imaging: Each equilibrated I Domain-scaffold sample is transferred into a clean glass-bottom dish, and is imaged in the spectral multi-photon microscope described in Section 2.3.1. Imaging is implemented by acquiring a z-stack of 21 planes, each 256×256 pixels and 55×55 µm wide. The planes are 1 µm apart from each other and are located approximately 10 to 30 µm away from thje scaffold surface. Imaging settings are: 4.8 mW laser power, 775 nm laser wavelength, 40 µm pixel acquisition time. For each experiment three samples are imaged, two z-stacks per sample.

Experimental design

Three kinds of experiments were conducted.

- the baseline collagen scaffold "D" is treated and imaged with solutions of increasing concentration of stained TC-tagged I Domains, in the presence of 2 mM Mg^{+2} or 10 mM EDTA. The purpose of this experiment is to verify that the presence of EDTA reduces I Domain binding to collagen significantly, and verify that binding of TC-tagged I Domains to collagen requires Mg^{+2}.
- The baseline collagen scaffold "D" is treated and imaged with solutions of constant concentration of stained TC-tagged I Domains but increasing concentration of non- tagged I Domains (same type), in the presence of 2 mM Mg^{+2} or 10 mM EDTA. This is a competitive binding assay that aims to evaluate the specificity of detected I Domain binding to collagen scaffolds.
- The three collagen scaffolds ("A", "D", and "E") are treated and imaged with identical solutions of stained TC-tagged I Domains, in the presence of 2 mM Mg^{+2} or 10 mM EDTA. The purpose of this experiment is to evaluate the differences between the density of

adhesion ligands for the two CBI available for cell binding on the surface of these three materials.

Image Processing

The acquired images are processed by the image processing pipeline described in Chapter 2. Each image contains two types of regions (classes): "scaffold" and "solution". The "scaffold" region contains three sources of emission: FlAsH fluorescence, collagen intrinsic fluorescence, and second harmonic emission emitted by collagen. The "solution region" contains a single kind of emission, FlAsH fluorescence. What makes it feasible to segment the pixels of the two classes is the presence of collagen emission in the scaffold as well as the fact that the emission rate of FlAsH in solution (less than 5 photons per pixel) is at least one order of magnitude less compared to the fluorescence emission of FlAsH in scaffold pixels. This large FlAsH signal corresponds to FlAsH molecules bound to I Domains bound to ligands in collagen.

The ligand density assay aims to quantify the density of ligands on the surface of the scaffolds. This is done by quantifying the emission rate of FlAsH located in scaffold pixels located at the outer surface of the scaffold. Pixels located on the outer surface of the scaffold are identified as:

$$\mathcal{B} = \mathcal{F}_{scaffold} \cap \delta_2(\mathcal{F}_{solution})$$

where $\mathcal{F}_{scaffold}$, $\mathcal{F}_{solution}$ are the binary images corresponding to pixels classified as "scaffold" and "solution" correspondingly, and δ_2 is the dilation morphological operator (using a disk structural element or radius 2), see Chapter 2. Regions of interest are then manually picked, and the mean FlAsH intensity at the scaffold boundary pixels $\mathcal{B}$ inside this region corresponds to one measurement. The regions of interest are picked manually for consistency at locations of similar geometry.

Statistics

Each assay consists of two experiments. One experiment takes place in the presence of 2 mM Mg^{+2} in the buffer and one (control measurement) in the presence of 10 mM EDTA in the buffer). For each experiment, the mean μ and the standard error of the mean $\bar{\sigma}$ of FlAsH intensity at the outer surface of the scaffold is calculated based on four to six measurements (two to three scaffold samples per experiment, two measurements per scaffold sample). Measurements of FlAsH inetensity are then converted into equivalent FlAsH concentration (equivalently I Domain concentration) based on a standard calibration curve. The brightness of FlAsH bound on the TC tag is measured as B = $9.381 \cdot 10^{-6}$ photons/pixel/μsec/μM/(mW^2).

5.3.8 Analysis of Protein Binding Experiments

Binding Assays

This section summarizes assays for studying the binding of two molecules (called binding partners), denoted as "L" and 'I", that form a binding complex, denoted as "B". The choice of symbols is such that "I" corresponds to I domain molecules, and "L" corresponds to ligands of I domains located on ECM macromolecules.

Most binding assays quantify binding by measuring the amount of steady state binding complexes B_{ss} after the initial transient response. Such methods include the BIACORE assay and the adhesion ligand density assay described in Section 5.3.7. Few methods provide

accurate kinetic response $B(t)$ data that can be used to quantify the binding kinetic parameters (association and dissociation rates). One such method is the BIACORE assay (Section 5.3.5). Nevertheless, most estimates of binding parameters are based on how the steady-state complex concentration B_{ss} varies as a function of L_0 and I_0.

Binding assays can take place either in solution or on a surface. The key feature of all binding assays is a method to measure $B(t)$ without interference from the unbound partners L, I. In binding assays in solution, both binding partners and the resulting complex are located in the same solution. In binding assays on a surface, one of the binding partners (L) is immobilized on a surface, and the other partner (I) is in a solution that is in contact with the surface. In this case the resulting binding complexes B are also attached to the surface. Adhesion assays on surface include the BIACORE assay (Section 5.3.5), the developed adhesion ligand density assay (Section 5.3.7), and the solid phase binding assay.

In most binding assays, it is necessary to stain one of the binding partners (whose concentration is denoted as I^* to distinguish from the concentration of the unstained partner I) by attaching fluorescent entities or incorporating radioactive atoms. When binding reaches steady state, the unbound stained partner I^* is removed and the amount of B is estimated by measuring the amount of B bound to stained I^*. It these assays, it is assumed that the "hot" (stained) form I* and the "cold" form I bind identically with their binding partner L.

1:1 Binding Reaction in Solution

In one to one (1:1) binding models, it is assumed that each soluble I domain binds its binding partner that contains a single ligand per molecule. When the total concentration of ligands L_0 is unknown, L_0 can be estimated by quantifying how the steady state concentration B_{ss} of bound complexes B varies as the initial concentration I_0 of I domains varies. In this section it is assumed that all I domains are "hot" (stained) so the superscript * for "hot" I is omitted.

One to one (1:1) binding of the two binding partners into a binding complex is described by the chemical reaction:

$$I + L \leftrightarrow B$$

The concentrations of L, I, and B are denoted as $L(t)$, $I(t)$ and $B(t)$ respectively (in units of [M]). The kinetics of the 1:1 binding interaction model are described by the rate of B formation:

$$\frac{dB}{dt} = \dot{B} = k_{on} \cdot I \cdot L - k_{off} \cdot B$$

$$B(0) = 0$$

where k_{on} and k_{off} are the association and dissociation rates (in units of [$M^{-1}s^{-1}$] and [s^{-1}] respectively). The initial condition $B(0) = 0$ implies that initially there is no complex. As binding proceeds and B complexes form, the concentration of unbound L, I are reduced due to the formation of binding complexes B (depletion). At any time $L(t)$, $I(t)$ and $B(t)$ are constrained by mass conservation, which in solution is equivalent to a molarity constraint:

$$I_0 = I(t) + B(t)$$

$$L_0 = L(t) + B(t)$$

where L_0 and I_0 are the initial concentrations of L and I respectively. Expressing $L(t)$ and $I(t)$ as a function of $B(t)$ and substituting in the differential equation, provides an initial value problem that describes the binding response:

$$\dot{B} = k_{on} \cdot (I_0 - B) \cdot (L_0 - B) - k_{off} \cdot B$$

$$B(0) = 0$$

At steady state the rates of association and dissociation are equal. Setting $\dot{B} = 0$ and collecting the terms of $B = B_{ss}$ provides a second-order polynomial:

$$B_{ss}^2 - (k_D + L_0 + I_0) \cdot B_{ss} + I_0 \cdot L_0 = 0$$

The valid solution for B_{ss} , the root of this polynomial that lies between 0 and $\min(L_0, I_0)$. $k_D = k_{off}/k_{on}$, is the dissociation constant (in units of [M]). Therefore, the steady state concentration B_{ss} depends on the binding affinity k_D and the initial concentrations L_0 and I_0. Alternatively, the fraction of bound ligands at steady state $f = \frac{B_{ss}}{L_0}$ equals the root of the following polynomial that lies between 0 and min(1,λ):

$$f^2 - (1 + \kappa + \lambda) \cdot f + \lambda = 0 \qquad [5.3\text{-}6]$$

where κ, λ are non-dimensional parameters that describe the system affinity and stoichiometry:

$$\kappa = \frac{k_D}{L_0}$$

$$\lambda = \frac{I_0}{L_0}$$

Figure 5.3.6 shows contours of the steady state bound fraction f for various values of κ and λ (both axes are logarithmic). The plot can be separated in several regions. The right part of the plot corresponds to the case $I_0 > L_0$. The presence of "few" or "many" ligands in the binding reaction depends on the parameter κ, or equivalently on the value of L_0 relative to k_D. Depletion of I partners causes the curves of constant f to bend vertical, starts taking place for κ<10 and is significant for κ<1. For any value of κ, the fraction of steady-state bound ligands f is a monotonically increasing value of λ, whose slope equals:

$$f' = \frac{df}{d\lambda} = \frac{1 - f(\kappa,\lambda)}{\kappa + \lambda + 1 - 2f(\kappa,\lambda)}$$

The value λ_{50} of λ that results in $f = 0.5$ (50% of ligands L bound) depends on the value of κ. When there are "few" ligands (κ >10), then $\lambda_{50}(\kappa) = \kappa$ or equivalently $I_0 = k_D$. When there are "many ligands" (κ<0.1), then $\lambda_{50} = 0.5$ (equivalently $I_0 = 0.5 \cdot L_0$) is independent of κ.

The value of slope f' at λ_{50} equals:

$$f'_{50} = df/d\lambda\,|_{\lambda=\lambda_{50}} = \frac{1}{2(\kappa + \lambda_{50}(\kappa))}$$

In the case of "few" ligands (κ >10) then the slope $\lambda_{50}(\kappa) = (4\kappa)^{-1} = (4\lambda)^{-1}$ is a monotonous increasing function of k_D. In the case of "many" ligands then $\lambda_{50}(\kappa) = (2\kappa + 1)^{-1}$.

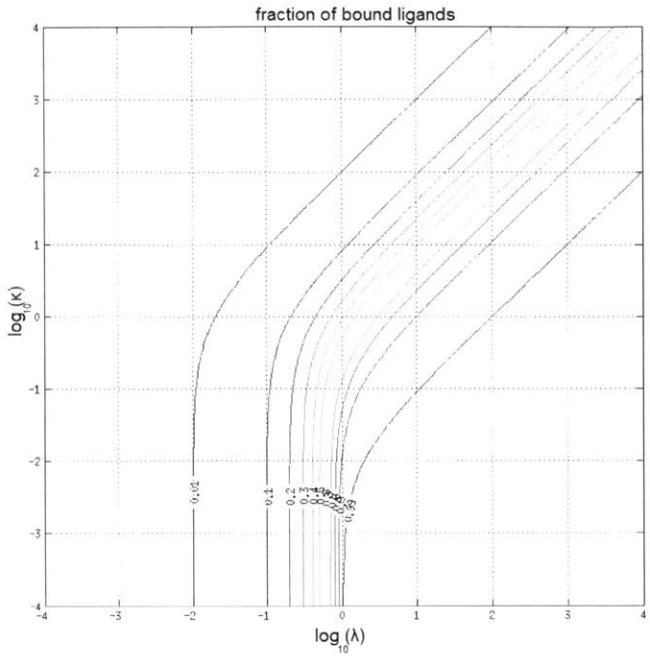


Figure 5.3.6: Contour plot of the fraction of bound ligands $f = B_{ss}/L_0$ at steady state in a 1:1 binding reaction as a function of the non-dimensional binding parameters $\boldsymbol{\kappa}$ and $\boldsymbol{\lambda}$.

Multiple Independent 1:1 Binding Reactions in Solution

The previous sub-section assumes that each I domain binds its partner (ECM molecule) at a single ligand. However, many extracellular molecules contain more than one distinct ligands recognized by a particular adhesion receptor. For example, collagen contains multiple kinds of ligands for the two major collagen-binding integrins $\alpha_1\beta_1$ and $\alpha_2\beta_1$ (Sections 5.2.3 to 5.2.4). The total concentration of ligands L_T available for binding to a receptor, can be estimated by studying how the steady-state number of I Domain-ligand complexes varies as the initial concentration of I Domains I_0 changes.

Assuming that each I domain I can bind J distinct kinds of ligands L_1, L_2, … L_J found on an ECM molecule in a non-cooperative manner (e.g. binding of an I Domain to a ligand in a collagen molecule does not affect the affinity of I domains for other ligands on the same molecule), total binding can be modeled as multiple 1:1 binding reactions that take place in parallel. In this case, the total concentration of ligands L_T in solution is:

$$L_T = L_0 \cdot \sum_{j=0}^{J} n_j$$

where n_j is the number of type j ligands per ECM molecule and L_0 is the initial concentration of the ECM molecule. Due to the non-cooperative binding assumption, the steady state concentration $B_{ss,j}$ (j=1,2,…,J) of the complex $I{:}L_j$ can be calculated using the steady state equations of the corresponding 1:1 binding reaction:

$$B_{ss,j} = \frac{L_j \cdot I}{k_{D,j}} = \frac{(n_j \cdot L_0 - B_j)\cdot(I_0 - B)}{k_{D,j}}$$

where $B_{ss} = \sum B_{ss,j}$ is the total concentration of bound ligands, $k_{D,j}$ is the dissociation constant for the binding complex B_j, and I_0 is the initial concentration of I domains. Solving this equation for $B_{ss,j}$ and then substituting to $B_{ss} = \sum B_{ss,j}$ provides a J+1 order polynomial for B_{ss}:

$$B_{ss} = L_0(I_0 - B_{ss}) \sum_{j=1}^{J} \{\frac{n_j}{k_{D,j} + I_0 - B_{ss}}\}$$

The steady-state concentration B_{ss} of I domains bound to ligands is the root of this polynomial that lies between 0 and $\min\{L_T, I_0\}$. The steady-state fraction of ligands bound to I domains equals $f = B_{ss}/R_T$.

In the case of J=2 distinct kinds of ligands per molecule, the steady state fraction f of ligands bound to I domains is the root of the following polynomial that lies between 0 and λ:

$$f^3 - (1 + 2\lambda + \kappa_1 + \kappa_2) \cdot f^2 + ((\lambda + \kappa_1) \cdot (\lambda + \kappa_2) + \delta + \lambda) \cdot f + \lambda \cdot \delta = 0 \quad \textbf{[5.3-7]}$$

$$\kappa_j = \frac{k_{D,j}}{L_T}$$

$$\lambda = \frac{I_0}{L_T}$$

$$\delta(\lambda, \kappa_1, \kappa_2) = \frac{n_1}{n_1+n_2}(\lambda + \kappa_2) + \frac{n_2}{n_1+n_2}(\lambda + \kappa_1)$$

where κ_j are non-dimensional numbers that describe the affinity of each kind of ligand for I Domains, λ is a non-dimensional number that describes the ratio of total available I domains to total available ligands, and δ is a non-dimensional number that describes the contribution of each kind of ligand to the total response f.

Figure 5.3.7 shows contours of the steady state fraction f of bound ligands for several cases of κ_1/κ_2 and n_1/n_2. When the concentration of a ligand dominates (e.g. $n_1 \gg n_2$), then the system becomes a 1:1 binding system. In the case of collagen, the numbers of different types of ligands per molecule are similar (between 1 and 4), see Section 5.2.5. For example, type I collagen contains 2 GFOGER, 4 GLOGER and 2 GMOGER ligands per molecule [Siljander et al. 2004].

Due to the large number of unknown parameters (n_1, n_2, k_{D1}, k_{D2}, L_0) involved even in the case where I Domains bind to just J=2 independent sites on collagen it seems hard to estimate reliably all these parameters based on noisy binding experiment data. Therefore, in this thesis, it is chosen to describe the binding response f of CBI I domains with collagen by an equivalent 1:1 binding system of just two parameters: the total number of ligands L_T and an "equivalent" affinity k_{eq}:

$$\kappa_{eq} = \frac{n_1}{n_1+n_2}\kappa_1 + \frac{n_2}{n_1+n_2}\kappa_2$$

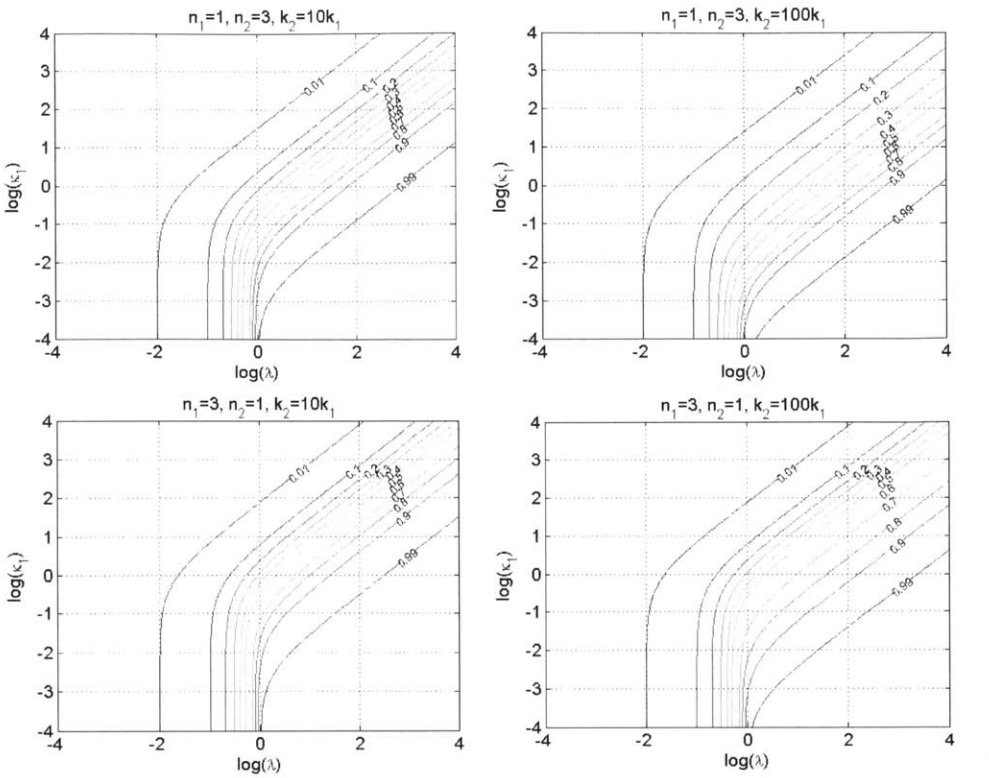


Figure 5.3.7: Contour plot of the fraction of bound ligands $\boldsymbol{f = B_{ss}/L_T}$ in the case of I Domain binding to 2 independent ligand sites on its binding partner L as a function of the non-dimensional binding parameters $\boldsymbol{\kappa_1}$ and $\boldsymbol{\lambda}$, for four cass of $\boldsymbol{n_1/n_2}$ and $\boldsymbol{\kappa_1/\kappa_2}$.

Figure 5.3.8 compares the steady state fraction of bound ligands f for a sample system of J=2 independent binding sites system and of the equivalent 1:1 binding system of L_T sites and k_{eq} affinity parameter. The approximation provides a good estimate of the locus $\lambda_{50}(\kappa_1, \kappa_2)$. Approximation error increases as κ_1/κ_2 deviates away from 1.

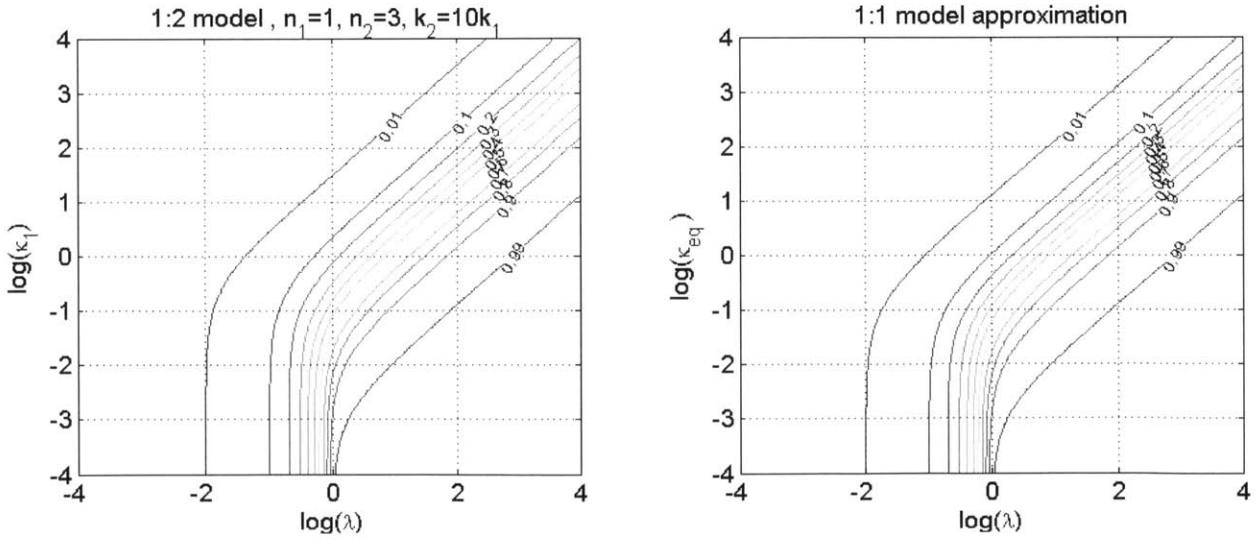


Figure 5.3.8: Approximation of the steady state fraction of bound ligands $\boldsymbol{f = B_{ss}/L_T}$ in the case of I Domain binding to J=2 independent ligands on collagen by a single 1:1 binding system of $\boldsymbol{L_T}$ binding sites and $\boldsymbol{k_{eq}}$ affinity parameter.

1:1 Binding Reaction on a Surface

In the two previous sub-sections, it is assumed that both binding partners L, I and the binding complex B are soluble molecules. However, in many experimental assays, this is not the case. Instead, one of the binding partners (L) is immobilized on a surface while the other partner (I) is dissolved in a solution that is in contact with the surface. When diffusing I molecules bind to immobilized L molecules, they are "caught" so that the resulting B complex is also immobilized on the surface. Examples of such assays include the BIACORE assay (Section 5.3.5), the developed adhesion ligand assay (Section 5.3.7), and the solid phase binding assay.

The geometry of the surface where the binding reaction takes place can be simple or complex. The simplest geometry is a planar surface in contact with a solution, e.g. in the BIACORE assay. An example of a complex geometry is binding of soluble I domains on their adhesion ligands on the surface of a porous biomaterial. In order to focus on the binding reaction, any geometry of a solution of volume V in contact with a surface of area S is modeled as a simpler 1-dimensional geometry, where a planar surface located at coordinate $x = 0$ is in contact with a solution that extends between $x = 0$ and $x = R$, where:

$$R = \frac{V}{S}$$

Molecules I can diffuse in the fluid and can bind to immobilized molecules L located at $x = 0$ assuming a 1:1 binding model. No binding interactions takes place at $x = R$.

A simple 1D mathematical model that describes binding interactions on a surface contains two state variables: the density of unbound ligands on the surface $\rho_B(t)$ (units: [moles/area]) and the concentration $I(x,t)$ of unbound I molecules (units: M), and is set as a system of differential equations.

$$\frac{\partial I(x,t)}{\partial t} = D\frac{\partial^2 I(x,t)}{\partial x^2}$$

$$D\frac{\partial I(0,t)}{\partial x} = k_{on} \cdot I(0,t) \cdot \rho_L(t) - k_{off} \cdot B(t)$$

subject to initial conditions:

$$\rho_B(0) = 0$$

$$I(x, t = 0) = I_0$$

and boundary conditions:

$$\frac{\partial I(R,t)}{\partial x} = 0$$

$$\frac{dB(t)}{dt} = \dot{B}(t) = k_{on} \cdot I(0,t) \cdot \rho_L(t) - k_{off} \cdot \rho_B(t)$$

where D is the diffusion coefficient for the I molecules, k_{on} and k_{off} are the association and dissociation rates (in units of [$M^{-1}s^{-1}$] and [s^{-1}] respectively) of the 1:1 $L + I \leftrightarrow B$ binding reaction. The initial condition $I(x,0) = I_0$ describes the assumption that when the reaction is initiated the concentration of I molecules in solution is constant and equal to I_0. The initial condition $\rho_B(0) = 0$ describes that initially no ligands are bound to I Domains. Conservation of mass imposes the following constraints:

$$\rho_0 = \rho_L(t) + \rho_B(t)$$

$$I_0 \cdot V = (\int_0^R I(x,t)dx + \rho_B(t)) \cdot S \quad \Longrightarrow \quad I_0 \cdot R = \int_0^R I(x,t)dx + \rho_B(t)$$

Where ρ_0 is the total surface density of immobilized L. The first equation describes mass conservation of the immobilized molecule L. The second equation describes the mass conservation of I Domains, which can be found either in solution at spatially-varying concentration $I(x,t)$ or can be immobilized on the surface as a complex with L.

The steady state density of bound ligands can be calculated by setting all time derivatives and spatial derivative to zero, resulting in the following system:

$$I_{ss} \cdot \rho_{L,ss} - k_D \cdot \rho_{B,ss} = 0$$

$$\rho_0 = \rho_{L,ss} + \rho_{B,ss}$$

$$I_0 \cdot R = I_{ss} \cdot R + \rho_{B,ss}$$

Where I_{ss} is the steady state concentration of I Domains in solution. The steady-state fraction of bound ligands $\varphi = \rho_{B,ss}/\rho_0$ is the root of the following polynomial that lies between 0 and 1:

$$\varphi^2 - (1 + \hat{\kappa} + \hat{\lambda}) \cdot \varphi + \hat{\lambda} = 0 \qquad [5.3\text{-}8]$$

$$\hat{\kappa} = R\frac{k_D}{\rho_0}$$

$$\hat{\lambda} = R\frac{I_0}{\rho_0}$$

This polynomial is the equivalent to Equation 5.3-6 derived for the case of 1:1 binding in solution. The non-dimensional parameters $\hat{\kappa}, \hat{\lambda}$ differ from the parameters κ, λ, in that they incorporate the length R, i.e. the ratio of solution volume to surface area. The shape of $\varphi(\hat{\kappa}, \hat{\lambda})$ is identical to the shape of $f(\kappa, \lambda)$ shown in Figure 5.3.6. It is important to note that the steady state fraction of bound ligands φ on the surface does not depend simply on the initial I domain concentration I_0 and the dissociation constant k_D, but also depends on the length R.

The vast majority of published papers on this topic do not include R in the interpretation of surface binding experiments (BIACORE, solid phase binding assays), and estimate k_D as $\hat{k}_D = I_0(\varphi = 0.5)$, that is the value $I_{0,50}$ of I_0 where $\varphi = 0.5$. This can be tricky because the parameter $\hat{\kappa}(R)$ controls the shape of the binding curve $\varphi(I_0)$. In BIACORE assays $\hat{\lambda} \to \infty$ (assuming adequate analyte flow rate) and therefore indeed$I_{0,50} = k_D$. However, when $\hat{\kappa} < 0.1$ then $I_{0,50} = \rho_0/2R \neq k_D$ and the estimate $\hat{k}_D = I_{0,50}$is misleading. This observation may explain why published estimations of k_D for I Domain binding to collagen vary so much (Section 5.2.5). Small values of $\hat{\kappa}$ correspond to either small R or large ρ_0 (large amount of immobilized collagen).

Application of Binding Models in the Developed Adhesion Ligand Density Assay

In the ligand density assay described in Section 5.3.7, the situation seems to be an intermediate between a binding assay in solution and a binding assay on a surface. Although fluorescently labeled I Domains are initially in solution and their ligands are present on collagen molecules immobilized on the scaffold, imaging experiments show that FlAsH-stained TC-tagged I Domains are small enough to diffuse inside the scaffold struts. This is not surprising given that the diameter of an I Domain is approximately 5nm [Emsley et al. 2000]. The detected FlAsH fluorescence signal is usually larger close the surface of the collagen scaffold strut, however there is signal inside the collagen struts.

In order to describe this case, Equation 5.3-9 is modified by expressing the detected signal (concentration of FlAsH-stained I domains) as $\rho_{B,ss} = r \cdot B_{ss}$ and the total density of adhesion ligands as $\rho_0 = r \cdot L_0$, where B_{ss}, L_0 are volume concentrations (units: M), and r^{-1} is the surface volume density of the collagen struts (collagen surface per strut colume). Equation 5.3-10 then becomes identical to equation 5.3-6 (the equation that describes binding in solution) with binding parameters

$$\kappa = G\frac{k_D}{L_0}$$

$$\lambda = G\frac{I_0}{L_0}$$

where the immobilization factor

$$G = \frac{R}{r}$$

describes the ratio of the volume that contains soluble I Domains to the volume that contains the immobilized ligands. In the case of the experimentqal protocol described in Section 5.3.7 the volume of I Domain solution is approximately 25μl, while the volume of the collagen struts that is approximately 0.2μl (0.5% of the volume of ¼ of a cylindrical scaffold of diameter 8mm and height 3mm). The immobilization factor for the ligand density experiment of Section 5.3.7 is on the order of $G = 125$.

The immobilization factor G acts as an amplifier. In an ordinary binding experiment in solution (equation 5.3-6) if the initial concentration of I Domains is I_0, then the concentration of I Domains bound to ligands lies between 0 and $\min(L_0, I_0)$. Due to the immobilization factor G the concentration of I Domains bound to ligands inside the collagen strut can lie between 0 and $\min(L_0, G \cdot I_0)$. This means that the local density of Fluorescent I Domains bound to collagen inside the strut will be higher than the concentration of I Domains in solution. This amplification is a key element that enables distinguishing the signal of I Domains bound to collagen from the signal of the surrounding I Domains in solution. Of course this happens after the dynamics of the solution-surface binding system have equilibrated.

Due to the large value of G, it is expected that both κ and λ parameters will be large, therefore I Domain binding to collagen is described by the top-right part of Figure 5.3.6 In this case the concentration $I_{0,50}$ of I Domains where half ligands are bound to I Domains equals $I_{0,50} = k_D$, i.e. half the ligands in the scaffold will be bound to I Domains when $I_{0,50} = k_D$.

5.3.9 Electron Microscopy

Scanning Electron Microscopy

Dry porous collagen scaffolds were imaged in a Jeol 5600LV scanning electron microscope located in the Department of Materials Science and Engineering (MIT) under guidance by Donald Galler.

Transmission Electron Microscopy

Transmission electron microscopy images of dry CGS were conducted by Nicki Watson (Keck imaging facility, Whitehead Institute, MIT). Dry porous collagen scaffolds were fixed with glutaraldehyde, stained with osmium tetroxide and uranyl acetate, dehydraded in 100% ethanol,

embedded in Spurr's resin polymerized at 50°C, and cut into 50 nm thick sections. Sections were post-stained with uranyl acetate and lead citrate, and imaged in a FEI Technai Spirit Transmission Electron Microscope.

5.4 Results

5.4.1 Recombinant Expression of CBI I Domain

Figure 5.4.1 shows a representative example of a SDS-PAGE gel loaded with purified recombinant I Domains. The gel has been stained with Coomassie Blue. Lanes 1-4 contain I Domains before thrombin cleavage, where the GST tag ("G") is fused with either non-tagged (G-α1, G-α2) or TC-tagged (G-TC-α1, G-TC-α2) I Domains. Lane 5 contains the protein standard (STD). Lanes 6-9 contain I Domains after thrombin cleavage, where the GST tag is removed.

SDS-PAGE reveals that the size of fusion proteins is approximately 50-51 kDa and the size of I domains after cleavage is approximately 23 kDa. These numbers agree with predictions based on the known primary structure of I Domains, the GST tag and the TC tag (see protocol J.1). TC-tagged proteins are 1014 Da larger than non-tagged I Domains, and this is also evident in the gel. All cleaved I Domain fractions apart from α_1I contain small fractions of uncleaved fusion proteins. The impurity of uncleaved fusion proteins is more severe in samples that are harder to cleave (TC-tagged I Domains). All lanes were loaded with 4 µg of protein sample, as estimated based on A280 nm absorption.

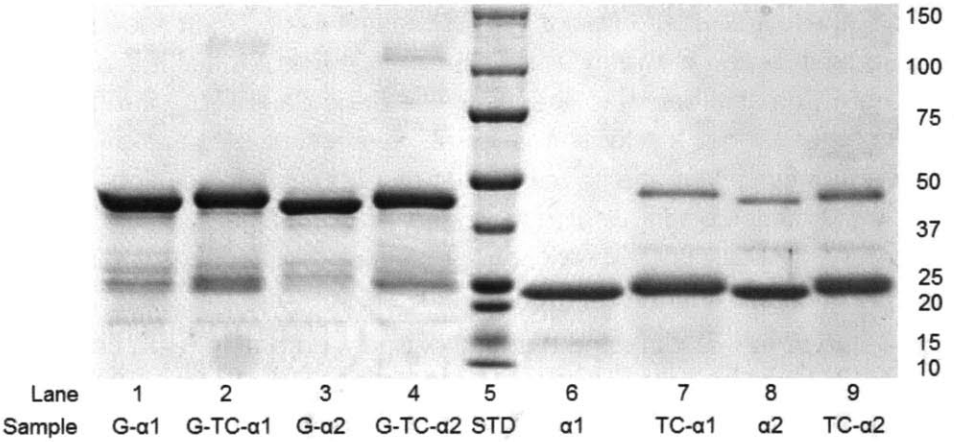


Figure 5.4.1: SDS-PAGE of purified fractions of non-tagged and TC-tagged I domains before and after thrombin cleavage. The gel was stained using coomasie blue. STD: protein standards (sizes on the right in kDa).

Figure 5.4.2 shows a representative example of a SDS-FlAsH-PAGE gel loaded with the outcome of recombinant expression of I Domains (different batch compared to the proteins shown in Figure 5.4.1). Lanes are loaded the same way as the ones of the gel shown in Figure 5.4.1. All samples contain 4 µg protein (based on A280 absorption assay) that were treated first with TCEP and 5% excess FlAsH, before being loaded in the gel. After running, the gel was imaged in a fluorescent imager (bottom image), and then stained with coomasie blue and imaged in a conventional imager (top image). The bottom image shows that FlAsH fluorescent signal is emitted only by TC-tagged I domains, either cleaved I Domains (TC-α_1I, TC-α_2I) or fusion proteins (G-TC-α_1I, G- TC-α_2I). Non TC-tagged I Domains emit no FlAsH signal even though they were treated with FlAsh.

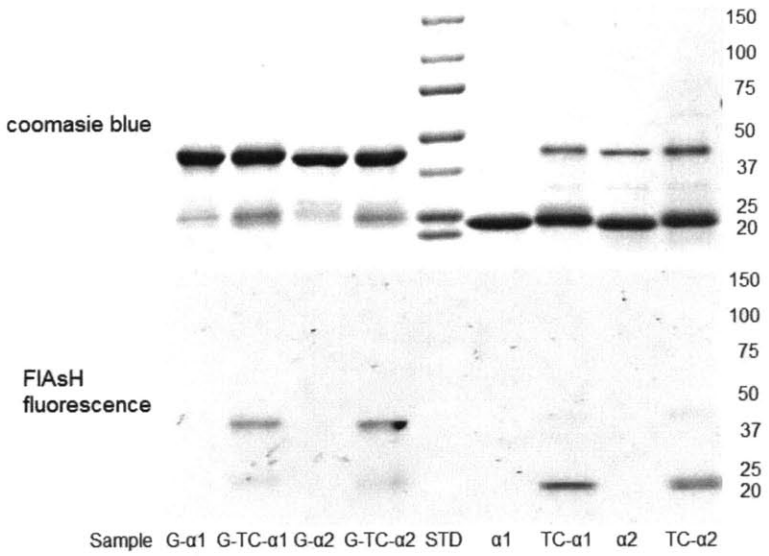


Figure 5.4.2: SDS-FlAsH-PAGE of purified fractions of non-tagged and TC-tagged I domains before and after thrombin cleavage. Samples were reduced by TCEP and stained by FlAsH before denaturing and loading. The gel was initially imaged in a fluorescent imager (bottom). Then it was stained using coomasie blue and imaged in a conventional imager (top). STD: protein standards (sizes on the right in kDa).

In summary Figure 5.4.1and Figure 5.4.2 show that I Domains are purified with reasonable purity. After thrombin cleavage, impurities of approximately 15% (as revealed by densitometry) correspond to uncleaved GST fusion proteins. The A280 nm absorption protein assay seems to provide quite reasonable results, since the signal of all lanes appears to be reasonably similar (within 20%). Treatment with FlAsH before gel loading stained TC-tagged I Domains without staining non-tagged I Domains.

5.4.2 I Domain Characterization: Circular Dichroism Spectroscopy

Results

The first CD measurements evaluated the ability to conduct CD spectroscopy in the purified I Domain solutions, given that I Domains are stored in PBS+3 buffer (PBS supplemented with adding 2mM TCEP, 2mM $MgCl_2$ and 0.02% tween20). It is of interest to see how the addition of TCEP, $MgCl_2$ and tween 20 affect the CD spectrum of the PBS buffer. Figure 5.4.3 shows that the spectrum of the protein buffer (PBS+3) is indistinguishable from the CD spectrum of plain PBS, as well as the CD spectrum of the empty cuvette. The background signal is significantly different when a different cuvette is used. These results suggest that the majority of the background signal is caused by cuvette birefringence.

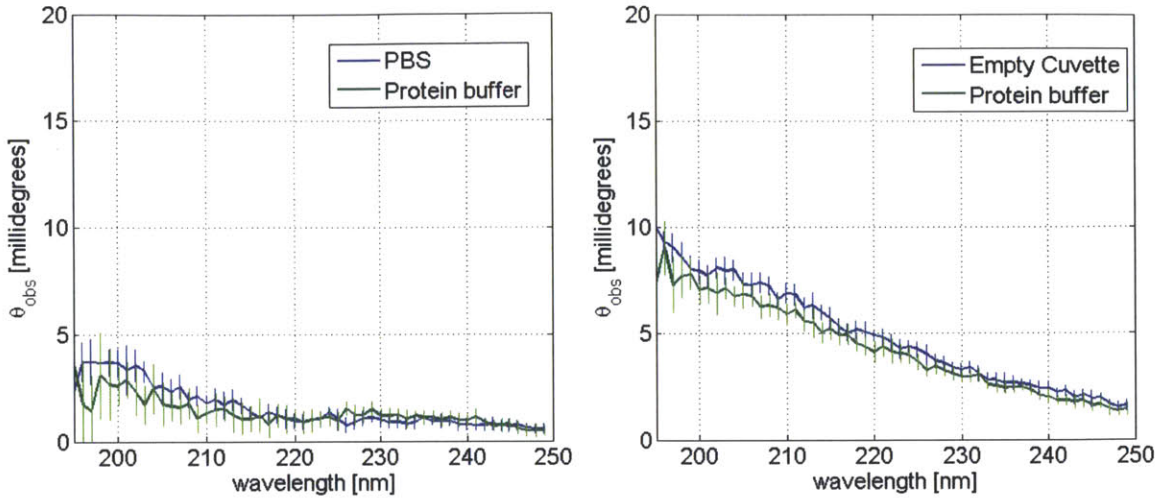


Figure 5.4.3: Evaluation of the magnitude of the background CD signal $\boldsymbol{\theta(\lambda)}$. Left: Comparison of the PBS CD spectrum versus the protein buffer (PBS, 2mM TCEP, 2mM $MgCl_2$ and 0.02% tween20) CD spectrum. Right: Comparison of the empty cuvette CD spectrum versus the protein buffer CD spectrum. The difference between the protein buffer CD spectra observed in the two experiments is due to the use of two different cuvettes. All CD spectra shown in this section have been acquired using the cuvette of right image.

Figure 5.4.4 left shows the measured CD spectra $\theta(\lambda)$ in millidegrees. Even though all solutions have similar 280nm absorbance, the magnitude of the CD spectra $\theta(\lambda)$ is quite different. Figure 5.4.4 right shows that the calculated mean residue ellipticity θ_{mrw} is almost identical for all I domains for λ>205 nm. The CD spectra θ_{mrw} of I domains differ significantly for wavelengths less than 205 nm. Tetracysteine-tagged I domains (TC-α_1I, TC-α_2I) have smaller θ_{mrw} values compared to non-tagged I domains (α_1I, α_2I) suggesting that the presence of the tetracysteine tag somewhat increases the amount of random coil structure.

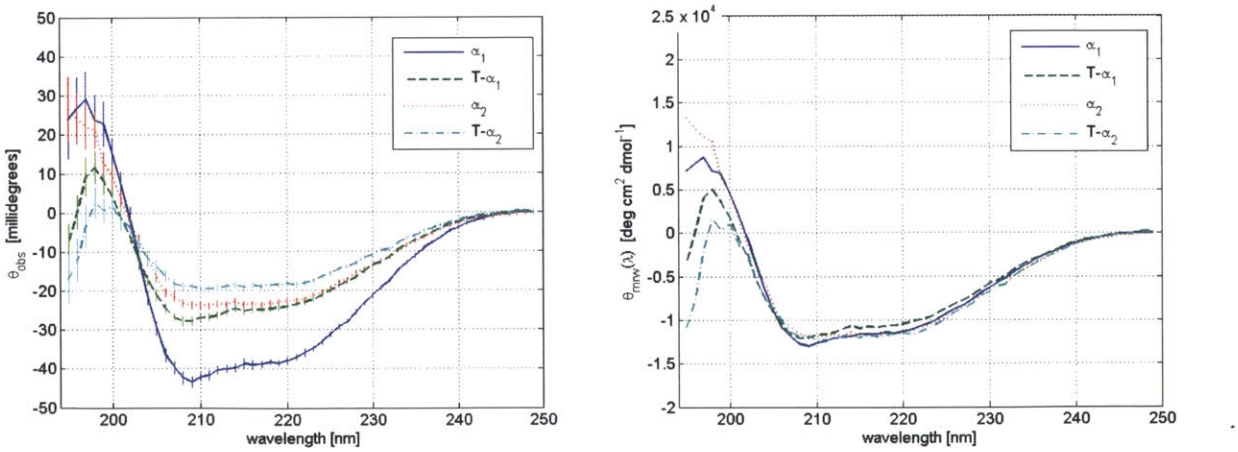


Figure 5.4.4: Circular dichroism spectra of purified I domain solutions. Left: Raw measured ellipticity data (after baseline signal subtraction). Right: Results for the mean residue ellipticity $\boldsymbol{\theta_{mrw}(\lambda)}$.

I domain	Secondary structure	X-ray data*	Estimation based on CD spectra		
			SELCON3	CONTINLL	CDSSTR
α_1I	Helix %	0.39	0.339	0.355	0.407
	Strand %	0.21	0.187	0.189	0.170
	Turn %	0.11	0.217	0.202	0.161
	Random coil %	0.29	0.263	0.253	0.262
	Fitting NRMSD	-	0.207	0.040	0.072
TC-α_1I	Helix %	-	0.312	0.236	0.420
	Strand %	-	0.162	0.230	0.159
	Turn %	-	0.216	0.240	0.184
	Random coil %	-	0.298	0.292	0.236
	Fitting NRMSD	-	0.420	0.047	0.124
α_2I	Helix %	0.390	0.346	0.336	0.376
	Strand %	0.200	0.171	0.184	0.163
	Turn %	0.093	0.211	0.214	0.186
	Random coil %	0.317	0.287	0.265	0.280
	Fitting NRMSD	-	0.149	0.034	0.060
TC-α_2I	Helix %	-	0.258	0.176	0.411
	Strand %	-	0.174	0.201	0.251
	Turn %	-	0.218	0.262	0.145
	Random coil %	-	0.372	0.361	0.192
	Fitting NRMSD	-	0.276	0.105	0.097

Table 5.4-1: Estimations of the contribution of each secondary structure kind (helix, strand, turn, random coil) in each kind of purified I domains based on three computational tools (SELCON3, CONTINLL, CDSSTR) that deconvolve CD spectra. For each fit, the normalized root-mean-square deviation (NRMSD) is provided as a measure of the goodness of fit. *X-ray data for α_1I are based on crystal structure 1QCY [Nymalm et al. 2004] that contains residues 169...361 of ITGA1(P56199). X-ray data for α_2I are based on crystal structure 1AOX [Emsley et al. 1997] that contains residues 168...366 of ITGA2(P17301).

Table 5.4-1 shows the estimated contributions of particular secondary structures (α-helices, β-strands, turns, and random coil (irregular structure)) for each one of the four purified I domains, obtained from three CD analysis software tools (SELCON3, CONTINLL, CDSSTR). The known contributions of secondary structures in α_1I and α_2I (based on the annotations of the known crystal structures 1QCY [Nymalm et al. 2004] and 1AOX [Emsley et al. 1997] provided by the protein data bank (http://www.rcsb.org)) are also shown for comparison. Results show that

- With the exception of TC-α_2I the three algorithms provide similar estimates.
- All three algorithms estimate relatively well the contributions of α-helices and β-strands but overestimate the contributions of β-turns and underestimate the contribution of random coils. CDSSTR estimates for α_1I and α_2I seem to be slightly better compared to SELCON3 and CONTINLL.
- The normalized root mean square deviation (NRMSD) of each fitting seems not to have a practical use, since CONTINLL always has the lowest NRMSD without significantly better accuracy in estimating the secondary structure of α_1I and α_2I.
- The addition of the tetracysteine tag seems to cause slight chances in the contributions of the secondary structures. In general SELCON3 and CONTINLL suggest that the TC tag decreases slightly the contribution of α-helices and increase slightly the contribution of random coils (a reasonable response). CDSSTR suggests the opposite change (increased helical contribution), which does not make sense since the TC tag is known to fold in a turn.

Discussion

Circular dichroism data were acquired for solutions of I domains, with and without a TC tag. Proteins were reconstituted in the same PBS buffer as the one used in subsequent binding experiments (BIACORE, adhesion ligand surface density assay). Data suggest that the buffer and the cuvettes used permitted low noise CD spectra measurements for wavelengths larger than 200nm (Figure 5.4.3 and Figure 5.4.4). The most likely source for the significant noise observed below 200nm is light absorption by salts of the PBS buffer, dissolved oxygen, or the quartz cuvette [Greenfield 2006]. It is possible that different I domain concentrations (resulting in A280 different than 0.17) could improve the signal to noise ratio.

The detected CD spectra for the four I domains are almost identical for λ>205 nm and vary for λ<205 nm (Figure 5.4.3). Since CD spectra for λ>210nm are dominated by the α-helical contribution, data could suggest that the TC tag does not affect significantly the presence of the nine α-helices present in I domains. However, TC-tagged I domains show reduced ellipticity around 200nm, suggesting increased random coil contribution. Due to the presence of the α-helix breaking PG residues between two cysteine pairs, TC motifs have hairpin 3D structure as shown by a recent NMR study [Madani et al. 2009]. This suggests that adding the 9-residue WDCCPGCCK tag to an I domain (210 residues approximately) should slightly decrease the contributions of α-helices and β-sheets, and increase the contributions of turns and random coil.

The three CD analysis tools provide similar predictions of the secondary structure, with the note that CDSSTR provides increased estimates of the helix contribution. The estimated contributions of secondary structures for the tagged-free I Domains (α_1I, α_2I) agree reasonably with the known contributions provided by the x-ray structures. Compared to TC-free α_1I, the TC-tagged TC-α_1I is predicted to have slightly more contribution of turns and random coil, which agrees with the published structure of the TC tag. On the other hand, the three predictions for TC-α_2I differ more compared to each other, perhaps due to the presence of more contaminants in the protein solution. The mean of the three predictions suggests that adding the TC tag to α_2I reduces helix contribution by 7% and favors mostly random coil structure. Overall these results supports that the addition of the TC tags do not disturb the secondary structure of I Domains significantly, and that the predicted secondary structure of I Domains agrees very well with the one provided by x-ray crystallography.

5.4.3 I Domain Characterization: BIACORE Binding Assay

The objectives of the BIACORE experiments are: i) to evaluate the binding affinity of the purified I Domains for their adhesion ligands in collagen molecules, ii) to evaluate the effect of the tetracysteine (TC) tag on the binding affinity of I Domains for their ligands. Indirectly, the BIACORE assay acts as a way to evaluate the correct folding of I Domains after their recombinant expression and purification, since only well-folded I Domains are expected to bind their ligands.

Processing BIACORE data can be implemented by two approaches. The first approach is to manually condition the raw signal provided by the instrument ($R_{21}^{raw}(t)$ and $R_{43}^{raw}(t)$) in order to calculate first the baseline-corrected sensorgrams $R_{21}^{bs}(t)$ and $R_{43}^{bs}(t)$, then the corrected sensorgrams $R_{21}(t)$ and $R_{43}(t)$, and then fit the later into a 1:1 binding model using generic nonlinear optimization algorithms (Section 5.3.5). The second approach is to use the BIAeval software tool (provided by BIACORE) that automatically calculates the baseline-corrected

sensorgrams $R_{21}^{bs}(t)$, $R_{43}^{bs}(t)$ and then fits them into a 1:1 binding model that also incorporates signal jumps.

Baseline-corrected Sensorgrams

Figure 5.4.5 and Figure 5.4.6 show the calculated baseline-corrected sensorgrams that correspond to binding of α_1I-based analytes and α_2I-based analytes respectively to collagen and gelatin. Each figure contains four images, that show the baseline-corrected sensorgrams $R_{21}^{bs}(t)$ (left column) and $R_{43}^{bs}(t)$ (right column) for non-tagged I Domains (top row) and TC-tagged I Domains (bottom row). For clarity only the three-four sensorgrams that correspond to larger analyte concentration are shown. Baseline-corrected sensorgrams were calculated in MATLAB using the procedure described in Section 5.3.5.

Figure 5.4.5 and Figure 5.4.6 show that there are signal jumps in the beginning of the association and dissociation steps. The magnitude of signal jumps depends on the identity and concentration of the analyte. Signal jumps are larger in analyte solutions of increased concentration. Signal jumps are also larger in solutions of analytes whose stock solutions are more dilute (stock TC-tagged I Domains are more dilute than non-tagged I Domains; α_2I-based I Domains are more dilute than α_1I-based I Domains), therefore require less dilution in running buffer. Both observations support that these signal jumps induced by the properties (e.g. RI and ionic strength) of the stock analyte buffer (PBS+3) and the running buffer (HBS-P+ plus 1 mM MgI_2) used to prepare the dilutions. Larger signal jumps correspond to solutions where the analyte buffer has larger mass fraction compared to the running buffer. The relationship between the analyte concentration and the mass fraction of PBS+3 buffer in the solution is not linear due to dilution math.

Figure 5.4.5 and Figure 5.4.6 also show that the signal jumps are smaller in $R_{43}^{bs}(t)$ compared to $R_{21}^{bs}(t)$. This also supports the idea that these signal jumps are due to matrix effects because the matrix of chambers 2, 4 (collagen immobilized on dextran) is more different than the matrix of chamber 1 (dextran only) compared to the matrix of chamber 3 (gelatin immobilized on dextran).

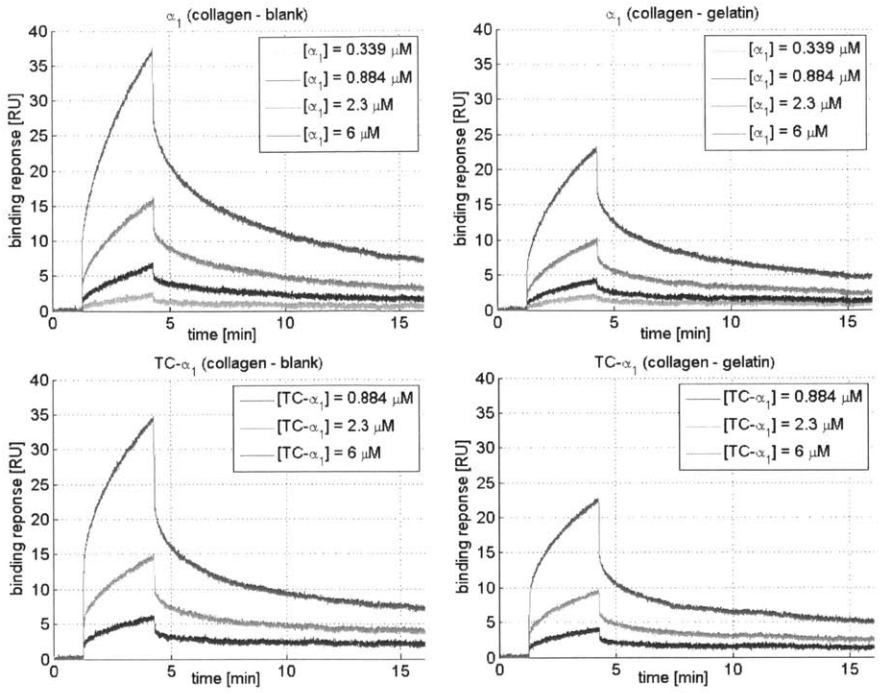


Figure 5.4.5: Baseline-corrected sensorgrams $\boldsymbol{R_{21}^{bs}(t)}$, $\boldsymbol{R_{43}^{bs}(t)}$ for analyte solutions of I Domains of integrin α1. Top row: Results for solutions of non-tagged I Domains α1I. Bottom row: Results for solutions of TC-tagged I Domains TC-α1I. Left Column: Results for $\boldsymbol{R_{21}^{bs}(t)}$ (collagen surface minus blank surface). Right Column: Results for $\boldsymbol{R_{43}^{bs}(t)}$ (collagen surface minus gelatin surface).

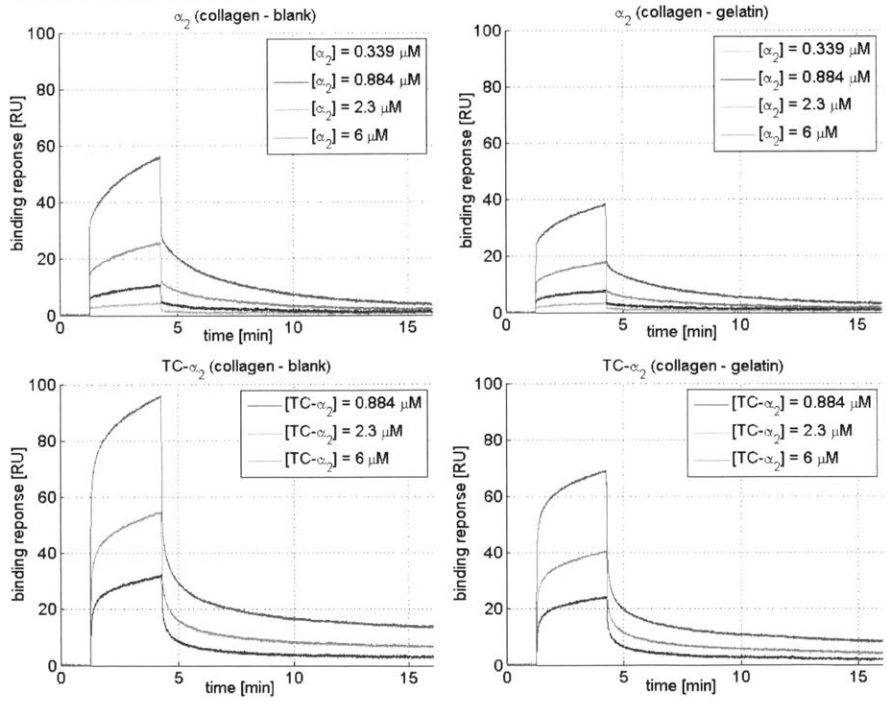


Figure 5.4.6: Baseline-corrected sensorgrams $\boldsymbol{R_{21}^{bs}(t)}$, $\boldsymbol{R_{43}^{bs}(t)}$ for analyte solutions of I Domains of integrin α_2. Top row: Results for solutions of non-tagged I Domains α_2I. Bottom row: Results for solutions of TC-tagged I Domains TC-α_2I. Left Column: Results for $\boldsymbol{R_{21}^{bs}(t)}$ (collagen surface minus blank surface). Right Column: Results for $\boldsymbol{R_{43}^{bs}(t)}$ (collagen surface minus gelatin surface).

Corrected Sensorgrams and Global Fitting

Figure 5.4.7 and Figure 5.4.8 show the calculated corrected sensorgrams that correspond to binding of α_1I-based analytes and α_2I-based analytes respectively to collagen and gelatin. Each figure contains four images, that show the corrected sensorgrams $R_{21}(t)$ (left column) and $R_{43}(t)$ (right column) for non-tagged I Domains (top row) and TC-tagged I Domains (bottom row). For clarity only the three or four sensorgrams that correspond to larger analyte concentration are shown. Corrected sensorgrams were calculated in MATLAB from baseline-corrected sensorgrams (Figure 5.4.5 and Figure 5.4.6) after subtracting the contribution of the signal jumps at the beginning of the association and dissociation steps, as described in Section 5.3.5.

Figure 5.4.7 and Figure 5.4.8 also provide two parameters of the 1:1 binding model (dissociation constant $k_D = k_{off}/k_{on}$, number of adhesion ligands per immobilized molecule n) estimated by global fitting of the corrected sensorgrams. A detailed list of the 1:1 binding system parameters (k_{on}, k_{off}, n) obtained by global fitting of the corrected sensorgrams $R_{21}(t)$, $R_{43}(t)$ can be found in the "global fitting" columns of Table 5.4-2. Figure 5.4.7 and Figure 5.4.8 provide the binding responses (dashed lines) that correspond to the parameters estimated by global fitting based on Eq. 5.3-5.

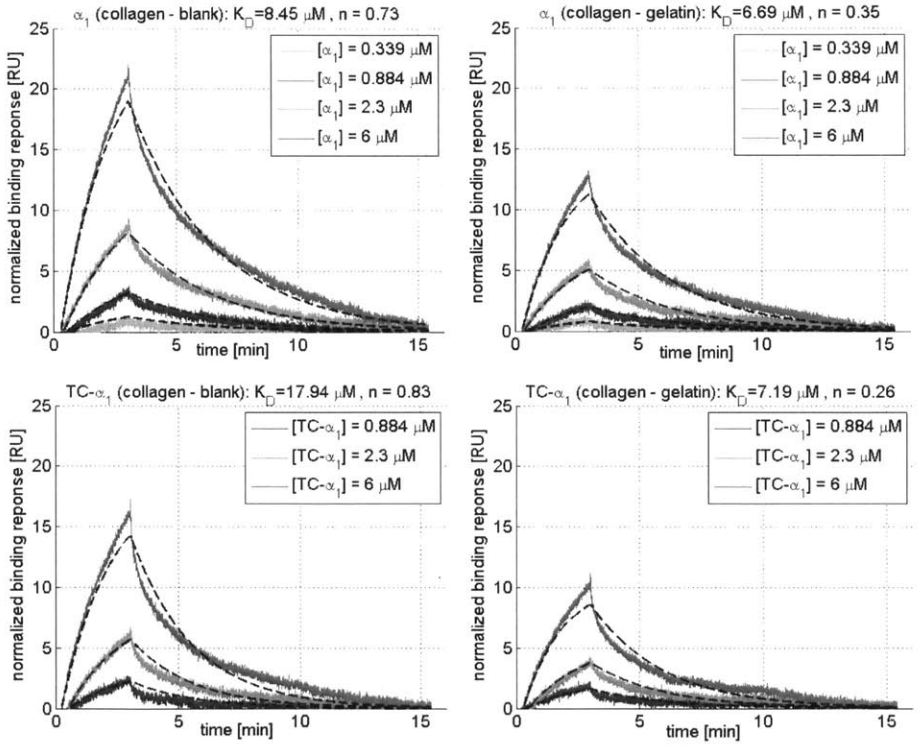


Figure 5.4.7: Corrected sensorgrams $\boldsymbol{R_{21}(t)}$, $\boldsymbol{R_{43}(t)}$ and global fits (dashed lines) to a 1:1 binding model for analyte solutions of I Domains of integrin α_1. Top row: Results for solutions of non-tagged I Domains α_1I. Bottom row: Results for solutions of TC-tagged I Domains TC-α_1I. Left Column: Results for $\boldsymbol{R_{21}(t)}$ (collagen surface minus blank surface). Right Column: Results for $\boldsymbol{R_{43}(t)}$ (collagen surface minus gelatin surface). The title of each image provides the dissociation constant $\boldsymbol{k_D}$ and the number of adhesion ligand sites per collagen molecule $\boldsymbol{n}$ estimated from the corrected sensorgrams by global fitting.

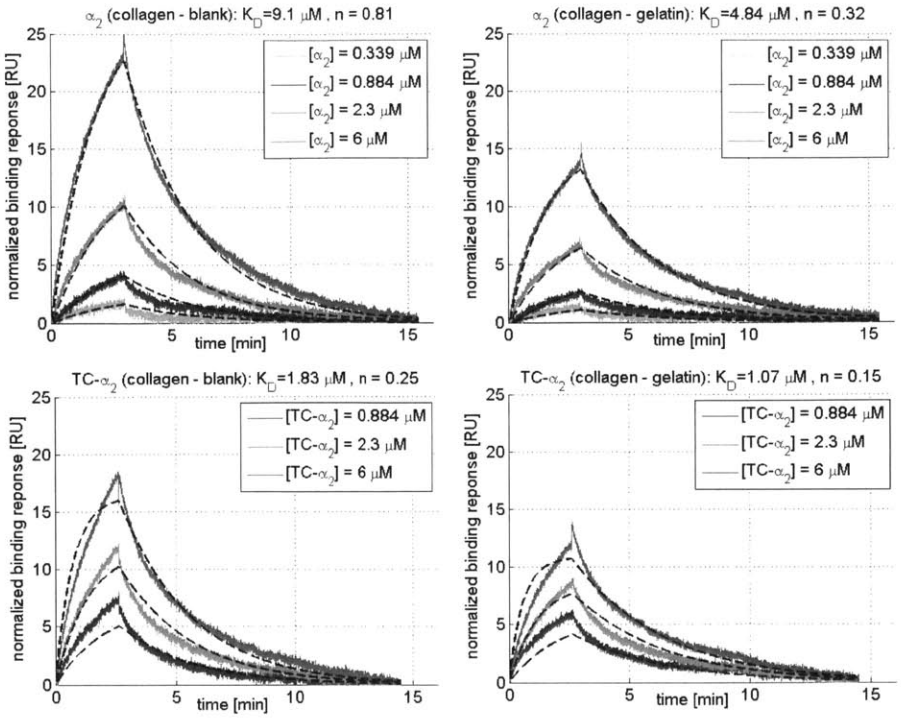


Figure 5.4.8: Corrected sensorgrams $\boldsymbol{R_{21}(t)}$, $\boldsymbol{R_{43}(t)}$ and global fits (dashed lines) to a 1:1 binding model for analyte solutions of I Domains of integrin α_2. Top row: Results for solutions of non-tagged I Domains α1I. Bottom row: Results for solutions of TC-tagged I Domains TC-α1I. Left Column: Results for $\boldsymbol{R_{21}(t)}$ (collagen surface minus blank surface). Right Column: Results for $\boldsymbol{R_{43}(t)}$ (collagen surface minus gelatin surface). The title of each image provides the dissociation constant $\boldsymbol{k_D}$ and the number of adhesion ligand sites per collagen molecule $\boldsymbol{n}$ estimated from the corrected sensorgrams by global fitting.

Overall, the shape of corrected sensorgrams of the same type $R_{ij}(t)$ for TC-tagged TC-α_iI and non-tagged α_iI shown in Figure 5.4.7 and Figure 5.4.8 appear quite similar (similar rates of association and dissociation kinetics, similar maximum signal within the same association period). Unfortunately, the three minute duration of the association period is not enough to reach the steady-state signal that corresponds to each analyte concentration (estimated k_{on} rates range between 300 and 1000 $M^{-1}s^{-1}$ with the exception of TC-α_2I, see later discussion, Table 5.4-2). On the other hand, the 12 minute duration of the dissociation period seems adequate for complete dissociation of the bound analyte (estimated k_{off} rates range between 440 and 600 s^{-1}, Table 5.4-2). The corresponding dissociation constant k_D, a metric of binding affinity lies in the 5-20 μM range. Global fits seem to match well with all binding curves that correspond to different concentrations of the analyte, with the exception of TC-α_2I.

Binding curves for TC-α_2I had the largest signal jumps at the beginning of the association and dissociation steps compared to all other analytes, and the largest residual signal in the steady state of the dissociation step, (Figure 5.4.5 and Figure 5.4.6). These artifacts could be due to matrix effects or protein aggregation artifacts described in Section 5.3.5. The reason was that TC-α_2I was the I Domain that was hardest to purify and cleave using thrombin, resulting in the lowest yield and most dilute stock solution (approximately 10 to 12 μM TC-α_2I in PBS+3). TC-α_2I solutions were also the most sensitive to protein aggregation, and required careful handling (Section 5.3.2). Global fits for TC-α_2I data seem quite worse compared to global fits for the remaining analytes. The estimated association rates k_{on} are approximately 5 times faster

compared to the corresponding non-tagged I Domain (α_2I), and the simulated response curves do not agree well with measured curves. It is possible that these errors are induced by impact compensation of artifacts related to matrix effects or protein aggregation [Schuck et al. 1999].

In all cases, the corrected sensorgrams $R_{43}(t)$ (collagen minus gelatin) have approximately 30% less magnitude compared to the corrected sensorgram $R_{21}(t)$ (collagen minus blank). Gelatin is used as a negative control molecule that has the same primary structure as collagen, is expected to lack its adhesion ligands due to the loss of the triple helical conformation (Section 5.2.3), and hopefully has the same non-specific binding properties. The difference between the $R_{43}(t)$ and $R_{21}(t)$ signals suggests that most of the binding detected in $R_{21}(t)$ is specific to collagen and does not take place in gelatin. It is not clear if the difference can be attributed to different non-specific binding or slight affinity of CBI I domains for adhesion ligands in gelatin. Global fits for $R_{43}(t)$ result in larger association rates k_{on}, smaller dissociation rates k_{off} (resulting in smaller dissociation constant k_D), and fewer adhesion sites per collagen molecule n compared to $R_{21}(t)$, which agree with the hypothesis that the signal $R_{43}(t)$ lacks some part of non-specific binding to collagen that is included in signal $R_{21}(t)$. Nevertheless this cannot be proved based on the presented data.

The presence of the TC tag qualitatively seems to decrease slightly the corrected sensorgrams $R_{21}(t)$ and $R_{43}(t)$ compared to the corresponding non-tagged I Domain. Compared to α_1I data, TC-α_1I data are fitted by 1:1 binding models of slightly slower association rates k_{on}, slightly larger dissociation rates k_{off} (resulting in larger dissociation constant k_D). Compared to α_2I data, TC-α_2I data are fitted by 1:1 binding models of similar dissociation rates k_{off}. The must larger association rates k_{on} for TC-α_2I (resulting in smaller dissociation constant k_D) are probably caused by artifacts and are not considered meaningful.

So far, the discussion in this Section deals with results calculated using the signal conditioning and global fitting procedure described in Section 5.3.5. Figure 5.4.9 shows a representative result of the global fitting procedure carried in the BIAeval software. It shows the baseline-corrected data $R_{21}^{bs}(t)$ as calculated by the BIAeval software, and the resulting global fit using a 1:1 binding model augmented with signal jumps.

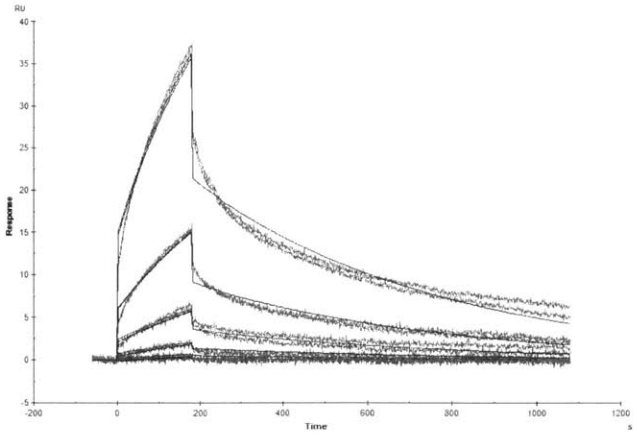


Figure 5.4.9: Baseline-corrected sensorgrams $\boldsymbol{R_{21}(t)}$ (noisy curves) and global fits (thin smooth lines) to a 1:1 binding model for analyte solutions of α_1I calculated using the BIAeval software.

A comparison of the baseline-corrected data of Figure 5.4.9 with the baseline-corrected data of Figure 5.4.5 (top left) calculated based on the procedure described in Section 5.3.5, highlights their similarity. A comparison between the properties of the 1:1 binding model calculated by the global fitting approach described in Section 5.3.5, or by the global fitting algorithms of the BIAeval software is shown in Table 5.4-2. BIAeval software predicts faster association rates k_{on}, slower dissociation rates k_{off} and similar R_{max} values (calculated from n using Eq. B4c) most probably due to different way of modeling the signal jumps in the beginning of association and dissociation steps.

Analyte	Data	Global Fitting					BIAeval Fitting			
		k_{on} [$M^{-1}s^{-1}$]	k_{off} [$10^{-5}s^{-1}$]	k_D [μM]	n	R_{max} [RU]	k_{on} [$M^{-1}s^{-1}$]	k_{off} [$10^{-5}s^{-1}$]	k_D [μM]	R_{max} [RU]
α_1I	2-1	555	469	8.45	0.73	63.2	359	178	4.97	77.3
	4-3	744	497	6.69	0.35	30.3	1320	131	0.99	17.8
TC-α_1I	2-1	363	652	17.94	0.83	71.9	1422	96	0.67	20.3
	4-3	796	572	7.19	0.26	22.5	1231	86	0.69	14.2
α_2I	2-1	619	563	9.10	0.81	70.1	831	262	3.15	44.2
	4-3	994	481	4.84	0.32	27.7	1013	216	2.14	23.5
TC-α_2I	2-1	3048	558	1.83	0.25	21.7	1644	124	0.76	38.3
	4-3	4134	440	1.07	0.15	12.9	1960	141	0.72	24.9

Table 5.4-2: Results of the parameters for the 1:1 binding model of I domains to collagen minus blank (2-1) or collagen minus gelatin (4-3). Results are provided for two kinds of computations: i) computations using global nonlinear optimization of baseline-corrected data whose jump artifacts are removed manually, and ii) computations sing the BiaEval software.

Discussion

The presentation of the results highlights some of the difficulties involved in processing and interpreting BIACORE data. The presence of signal artifacts such as mass transfer, matrix effects, RI differences or protein aggregates can affect significantly the shape of sensorgrams and the resulting global fits. In this study, all curves were reasonably well globally fitted into 1:1 binding models apart from the TC-α_2I analyte, where the large content of analyte buffer in the analyte solutions induced severe artifacts that result in non-convincing global fits. Nevertheless even for the TC-α_2I analyte some qualitative comments can be made.

The motivation for globally fitting the data using 1:1 binding models was to provide a qualitative rather than quantitative description of binding of I Domains to collagen, and how this is affected by the TC tag. The results obtained by globally fitting for the association and dissociation rates k_{on}, k_{off} (and the resulting dissociation constant k_D) can be used only as a coarse estimation. Although it is known that collagens contain multiple binding sites for I Domains, the BIACORE experiment as designed and implanted in this thesis cannot resolve the properties of the individual I Domain - adhesion ligand binding, since data contain enough artifacts that cannot be adequately removed.

A further point that complicates the analysis of BIACORE data is that I Domains in solution are found in two conformations ("open" and "close") and one of them ("open") seems to have much larger affinity for its ligands compared to the other (see Section 5.2.4). Previous experiments suggest that in solution 20% to 40% of α_LI are in "open" configuration [Vorup-Jensen et al. 2007], which would result in increasing the estimated dissociation constant k_D by 2.5 to 5 times.

Overall, the shape of BIACORE curves suggest that the majority (approximately 2/3) of I Domains bound on immobilized collagen does not bind to the negative control molecule (gelatin). Therefore most of I Domain bound to collagen is collagen specific and does not bind to gelatin. The presence of the TC tag seems to slightly worsen the affinity of the I Domain for its ligands. The TC does not seem to prevent I Domains from binding their ligands, does not cancel the specificity of I Domain binding to collagen (rather than gelatin), and does not reduce the affinity of I Domains for collagen significantly (e.g. at least one order of magnitude). Evidence for the effects of the TC tag is stronger for TC-α_1I compared to TC-α_2I, where more severe artifacts make global fitting more challenging.

The reported affinities of 5 to 20 μM for CBI I Domain binding to collagen I show some level of agreement with previous BIACORE measurements of I Domain (α_1I, α_2I) binding to collagen I (summarized in Table 5.2-3, Section 5.2.5), however comparisons need to be carefully. The previous publication that estimated the affinity of CBI I Domains for collagen I based on kinetic analysis of BIACORE data suggests k_D = 24 nM for binding of α_1I to collagen I, and k_D = 180 nM for binding of α_2I to collagen I [Calderwood et al. 1997]. Such tight affinities seem questionable given the experiment design of that study: the very small applied flow (5μl/min) induces significant mass transfer that affect kinetics, and the very slow dissociation rates estimates (which result in the very small values of dissociation constants) were based on a dissociation step of very small duration that can induce significant estimation error and does not reveal what is the steady state response of the dissociation step. Several different studies attempted to estimate the affinities of CBI I Domains for collagen I from BIACORE using the steady-state response of the sensorgram in the end of the association step, and analyzing results using Scatchard plots assuming a binding model that includes one or two kinds of independent 1:1 binding sites. Their findings suggest one kind of sub-micromolar affinity k_D binding site (0.09 to 0.32 μM for α_1I, 0.26 μM for α_2I) and one micromolar affinity k_D binding site (5.5 to 13.9 μM for α_1I, 3.99 to 10.7 μM for α_2I) [Rich et al. 1999; Xu et al. 2000; Kim et al. 2005]. The μM binding sites of these models usually correspond to 4 to 6 more binding sites per molecule n compared to the sub-μM sites [Rich et al. 1999; Xu et al. 2000] therefore seem to affect the binding response more. Overall these estimates agree with the findings of this study. Finally, several studies studied the affinities of CBI I Domains for collagen I using solid-phase binding assays (SPBA). These studies model binding using a single kind of 1:1 binding, and report much smaller dissociation constants (23 to 260 nM) for α_1I binding to collagen I and widely varying results (20 nM, 54 nM and 5800 nM for α_2I binding to collagen I), see Table Table 5.2-3. There is no clear explanation for the disagreement between BIACORE results and SPBA results.

The result of the BIACORE experiments presented in this study could be improved in several ways. One way would be to infer the binding properties by using the steady state response of the association step, instead of trying to fit kinetic data as attempted here. This approach may be less sensitive to artifacts. In this case, analyte flow rates could be further reduced, leading to savings of valuable analyte. Another critical way would be to change the buffer of the I Domain stock solutions into something closer to the running buffer (HBS-P+ plus 1 mM MgI_2) in order to reduce matrix effects and RI difference effects. Buffer change requires some optimization, as it can affect the rate of I Domain aggregation, see Section 5.3.2. Finally, it seems advisable to reduce the amount of collagen or gelatin immobilized on the CM5 chip, since the amount

immobilized in this study (1200 RU) is suggested to be quite high for kinetic experiments and can induce mass transfer and matrix effects [Karlsson and Fält 1997; Myszka 1997; Schuck et al. 1999].

5.4.4 Quantifying the Density of CBI Ligands on Porous Collagen Biomaterials

Imaging Results

Figure 5.4.10 shows representative scanning electron microscopy images of the collagen scaffolds considered in this study. Scanning electron microscopy reveals the porosity of the scaffold, which qualitatively agrees with the published mean pore diameter of approximately 100 μm expected based on the chosen fabrication protocol [O'Brien et al. 2004]. Transmission electron microscopy reveals that scaffold struts have amorphous structure and there is only occasional presence of D-banded collagen [Yannas 1972; Gelse et al. 2003]. While type I collagen found in tissues is organized in fibrils that are "banded" (collagen is in a semi-crystalline state), during the fabrication of the scaffold this semi-crystaline state seems to be lost due to collagen fibril swelling in 0.05M acetic acid (protocol F.1).

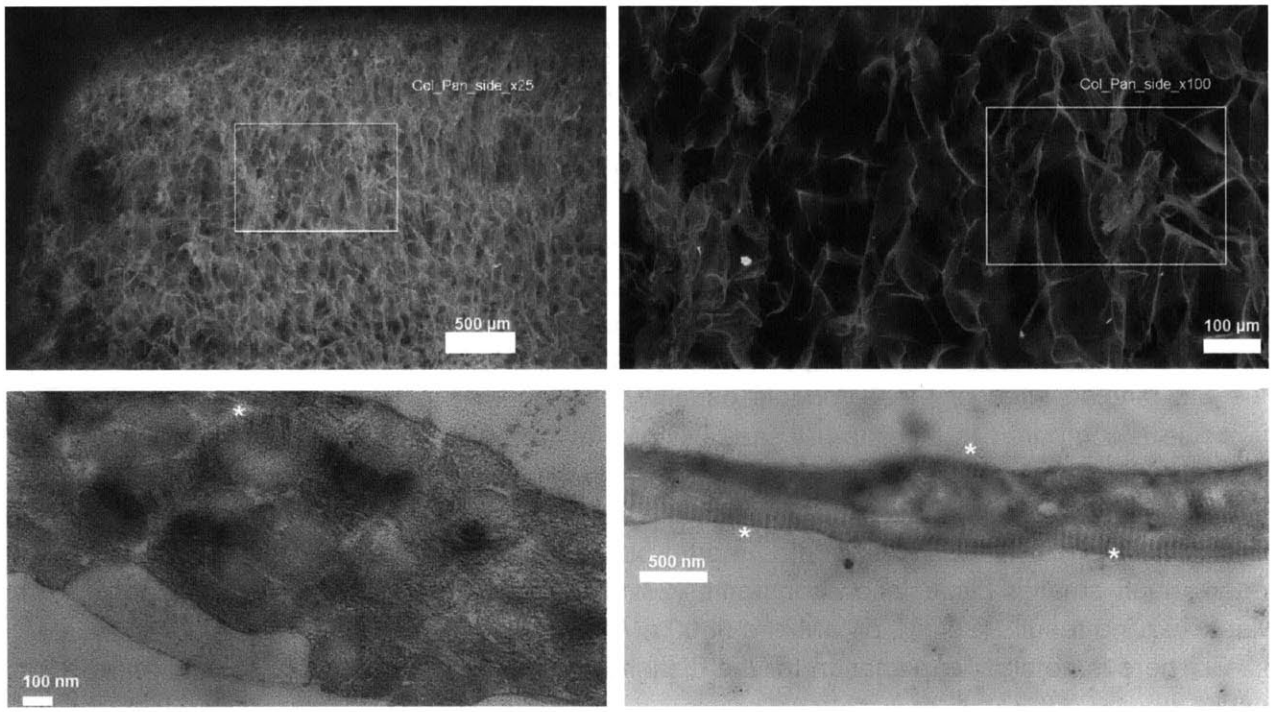


Figure 5.4.10: Electron microscopy images of porous collagen scaffolds highlight its porous structure and provide information about the conformation of collagen molecules. Top row: Scanning electron microscopy on dry porous collagen scaffolds. Bottom row: Transmission electron microscopy of thin (50nm) sections of fixed porous collagen scaffolds. Asterisks correspond to locations where the D-banding of collagen is evident.

Figure 5.4.11 and Figure 5.4.12 show representative images of three kinds of porous collagen scaffolds (A, D, and E) treated with 6 μM FlAsH-stained TC-α_1I and TC-α_2I respectively. Raw spectral multiphoton images are processed by the pipeline described in Chapter 2 in order to unmix and segment the images. The images in the left columns of Figure 5.4.11 and Figure 5.4.12 show the spectral unmixing outcome when the samples are treated with TC-α_iI in the

presence of $MgCl_2$. The images in the right columns show the spectral unmixing outcome when the samples are treated with TC-α_1I in the presence of EDTA. All unmixed images dispay the three emission sources present in the system (collagen fluorescence, FlAsH fluorescence, collagen second harmonic emission) using the same colormap.

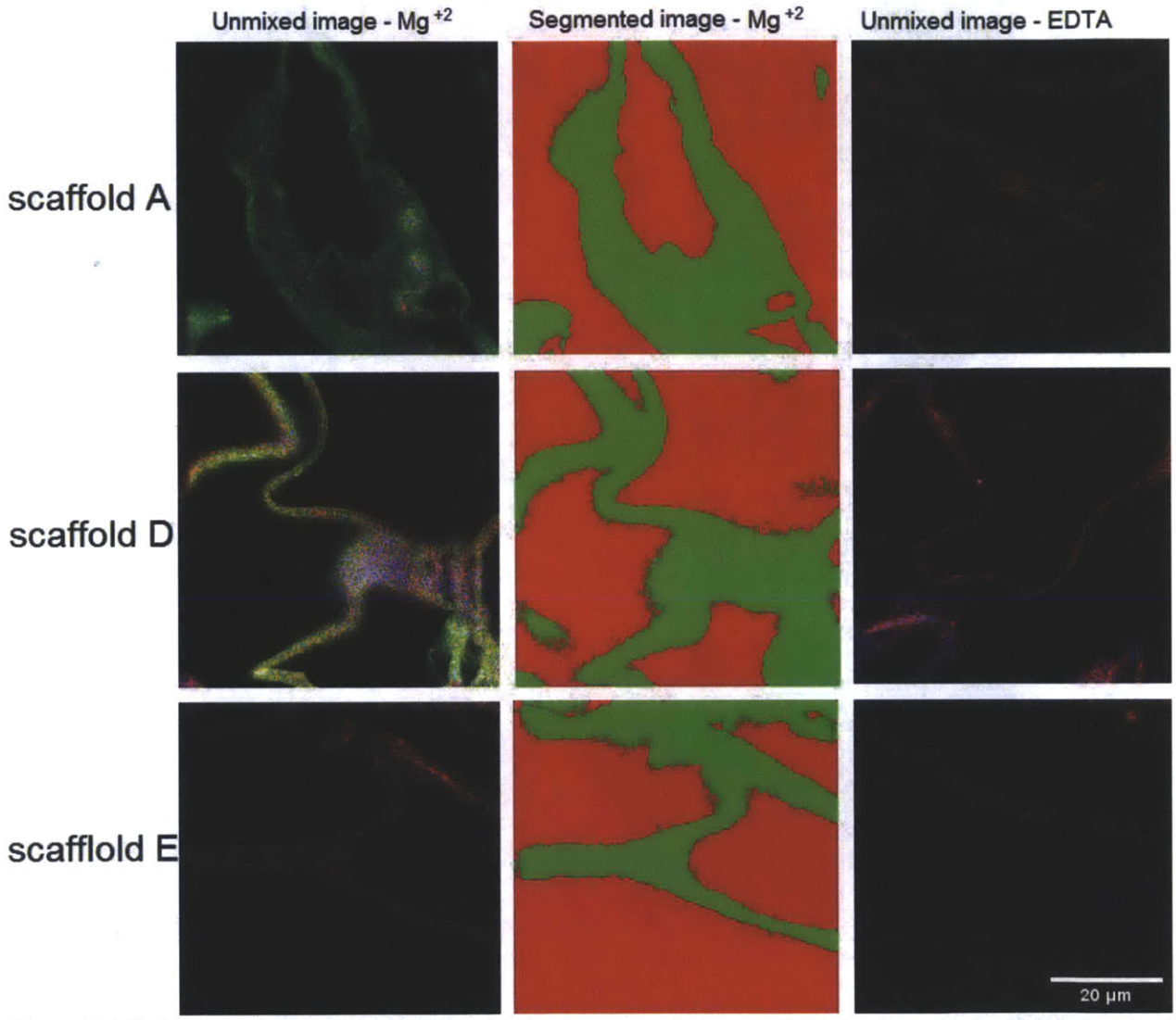


Figure 5.4.11: Representative images of three kinds of porous collagen scaffolds treated with 6 µM FlAsH-stained TC-α_1I domains in the presence of 2 mM $MgCl_2$ (left column) or 10 Mm EDTA (right column). 16-channel spectral multi-photon images have been spectrally unmix to resolve the signals of collagen autofluorescence (red; range: 1-10 photons/pixel), FlAsH (green; range: 5-150 photons/pixel) and second harmonic emission (blue; range: 1-4 photons/pixel). Middle column: the corresponding image segmentation maps (green: scaffold, red: solution) of the $MgCl_2$. All images are 55×55 µm wide.

The images of the middle columns of Figure 5.4.11 and Figure 5.4.12 show the outcome of the local image segmentation algorithms (class vidence calculations, see Section 2.5.2) scaffold and solution classes. The resulting segmentation is used to define pixels that belong to the scaffold outer surface, used in the following calculations. In the presence of 2 mM $MgCl_2$,

fluorescent I Domains stain the whole volume of the scaffold. In the presence of EDTA FlAsH emission from the scaffold is much less and is localized mostly at its outer surface.

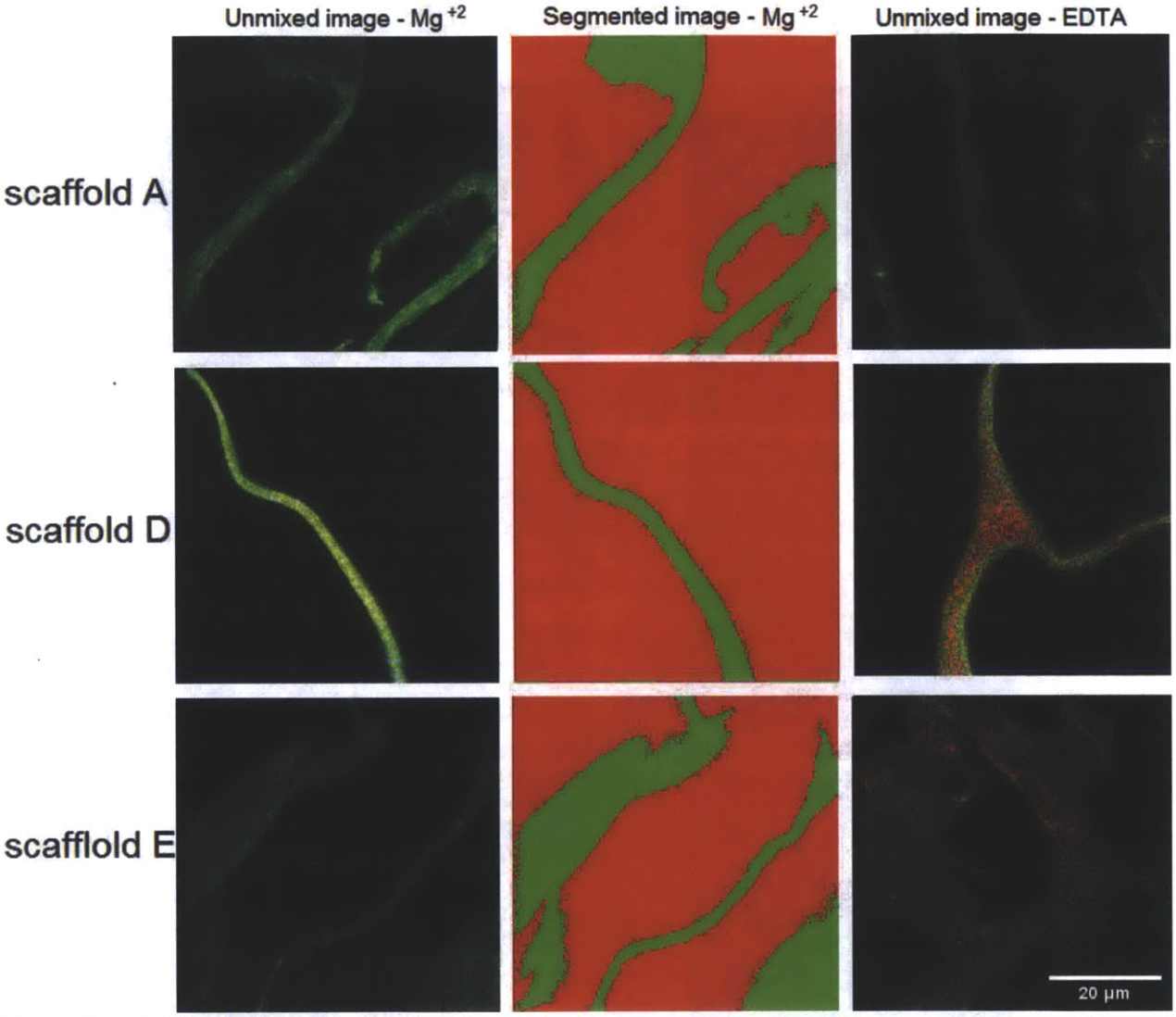


Figure 5.4.12: Representative images of three kinds of porous collagen scaffolds treated with 6 µM FlAsH-stained TC-α_2I domains in the presence of 2 mM $MgCl_2$ (left column) or 10 Mm EDTA (right column). 16-channel spectral multi-photon images have been spectrally unmix to resolve the signals of collagen autofluorescence (red; range: 1-10 photons/pixel), FlAsH (green; range: 5-150 photons/pixel) and second harmonic emission (blue; range: 1-4 photons/pixel). Middle column: the corresponding image segmentation maps (green: scaffold, red: solution) of the $MgCl_2$. All images are 55×55 µm wide.

I Domains Bind to Collagen Scaffolds in a Mg^{+2}-dependent and Specific Manner

Figure 5.4.13 shows the fluorescence emission rate of FlAsH on the surface of porous collagen scaffolds "D" treated with solutions of various concentration of TC-α_1I or TC-α_2I in the presence of either 2 mM Mg^{+2} or 10 mM EDTA. As discussed in Section 5.3.2 it was decided not to purify TC-tagged I domains in concentrations larger than 10-20 µM in order to prevent protein precipitation. Therefore the concentration of TC-tagged I Domain solutions was chosen in the region 0.75-6 µM. Results show that the FlAsH signal on the scaffold surface in the presence of Mg^{+2} is significantly larger than the signal in the presence of EDTA. These results agree with

previous knowledge that highlights the necessity of divalent cations for I Domain binding to collagen (Section 5.2.3).

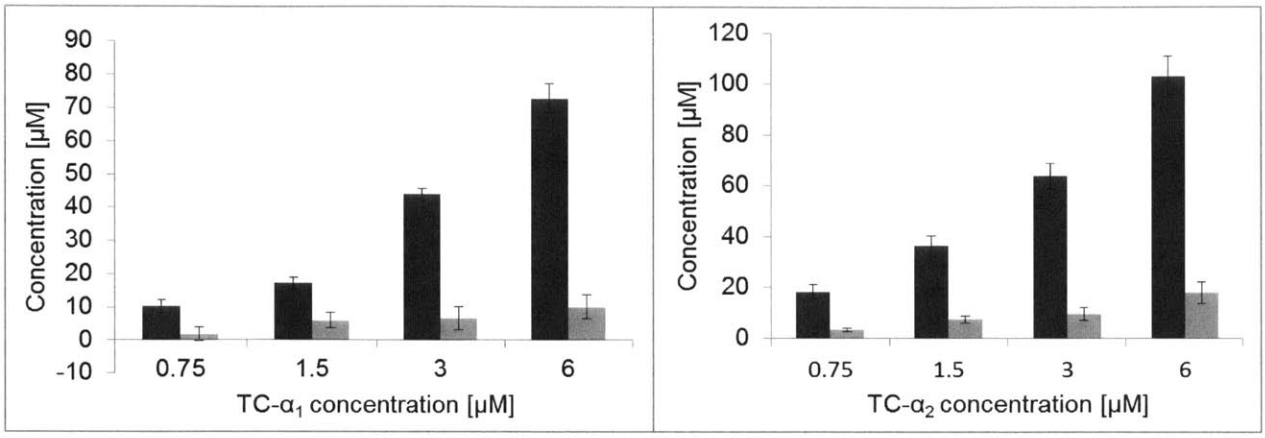


Figure 5.4.13: Concentration of bound I Domains on the surface of the porous collagen scaffold "D" when the scaffold is treated with various concentrations of FlAsH-stained TC-tagged I Domains. Dark grey bars correspond to signals in the presence of 2 mM Mg^{+2}. Light grey bars correspond to signals in the presence of 10 mM EDTA.

Figure 5.4.14 shows the fluorescence emission rate of FlAsH on the surface of porous collagen scaffolds "D" when the emission in the presence of EDTA is subtracted from the signal in the presence of Mg^{+2}. Since I Domain binding is Mg^{+2}-dependent, the signal in the presence of EDTA can be treated as mostly mon-specific and therefore it needs to be subtracted from the signal in the presence of Mg^{+2}. Results show that in both cases, solutions of TC-tagged I Domain of 6 μM concentration do not result in signal saturation.

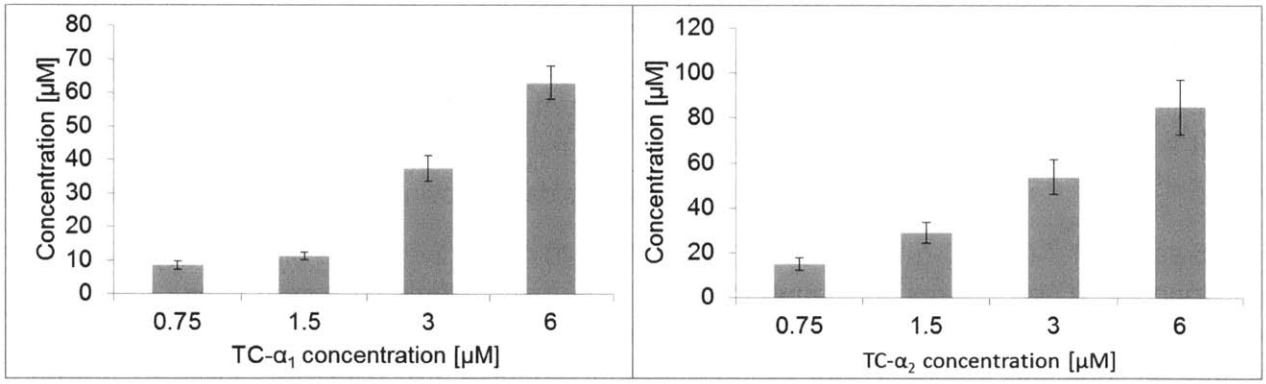


Figure 5.4.14: Concentration of bound I Domains on the surface of the porous collagen scaffold "D" when the scaffold is treated with various concentrations of FlAsH-stained TC-tagged I Domains. The results show the difference of the signal in the presence of Mg^{+2} minus the signal in the presence of EDTA.

Figure 5.4.15 shows the fluorescence emission rate of FlAsH on the surface of porous collagen scaffolds "D" when scaffolds are treated with solutions of constant FlAsH-stained TC-tagged I Domain concentration and increasing non-tagged I Domain concentration. Results highlight the fact that for both I Domain kinds, as the concentration of the non-tagged I Domain increases, the concentration of TC-tagged I Domains on the surface of the scaffold decreases. This signal decrease suggests that FlAsH-stained $T\alpha_iI$ and non-tagged I Domains α_iI bind to the same ligands, therefore TC-tagged binding to collagen scaffold is specific. Results suggest that the

dissociation constant of the TC-α_1I is approximately 8-10 μM, and that the dissociation constant of the TC-α_2I larger, probably around 15 μM.

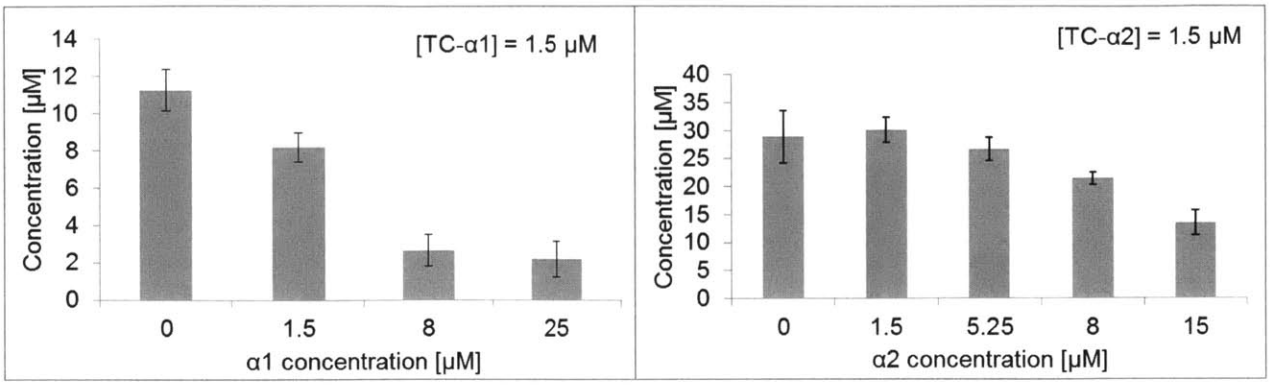


Figure 5.4.15: Concentration of bound I Domains on the surface of the porous collagen scaffold "D" when the scaffold is treated with 1.5μM FlAsH-stained TC-tagged I Domain and various concentrations of non-tagged I Domains. The results show the difference of the signal in the presence of Mg^{+2} minus the signal in the presence of 10 mM EDTA.

Quantifying the Total Density of Adhesion Ligands in Collagen Scaffolds

Based on the last part of Section 5.3.8, the total number of adhesion ligands (units: [M]) on the material can be estimated based on an equivalent 1:1 binding model as:

$$L_T = 4 \cdot k_D \cdot \left.\frac{dB_{ss}}{dI_0}\right|_{I_0=k_D}$$

where B_{ss} is the measured concentration of I Domains bound on the biomaterial (corrected data shown in Figure 5.4.14), $\frac{dB_{ss}}{dI_0}$ is the slope of the $B_{ss}(I_0)$ curve, and k_D is the dissociation constant for I Domain binding to collagen. Based on the BIACORE data (Section 5.4.3) and the competitive binding data of Figure 5.4.15, an equivalent dissociation constant for the binding of TC-α_1I and TC-α_2I to collagen is on the order of $k_D = 8$ μM and $k_D = 15$ μM respectively. The slopes $\left.\frac{dB_{ss}}{dI_0}\right|_{I_0=k_D}$ are numerically estimated from the data shown in Figure 5.4.14 and Figure 5.4.15 as 8.5±1.7 (for TC-α_1I) and 10.4±2.5 (for TC-α_2I) respectively. The slightly larger value of the slope for TC-α_2I agrees with analytic predictions that systems of worse affinity (larger k_D) have stiffer slope around $I_0 = k_D$. The resulting estimates of the total density of adhesion ligands for integrins $\alpha_1\beta_1$ and $\alpha_2\beta_1$ in the collagen scaffolds are: $L_T(\alpha_1\beta_1)$ 204.9±41 μM and $L_T(\alpha_2\beta_1)$ 248.3±61 μM respectively.

Quantifying the LD in Different Porous Collagen Materials

Figure 5.4.16 shows the fluorescence emission rate of FlAsH on the surface of porous collagen scaffolds of three different kinds ("A", "D", and "E") when treated with identical solutions of TC-tagged I Domains. Type "A" scaffolds are collagen scaffolds that have not been cross-linked. Type "D" scaffolds are collagen scaffolds that have been cross-linked by dehydro-thermal trearment for 48 hours at 120°C (Protocol F.3). Type "E" scaffolds are collagen scaffolds that have been cross-linked chemically by EDAC-NHS (Protocol F.4). Results show that "E" scaffolds contain much fewer ligands for integrins $\alpha_1\beta_1$ and $\alpha_2\beta_1$ compared to "D" scaffolds.

Results also suggest that type "A" scaffolds contain 25% less ligands for integrin $\alpha_1\beta_1$ and 10% less ligands for $\alpha_2\beta_1$ (not statistically significant) compared to "D": scaffolds.

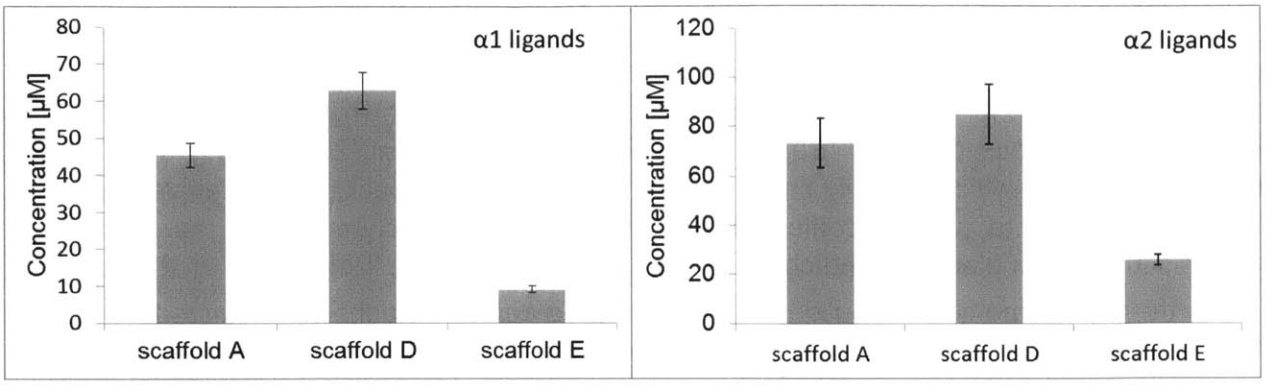


Figure 5.4.16: FlAsH signal (expressed as equivalent concentration of ligands) on the surface of the porous collagen scaffold "D" when the biomaterial is treated with 1.5μM FlAsH-stained TC-tagged I Domain and various concentrations of non-tagged I Domains. The results plotted are the difference of the signal in the presence of Mg^{+2} minus the signal in the presence of 10 mM EDTA.

Assuming that the cross-linking treatment (DHT, EDAC-NHS) does not change the identity of ligands but just their density, these results suggest that scaffolds E contain statistically significantly fewer ligands for both $\alpha_1\beta_1$ and $\alpha_2\beta_1$ (14.4% and 30% respectively) compared to the DHT treated scaffold "D".

5.5 Discussion

The insoluble environment (matrix) of cells is a critical modulator of cell phenotypes in physiology and pathology. Quantifying the molecular details of how the matrix modulates cell behavior is critical for in-depth understanding of critical biological processes (wound healing), diseases (fibrosis, cancer) and for axiomatic principle-based design of biomaterials. This effort has two parts: i) define a state vector for the matrix that includes the features of the matrix that can be sensed by cells, and ii) define how cells respond to the matrix state.

Quantification of the matrix state is hard. Cells can sense a wide range of matrix properties, including its chemical composition, its stiffness [Engler et al. 2004; Vogel and Sheetz 2006], and its topology in the nano and the micro level [Maheshwari et al. 2000; Griffith and Swartz 2006].

Chemical composition is a critical part of the matrix state. The surface chemistry of the matrix (adhesion ligands for particular adhesion receptors present on the matrix surface) defines what kind of cell receptors can be utilized by cells, and therefore what kind of stimuli are provided by the matrix to the cell. Unfortunately the chemical composition is hard to quantify: native ECM in tissue is composed of a large number of multi-domain crosslinked and hard to isolate bio-molecules [Naba et al. 2012]. Existing proteomic methods require digesting and homogenizing the matrix, and then applying modifications of standard biochemical/proteomics methods such as mass spectroscopy. However, matrix digestion destroys the 3D structure of the matrix and provides an average answer that may differ from what cells really feel. Clearly there is a need to describe the chemical composition of the matrix, from a cell-based perspective, that preserves the 3D structure of the matrix.

This Chapter presents a novel optical-based method for quantifying the surface chemistry of a 3D matrix *in situ*. The method relies on using fluorescently-labeled soluble markers that resemble the binding behavior of an adhesion receptor of interest, imaging the binding interaction between the matrix and the marker by 3D microscopy, and applying appropriate image analysis and binding reaction analysis to calculate the density of adhesion ligands present in the image for the adhesion receptor of interest. This methodology is a major advance compared to previous methods [Barber et al. 2005; Harbers et al. 2005; Kong et al. 2006; Hsiong et al. 2008] which quantify the surface chemistry in very specific biomaterials based on RGD peptides, or to spectroscopic methods that quantify the chemistry in the surface of materials in terms of elements and chemical groups [Ma et al. 2007; Kingshott et al. 2011].

In this thesis, the method is applied in quantifying the density of adhesion ligands for the two major collagen-binding integrins ($\alpha_1\beta_1$, $\alpha_2\beta_1$) in three kinds of porous collagen materials used in nerve regeneration studies, aiming to gain some understanding about why these materials induce so different wound healing outcomes in animal models of peripheral neve regeneration [Soller et al. 2012].

5.5.1 Development of Fluorescent CBI I Domain Markers

A critical aspect of the proposed methodology is the development of soluble markers that resemble the binding properties of the adhesion receptor of interest. In the case of the two major collagen-binding integrins (CBI: $\alpha_1\beta_1$, $\alpha_2\beta_1$) such biomarkers exist, and consist of the I Domains of their α subunit [Hynes 2002].

For experimental purposes two kinds of markers are required in binding experiments: fluorescently-labeled markers and non-labeled markers. Since the vast majority of proteins are not fluorescent when excited by imaging laser lines, it is necessary to develop fluorescent versions of the markers by fusing the marker with an appropriate peptide or protein. Due to the small size of I Domains (25 kDa) and the dependence of I Domain function on large conformation changes during binding to its ligands, fluorescent versions of I Domains were constructed by fusing them with a small tetra-cysteine motif at their N terminus and treating them with small biarsenical dyes (Section 5.3.6). All markers were expressed in BL21 e-coli as GST fusions and were purified by affinity chromatography and subsequent GST removel (Section 5.3.2).

Extensive biochemical characterization quantified the purity, size, structure and binding properties of the TC-tagged and non-tagged I Domains (Sections 5.3.3, 5.3.4, and 5.3.5). SDS-PAGE-FlAsH analysis revealed that only TC-tagged I Domains (and not non-tagged I Domains) bind to the small biarsenical dye FlAsH and become fluorescencent (Section 5.3.3). Circular dichroism analysis revealed that the secondary structure of non-tagged I Domains agrees with published x-ray results and that the presence of the TC tag does not alter the secondary structure significantly (Section 5.3.4). BIACORE analysis revealed that I Domains bind to collagen with affinity on the order of 8 – 15 μM, that they have significantly more affinity for collagen at its correct triple-helical state compared to its denatured state (gelatin), and that the presence of the TC tag has a small effect on the binding affinity of I Domains (Section 5.3.5). In all steps of purification and characterization the effect of the TC tag on the α_1 I Domain was less severe compared to the effects on the α_2 I Domain.

5.5.2 Measuring the Density of CBI Ligands on Porous Biomaterias *in Situ*

Several publications present biochemical studies on the binding interactions of CBI I Domains with ECM molecules such as collagens, and laminin (Table 5.2-3). However, most ECM molecules are normally not found in the soluble state. Instead after being expressed by cells they are organized into insoluble structures (fibrils, fibers, meshes) that form the complex macro-molecular network of the matrix (tissue ECM or biomaterials). So far, only a single biochemical study has considered binding of I Domains to collagen fibrils [Jokinen et al. 2004], where collagn molecules are organized based in the semicrystalline array that results in the formation of D-banding [Orgel et al. 2006; Sweeney et al. 2008]. This study concluded that fibril organization affects the avidity of I Domains binding and affects the way cells sense and spread on collagen. This study supports the need to develop methods that can quantify the surface chemistry of a matrix *in situ* since ECM molecule organization may affect the way integrins recognize and bind their ligands.

This study presents the first methodology that quantifies the surface chemistry on a 3D matrix *in situ*. Specifically, the TC-tagged and non-tagged I Domains are utilized to quantify the density of ligands for integrins $\alpha_1\beta_1$ and $\alpha_2\beta_1$ available on the surface of porous collagen scaffolds *in situ*. This is implemented by conducting an *in situ* imaging-based binding assay, where scaffolds are treated with various concentrations of FlAsH-stained TC-tagged I Domains and non-stained non-tagged I Domains, and then imaged by spectral imaging microscopy. The image processing pipeline described in Chapter 2 is then utilized to recover the spatial concentration of FlAsH on the surface of the scaffold. Results show that binding of stained TC-tagged I Domain to the scaffold surface is concentration-dependent and divalent-cation dependent. The detected signal is much larger in the presence of $MgCl_2$ (specific signal) compared to the signal in the presence of EDTA (control), Figure 5.4.13. This divalent cation dependence agrees with previous published reports on the I Domain [Tuckwell et al. 1995; Calderwood et al. 1997; Tulla et al. 2001]. The results of the competitive binding experiment suggest that binding of stained TC-tagged I Domains to the scaffold surface is also specific, since the detected signal of bound stained I Domains decreases as the concentration of non-tagged I Domains increases, Figure 5.4.15. The shape of competitive binding curves is consistent with the equivalent dissociation constants (approximately 8 μM for α_1I, 15 μM for α_2I) estimated based on the BIACORE assay. Imaging results were carefully interpreted into binding parameters (number of ligands, dissociation constants) using a 1:1 binding model that combines the properties of binding experiments in solution and on a surface (Section 5.3.8).

The developed methodology was applied in quantifying the density of ligands for the two major CBI ($\alpha_1\beta_1$, $\alpha_2\beta_1$) in three kinds of porous collagen scaffolds, used in a recent study of induced regeneration in peripheral nerves (PN) [Soller et al. 2012]. The three materials consist of the "active" material "D" (induces improved regeneration in severely injured PN) and the "inactive" scaffolds "A" and "E" (induce limited regeneration in severely injured PN). The three materials differ in their cross-linking treatment. A is not cross-linked, D is cross-linked by DHT, E is

crfosslinked by the chemical cross-linkers EDC and NHS. Scaffold D is similar to ones used for inducing regeneration in peripheral nerves.

The results show that "active" porous collagen scaffolds D contain 204.9±41 μM $\alpha_1\beta_1$ ligands and 248.3±61 μM $\alpha_2\beta_1$ ligands. Inactive scaffolds A contain approximately 25% less $\alpha_1\beta_1$ ligands and approximately 10% less $\alpha_2\beta_1$ ligands (not statistically significant) compared to scaffold D. This difference between the density of ligands of scaffolds A and D could be the result of partial denaturation of the collagen molecules in scaffold A, since cross-linking delays scaffold denaturation and scaffolds were imaged approximately 1 month after fabrication. Scaffolds A are known to have a very short *in vivo* half life (much less than 1 week) [Harley et al. 2004; Soller et al. 2012].

On the other hand, inactive scaffolds E contain significantly less ligands for both $\alpha_1\beta_1$ and $\alpha_2\beta_1$ integrins (85.6% and 70% less respectivel) compared to scaffolds D. This reduction is hypothesized to be the result of the chemical cross-linking trestment applied to scaffold E. Specifically, the cross-linker EDC is known to attack and activate carboxyl groups turning them reactive for primary amines [Hermanson 2008]. Since the ligands of CBI contain acidic residues (aspartic acid, glutamic acid) and the carboxylic groups of their side chains actively participate in formation of the bonds between CBI I Domains and collagen (Sections 5.2.4 and 5.2.5), it is expected that EDC treatment can destroy these ligands, and therefore deplete scaffolds E of their surface chemistry.

These results suggest that binding interactions between the collagen scaffold and cells inside the wound may be of critical importance for the ability of a biomaterial to induce regeneration. In the case of scaffold A, although this "inactive" uncrosslinked material contains as many ligands as the "active" scaffold D, it degrades quickly *in vivo* so the ligands disappear. In the case of scaffold E, the scaffold degrades very slowly *in vivo*, however it contains much fewer ligands compared to scaffold D. The low density of ligands on scaffold E can explain why contractile cells located just inside its inner lumen do not interact extensively with scaffold E: these cells seems to prefer to bind each other, become contractile (as shown by the significant upregulaton of actin) and are aligned circumferentiallu, parallel to the inner lumen surface of the scaffold. This structure suggests that the contractile forces applied by these cells are not transmitted to the scaffold. Furthermore, often these cells lose contact with the scaffold, as evident by the much smaller outer diameter of the capsule compared to the scaffold inner diameter. In contrast, in transected PN grafted with active scaffold D, the cells locatedin the outer surface of the nerve make always contact with the scaffold and do not get a preferential circumferential orientation. Further experiments are required to provide quantitative evidence about the effect of ligand density to particular cell phenotypes *in vitro* and *in vivo.*

5.6 Extensions of the Methodology and Future Work

The developed methodology provides a new tool that can be used to quantify the insoluble microenvironments felt by cells inside a 3D matrix. Instead of the standard approach to describe the chemistry of a matrix based on the composition of ECM molecules, the proposed methodology quantifies the density of ligands of particular adhesion receptors, which can be present in more than one kind of ECM molecules.

The methodology can be applied to quantify various kinds of 3D matrices including tissue ECM and biomaterials. The application of the methodology in biomaterial characterization follows the approach described in this Chapter. The methodology could be applied to biomaterials based on either natural biomolecules or artificial polymers functionalized with peptides/domains that contain adhesion ligands. This is in contrast to the existing specialized methods of quantifying RGD peptides, which can be applied only to a very limited range of artificial materials.

At the moment, cell-scaffold interactions are described by quantifying how cell phenotypes of interest depend on particular structural (mean pore size), mechanical (stiffness), and chemical (composition) properties of the scaffold [O'Brien et al. 2005; Harley et al. 2008; Yannas et al. 2010]. A useful approach to study the effect of a particular material property is by generating a librariy of scaffolds consisting of members that differ only in the property of interest, whose level varies smoothly across the scaffold members of the library. The methodology developed here introduces the concept of a small scaffold library with members that contain a variable density of ligands of particular adhesion receptors. Scaffold libraries in combination with high-throughput screening by proteomic/genetic assays or imaging (for example the method described in Chapter 3) [Hubbell 2004; Gallagher et al. 2006] can provide a quantitative description of the effects of the scaffold surface chemistry on particular phenotypes of interest.

The methodology can be also applied to quantify the surface chemistry of tissue ECM. Tissue ECM can be thought of as a very complex (in terms of chemical composition) information-rich scaffold that regulates cells [Badylak 2007]. The methodology developed here could be applied to quantify the surface chemistry of tissue ECM, after cellular components have been removed via a decellularization process [Uygun et al. 2010; Badylak et al. 2011]. This introduces opportunities for quantifying the surface chemistry felt by cells in healthy tissue and comparing it with the surface chemistry during pathological conditions, or for quantifying how the chemical environment felt by cells changes during the progression of diseases associated with significant changes in the ECM, for example cancer [Olumi et al. 1999; Lu et al. 2012].

In this study, the method is applied for quantifying the density of ligands of the two major collagen-binding integrins ($\alpha_1\beta_1$, $\alpha_2\beta_1$). The method can be extended and used to quantify the ligands of various cell adhesion receptors. This requires two steps:

- Identifyting a soluble protein that can act as a marker of the adhesion receptor of interest. An appropriate marker should have approximately the same specificity and affinity for ligands as the receptor. Identifying or developing such markers requires understanding at the molecular scale of the structure and function of the receptor of interest. Sometime this task is not hard. For example the marker of integrins that bind their ligands via the I Domain of their α subunit is that particular I Domain. Examples of such integrins include the two less-abudant collagen-binding integrins $\alpha_{10}\beta_1$, $\alpha_{11}\beta_1$. The development of markers for integrins that bind their ligands using both subunits (e.g. the RGD integrins $\alpha_3\beta_1$, $\alpha_5\beta_1$, $\alpha_8\beta_1$) is harder, and probably requires engineering an appropriate fusion of the β propeller of the α subunit with the I domain of the β subunit (see Figure 5.2.1). The development of such markers requires an understanding of the domain structure and conformation changes induced by binding, because it is possible that appropriate modifications are necessary so that the fusion protein can be in an "open" state. One example of such fusion protein design is the α_8-β_1 fusion protein design developed to identify ligands of integrin $\alpha_8\beta_1$ [Marciano et al.

2007]. Apart from integrins it is possible to seek for markers of adhesion receptors that belong to other families, for example DDR collagen receptors and sugar-binding receptors. Markers of DDR receptors can be based on the diskoidin domain of DDR receptors. As markers for particular sugar oligomers, it is possible to use sugar-binding lectins (WGA, SBA) that are already commercially available.

- Generate fluorescent markers. This can be tricky as the fluorescence entity should interfere minimally with the structure and the conformation changes that regulate the specificity and affinity of the marker for its ligands. Fluorescent markes can be engineered by fusing the non-fluorescent marker with a peptide or domain that will make it fluorescent, or by chemically conjugating the non-fluorescent markers with reactive fluorescent dyes [Hermanson 2008].

Finally, even though the implementation of the method described in this Chapter utilized spectral multiphoton microscopy, it is possible to apply the methodology using other 3D imaging techniques. Specifically, the method can be implemented using confocal microscopy, an imaging modality that is nowadays widely available. Finally, it could be combined with emergent methods of super-resolution, such as PALM, STORM and STED (although some of them require conjugating the markers with particular fluorophores) [Hell and Wichman 1994; Betzig et al. 2006; Rust et al. 2006].

Last but not least, a major impact of the developed methodology is that it can be modified in order to develop methods to edit the surface chemistry of a biomaterial. In combination with emerging 3D printing tools, such methods could lead to the development of biomaterials whose surface chemistry can edited in 3D at will [Derby 2012; Miller et al. 2012], in order to provide appropriate chemical clues to the various kinds of cells that will interact with the material at various locations.

5.7 Acknowledgements

This study would not have been feasible without the valuable help of many people at MIT. Prof. Peter Dedon (Dept. Biological Engineering, MIT), Prof. Dianne Newman (Dept. Biology, MIT), Prof. Richard Hynes (Dept. Biology, MIT), Prof. Ron Weiss (Dept. Biological Engineering, MIT), Prof. Alan Grozinsky (Dept. Mechanical Engineering, MIT), Prof. Roger Kamm (Dept. Mechanical Engineering, MIT), and the MIT Biotechnology Process Engineering Center gave me permission to utilize equipment in their labs. Sue Craig (Martin Humphries Lab, University of Manchester, UK) kindly provided the cDNA of I Domains of ITGA1 and ITGA2. Dr. Amit Roy transformated these I Domain plasmids into DH5α e-coli, performed the first batch purifications of GST-I Domain fusions, and taught me several experimental protocols. Members of the Newman Lab (Dr. Itzel Ramos-Solis, Prof. Yun Wang, Prof. L. Dietrich, and Dr. N. Sullivan), the Dedon Lab (Dr. Bahar Edrisi, Dr. Vassilios Dendroulakis, Dr. Mike DeMott), and the Hynes Lab (Dr. Kaan Certel, Jessica Louie, Dr.Tom Seegar) turned me from an ordinary mechanical engineer to someone who can design and conduct a large variety of biological experimental techniques, and assisted me in protocol troubleshooting. Debby Pheasant (Biophysical Instrumentation Facility, Department of Biology, MIT) helped conducting the circular dichroism experiments. Dr Albert Tai (Tufts University core facility genomics core) assisted in performing and interpretting the BIACORE experiments. Nicki Watson (Keck imaging facility, Whitehead Institute, MIT) conducted the TEM imaging of collagen scaffolds. Donald Galler (Dept. of

Materials Science and Engineering) assisted in SEM imaging. My former UROP students Amanda Mok and Lily Xu contributed in developing and troubleshooting the protein expression protocols.

5.8 Literature Cited

Biacore Sensor Surface Handbook, BIACORE, General Electric Healthcare, 2005.

Biacore Concentration Analysis Handbook, BIACORE, General Electric Healthcare, 2008.

GST Gene Fusion System Handbook, General Electric healthcare, 2009.

Optimization Toolbox™ User's Guide, version 2012b, The Mathworks Inc, Natick MA, 2012*GST Gene Fusion System Handbook*, Amersham Biosciences.

Sample Preparation for CD Measurements, Technical Note, Aviv Biomedical.

Bhairi SM, Detergents, *A guide to the properties and uses of detergents in biological systems*, Application Note, Calbiochem, 2001.

Guiducci C. *Surface plasmon resonance based systems*, Advanced Bioengineering methods laboratory, EPFL.

Adams SR, Campbell RE, Gross LA, Martin BR, Walkup GK, Yao Y, Liopis J, and Tsien RY. New Biarsenical Ligands and Tetracysteine Motifs for Protein Labeling In Vitro and In Vivo: Synthesis and Biological Applications, *J. Am. Chem. Society* **124**(21): 6063-76, 2002.

Adams SR, Tsien RY. Preparation of the membrane-permeant biarsenicals FlAsH-EDT2 and ReAsH-EDT2 for fluorescent labeling of tetracysteine-tagged proteins. *Nat Protoc.* **3**(9): 1527-34, 2008.

Allen M, and Jones JL. Jekyll and Hyde: the role of the microenvironment on the progression of cancer, *J. Pathol.* **223**(2): 162-76, 2011.

Amatangelo MD, Bassi DE, Klein-Szanto AJ, and Cukierman E. Stroma-derived three-dimensional matrices are necessary and sufficient to promote desmoplastic differentiation of normal fibroblasts. *Am. J. Pathol.* **167**: 475-88, 2005.

Badylak SF. The extracellular matrix as a biologic scaffold material. *Biomaterials* **28**(25): 3587-93, 2007.

Badylak SF, Taylor D, and Uygun K. Whole-organ tissue engineering: decellularization and recellularization of three-dimensional matrix scaffolds. *Annu Rev Biomed Eng.* **13**: 27-53, 2011. (doi: 10.1146/annurev-bioeng-071910-124743).

Barber TA, Harbers GM, Park S, Gilbert M, and Healy KE. Ligand density characterization of peptide-modified biomaterials. *Biomaterials* **26**(34): 6897-905, 2005.

Betzig E, Patterson GH, Sougrat R, Lindwasser OW, Olenych S, Bonifacino JS, Davidson MW, Lippincott-Schwartz J, and Hess HF. Imaging intracellular fluorescent proteins at nanometer resolution. *Science.* **313**(5793): 1642-5, 2006.

Bosman FT, and Stamenkovic I. Functional structure and composition of the extracellular matrix. *J Pathol.* **200**(4): 423-8, 2003.

Brown BN, Barnes CA, Kasick RT, Michel R, Gilbert TW, Beer-Stolz D, Castner DG, Ratner BD, and Badylak SF. Surface characterization of extracellular matrix scaffolds. *Biomaterials* **31**(3): 428-37, 2010. (doi: 10.1016/j.biomaterials.2009.09.061).

Calderwood DA, Tuckwell DS, Eble J, Kühn K, Humphries MJ. The integrin alpha1 A-domain is a ligand binding site for collagens and laminin. *J. Biol. Chem.* **272**(19): 12311-7, 1997.

Canning MT, Postovit LM, Clarke SH, and Graham CH. Oxygen-mediated regulation of gelatinase and tissue inhibitor of metalloproteinases-1 expression by invasive cells. *Exp Cell Res.* **267**(1):88-94, 2001.

Canty EG, and Kadler KE. Procollagen trafficking, processing and fibrillogenesis. *J Cell Sci.* **118**(7): 1341-53, 2005.

Carmeliet P, Dor Y, Herbert JM, Fukumura D, Brusselmans K, Dewerchin M, Neeman M, Bono F, Abramovitch R, Maxwell P, Koch CJ, Ratcliffe P, Moons L, Jain RK, Collen D, and Keshert E. Role of HIF-1alpha in hypoxia-mediated apoptosis, cell proliferation and tumour angiogenesis. *Nature* **394**(6692): 485-90, 1998.

Carracedo S, Lu N, Popova SN, Jonsson R, Eckes B, Gullberg D. The fibroblast integrin alpha11beta1 is induced in a mechanosensitive manner involving activin A and regulates myofibroblast differentiation. *J Biol Chem.* **285**(14): 10434-45, 2010.

Carrangher NO, Frame MC. Focal adhesion and actin dynamics: a place where kinases and proteases meet to promote invasion. *Tr. Cell Biol.* **14**(5): 241-9, 2004.

Chamberlain LJ, Yannas IV, Hsu HP, Strichartz G, and Spector M. Collagen-GAG substrate enhances the quality of nerve regeneration through collagen tubes up to level of autograft. *Exp Neurol.* **154**(2): 315-29, 1998. (doi:10.1006/exnr.1998.6955).

Chen B, Cao H, Yan P, Mayer MU, Squier TC. Identification of an orthogonal peptide binding motif for biarsenical multiuse affinity probes. *Bioconjug. Chem.* **18**(4): 1259-65, 2007.

Clarke DT. Circular dichroism and its use in protein-folding studies. *Methods Mol. Biol.* **752**: 59-72, 2011.

Connors WL, Jokinen J, White DJ, Puranen JS, Kankaanpää P, Upla P, Tulla M, Johnson MS, Heino J. Two synergistic activation mechanisms of alpha2beta1 integrin-mediated collagen binding. *J Biol Chem.* **282**(19): 14675-83, 2007.

Cooper MA. Optical biosensors in drug discovery, *Nat Rev Drug Discov.* **1**(7): 515-28, 2002.

Crivat G, and Taraska JW. Imaging proteins inside cells with fluorescent tags. *Trends Biotechnol.* **30**(1): 8-16, 2012.

Dagalakis N, Flink J, Stasikelis P, Burke JF, and Yannas IV. Design of an artificial skin. Part III. Control of pore structure. *J Biomed Mater Res.* **14**(4): 511-28, 1980.

Denk W, Strickler JH, and Webb WW. Two-photon laser scanning fluorescence microscopy. *Science* **248**(4951): 73-6, 1990. (doi:10.1126/science.2321027).

Derby B. Printing and prototyping of tissues and scaffolds. *Science* **338**(6109): 921-6, 2012.

Di Lullo GA, Sweeney SM, Korkko J, Ala-Kokko L, San Antonio JD. Mapping the ligand-binding sites and disease-associated mutations on the most abundant protein in the human, type I collagen. *J Biol Chem.* **277**(6): 4223-31, 2002.

Dickeson SK, Mathis NL, Rahman M, Bergelson JM, Santoro SA. Determinants of ligand

binding specificity of the alpha1beta1 and alpha2beta1 integrins. *J Biol Chem.* **274**(45): 32182-91, 1999.

DiPersio CM, Shah S, Hynes RO. a3Aβ1 integrin localizes to focal contacts in response to diverse extracellular matrix proteins. *J Cell Sci.* **108**(6): 2321-36, 1995.

Emsley J, King SL, Bergelson JM, Liddington RC. Crystal Structure of the I domain from integrin α2β1, *J. Biol .Chem.* **272**(45): 28512-7, 1997.

Emsley J, Knight CG, Farndale RW, Barnes MJ, Liddington RC. Structural Basis of Collagen Recognition by Integrin α2β1, *Cell* **101**(1): 47-56, 2000.

Engler A, Bacakova L, Newman C, Hategan A, Griffin M, and Discher D. Substrate compliance versus ligand density in cell on gel responses. *Biophys J.* **86**(1): 617-28, 2004.

Estavillo D, Ritchie A, Diacovo TG, Cruz MA. Functional analysis of a recombinant glycoprotein Ia/IIa (Integrin α2β1) I domain that inhibits platelet adhesion to collagen and endothelial matrix under flow conditions. *J Biol Chem.* **274**(50): 35921-6, 1999.

Flynn LA, Blissett AR, Calomeni EP, and Agarwal G. Inhibition of collagen fibrillogenesis by cells expressing soluble extracellular domains of DDR1 and DDR2. *J Mol Biol.* **395**(3): 533-43, 2010. (doi: 10.1016/j.jmb.2009.10.073).

Fukata Y, Amano M, Kaibuchi K. Rho-Rho-kinase pathway in smooth muscle muscle contraction and cytoskeletal reorganization of non-muscle cells, *Tr. Pharm. Sci.* **22**(1): 32-9, 2001.

Gallagher WM, Lynch I, Allen LT, Miller I, Penney SC, O'Connor DP, Pennington S, Keenan AK, and Dawson KA. Molecular basis of cell-biomaterial interaction: insights gained from transcriptomic and proteomic studies. *Biomaterials.* **27**(35): 5871-82, 2006.

Gelse K, Poeschl E, Aigner T. Collagens - structure, function, and biosynthesis, *Adv. Drug Del. Rev.* **55**(12): 1531-46, 2003.

Giepmans BN, Adams SR, Ellisman MH, Tsien RY. The fluorescent toolbox for assessing protein location and function. *Science* **312**(5771): 217-24, 2006.

Gomperts BD, Kramer IM, and Tatham PER. *Signal Transduction*, Elsevier Academic Press, 2002.

Goodrich JA, and Kugel JF. *Binding and kinetics for molecular biologists*, Cold Spring Harbor Laboratory Press, 2007.

Greenfield NJ. Using circular dichroism spectra to estimate protein secondary structure. *Nat Protoc.* **1**(6): 2876-90, 2006.

Grenache DG, Zhang Z, Wells LE, Santoro SA, Davidson JM, Zutter MM. Wound Healing in the α2β1 Integrin-Deficient Mouse: Altered Keratinocyte Biology and Dysregulated Matrix Metalloproteinase Expression. *J. Inv. Dermatol.* **127**(2): 455-66, 2007.

Griffin BA, Adams SR, and Tsien RY. Specific covalent labeling of recombinant protein molecules inside live cells. *Science* **281**(5374): 269-72, 1998.

Griffin BA, Adams SR, Jones J, and Tsien RY. Fluorescent labeling of recombinant proteins in living cells with FlAsH. *Methods Enzymol.* **327**: 565-78, 2000.

Griffith LG, and Swartz MA. Capturing complex 3D tissue physiology in vitro. *Nat Rev Mol Cell Biol.* **7**(3): 211-24, 2006.

Gross J, and Schmitt FO. The structure of human skin collagen as studied with the electron microscope. J Exp Med. 1948 Nov;88(5):555-68.

Grzesiak JJ, and Bouvet M. Determination of the Ligand-Binding Specificities of the α2β1 and α1β1 Integrins in a Novel 3-Dimensional In Vitro Model of Pancreatic Cancer, *Pancreas* **34**(2): 220-8, 2007. (doi:10.1097/01.mpa.0000250129.64650.f6).

Hamaia SW, Pugh N, Raynal N, Némoz B, Stone R, Gullberg D, Bihan D, Farndale RW. Mapping of potent and specific binding motifs, GLOGEN and GVOGEA, for integrin α1β1 using collagen toolkits II and III. *J Biol Chem.* **287**(31): 26019-28, 2012.

Harbers GM, Gamble LJ, Irwin EF, Castner DG, and Healy KE. Development and characterization of a high-throughput system for assessing cell-surface receptor-ligand engagement. *Langmuir* **21**(18): 8374-84, 2005.

Harley BA, Spilker MH, Wu JW, Asano K, Hsu HP, Spector M, and Yannas IV. Optimal degradation rate for collagen chambers used for regeneration of peripheral nerves over long gaps. *Cells Tissues Organs.* **176**(1-3): 153-65, 2004.

Harley BA, Kim HD, Zaman MH, Yannas IV, Lauffenburger DA, and Gibson LJ. Microarchitecture of three-dimensional scaffolds influences cell migration behavior via junction interactions. *Biophys J.* **95**(8): 4013-24, 2008. (doi: 10.1529/biophysj.107.122598).

Harper S, and Speicher DW, Expression and Purification of GST Fusion Proteins, *Current Protocols in Protein Science,* Unit 6.6, John Willey and Sons, 2008.

Heino J. The collagen receptor integrins have distinct ligand recognition and signaling functions, *Matr. Biol.* **19**(4): 319-23, 2000.

Heino J, Huhtala M, Käpylä J, Johnson MS. Evolution of collagen-based adhesion systems, *Int. J. Biochem. Cell Bio.* **41**(2): 341-8, 2009.

Hell SW, and Wichmann J. Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy. *Optics Letters* **19** (11): 780-782, 1994.

Hermanson GT. *Bioconjugate Techniques*, Second Edition, Academic Press, 2008.

Hoffmann C, Gaietta G, Bünemann M, Adams SR, Oberdorff-Maass S, Behr B, Vilardaga JP, Tsien RY, Ellisman MH, and Lohse MJ. A FlAsH-based FRET approach to determine G protein-coupled receptor activation in living cells. *Nat Methods* **2**(3): 171-6, 2005.

Hoffmann C, Gaietta G, Zürn A, Adams SR, Terrillon S, Ellisman MH, Tsien RY, and Lohse MJ. Fluorescent labeling of tetracysteine-tagged proteins in intact cells. *Nat Protoc.* **5**(10): 1666-77, 2010.

Holub A, Byrnes J, Anderson S, Dzaidzio L, Hogg N, and Huttenlocher A. Ligand density modulates eosinophil signaling and migration. *J. Leukocyte Biol.* **73**(5): 657-64, 2003. (doi:10.1189/jlb.0502264).

Hsiong SX, Huebsch N, Fischbach C, Kong HJ, and Mooney DJ. Integrin-adhesion ligand bond formation of preosteoblasts and stem cells in three-dimensional RGD presenting matrices. *Biomacromolecules* **9**(7): 1843-51, 2008. (doi: 10.1021/bm8000606).

Hubbell JA. Biomaterials science and high-throughput screening. Nat Biotechnol.**22**(7): 828-9,

2004.

Huebsch ND, and Mooney DJ. Fluorescent resonance energy transfer: A tool for probing molecular cell-biomaterial interactions in three dimensions. *Biomaterials* **28**(15): 2424-37, 2007.

Huebsch N, Arany PR, Mao AS, Shvartsman D, Ali OA, Bencherif SA, Rivera-Feliciano J, and Mooney DJ. Harnessing traction-mediated manipulation of the cell/matrix interface to control stem-cell fate. *Nat Mater.* **9**(6): 518-26, 2010. (doi: 10.1038/nmat2732).

Huijbers IJ, Iravani M, Popov S, Robertson D, Al-Sarraj S, Jones C, and Isacke CM. A role for fibrillar collagen deposition and the collagen internalization receptor endo180 in glioma invasion. *PLoS One.* **5**(3): e9808, 2010. (doi: 10.1371/journal.pone.0009808).

Hulmes DJ, and Miller A. Quasi-hexagonal molecular packing in collagen fibrils. *Nature.* **282**(5741): 878-80, 1979.

Hynes RO. Integrins: Bidirectional, Allosteric Signaling Machines. *Cell* **110**(6): 673-7, 2002.

Hynes RO. The extracellular matrix: not just pretty fibrils. *Science* **326**(5957): 1216-9, 2009. (doi: 10.1126/science.1176009).

Jason-Moller L, Murphy M, and Bruno J. Overview of Biacore Systems and Their Applications, *Current Protocols in Protein Science*, Unit 19.13, Willey and Sons, 2006.

Jin M, Andricioaei I, Springer TA. Conversion between Three Conformational States of Integrin I Domains with a C-Terminal Pull Spring Studied with Molecular Dynamics, *Structure* **12**(12): 2137-47, 2004.

Jirousková M, Jaiswal JK, and Coller BS. Ligand density dramatically affects integrin alpha IIb beta 3-mediated platelet signaling and spreading. *Blood.* **109**(12): 5260-9, 2007.

Johnson WC Jr. Protein secondary structure and circular dichroism: a practical guide. *Proteins* **7**(3): 205-14, 1990.

Jokinen J, Dadu E, Nykvist P, Käpylä J, White DJ, Ivaska J, Vehviläinen P, Reunanen H, Larjava H, Häkkinen L, Heino J. Integrin-mediated cell adhesion to type I collagen fibrils. *J Biol Chem.* **279**(30): 31956-63, 2004.

Kagan HM, and Li W. Lysyl oxidase: properties, specificity, and biological roles inside and outside of the cell. *J Cell Biochem.* **88**(4): 660-72, 2003.

Kamata T, Takada Y. Direct binding of collagen to the I domain of integrin α2β1 (VLA-2, CD49b/CD29) in a divalent cation-independent manner. *J Biol Chem.* **269**(42): 26006-10, 1994.

Kamata T, Liddington RC, and Takada Y. Interaction between Collagen and the α2 I-domain of Integrin α2β1. Critical role of conserved residues in the metal ion-dependent adhesion site (MIDAS) region. , *J. Biol. Chem.* **274**(45): 32108-11, 1999.

Karlsson R, and Fält A. Experimental design for kinetic analysis of protein-protein interactions with surface plasmon resonance biosensors. *J Immunol Methods.* **200**(1-2): 121-33, 1997.

Kauppila S, Stenbäck F, Risteli J, Jukkola A, and Risteli L. Aberrant type I and type III collagen gene expression in human breast cancer in vivo. *J Pathol.* **186**(3): 262-8, 1998.

Kelly SM, Jess TJ, Price NC. How to study proteins by circular dichroism. *Biochim Biophys Acta* **1751**(2): 119-39, 2005.

Keselowsky BG, Collard DM, and García AJ. Integrin binding specificity regulates biomaterial surface chemistry effects on cell differentiation. *Proc Natl Acad Sci U S A.* **102**(17): 5953-7, 2005.

Kim JK, Xu Y, Xu X, Keene DR, Gurusiddappa S, Liang X, Wary KK, Höök M. A Novel Binding Site in Collagen III for Integrins α1β1 and α2β1. *J. Biol. Chem.* **280**(37): 32512-20, 2005.

Kingshott P, Andersson G, McArthur SL, and Griesser HJ. Surface modification and chemical surface analysis of biomaterials. *Curr Opin Chem Biol.* **15**(5): 667-76, 2011. (doi: 10.1016/j.cbpa.2011.07.012).

Knight CG, Morton LF, Onley DJ, Peachey AR, Messent AJ, et al. 1998. Identification in collagen type I of an integrin α2β1-binding site containing an essential GER sequence. *J. Biol. Chem.* **273**(50): 33287–94, 1998.

Knight CG, Morton LF, Peachey AR, Tuckwell DS, Farndale RW, Barnes MJ. The collagen-binding A-domains of integrins α1β1 and α2β1 recognize the same specific amino acid sequence, GFOGER, in native (triple-helical) collagens, *J. Biol Chem.* **275**(1): 35-40, 2000.

Kong HJ, Boontheekul T, and Mooney DJ. Quantifying the relation between adhesion ligand-receptor bond formation and cell phenotype. *Proc Natl Acad Sci U S A* **103**(49): 18534-9, 2006.

Krahn KN, Bouten CV, van Tuijl S, van Zandvoort MA, Merkx M. Fluorescently labeled collagen binding proteins allow specific visualization of collagen in tissues and live cell culture. *Anal Biochem.* **350**(2): 177-85, 2006.

Kurose K, Hoshaw-Woodard S, Adeyinka A, Lemeshow S, Watson PH, and Eng C. Genetic model of multi-step breast carcinogenesis involving the epithelium and stroma: clues to tumour-microenvironment interactions. *Hum Mol Genet.* **10**(18): 1907-13, 2001.

Lahti M, Bligt E, Niskanen H, Parkash V, Brandt AM, Jokinen J, Patrikainen P, Käpylä J, Heino J, Salminen TA. Structure of collagen receptor integrin α1I domain carrying the activating mutation E317A. *J Biol Chem.* **286**(50): 43343-51, 2011.

Lakowicz JR. *Principles of Fluorescence Spectroscopy*, Third Edition, Springer, 2006.

Lee JO, Bankston LA, Arnaout MA, Liddington RC. Two conformations of the integrin A-domain (I-domain): a pathway for activation?, *Structure* **3**(12): 1333-40, 1995.

Leitinger B. Transmembrane collagen receptors. *Annu Rev Cell Dev Biol.* **27**: 265-90, 2011.

Leitinger B, and Hohenester E. Mammalian collagen receptors, *Matr. Biol.* **26**: 146-155, 2007.

Li R, Mitra N, Gratkowski H, Vilaire G, Litvinov R, Nagasami C, Weisel JW, Lear JD, DeGrado WF, Bennett JS. Activation of Integrin alIbb3 by modulation of transmembrane helix associations, *Science* **300**(5620): 795-8, 2003.

Lieberg B, Nylander C, and Lundström I. Surface plasmon resonance for gas detection and biosensing. *Sensors Actuators* **4**: 299-304, 1983.

Lippincott-Schwartz J. Emerging in vivo analyses of cell function using fluorescence imaging. *Annu Rev Biochem.* **80**: 327-32, 2011.

Lodish H, Berk A, Kaiser CA, Krieger M, Scott MP, Bretscher A, Ploegh H, and Paul Matsudaira. *Molecular Cell Biology*, Sixth Edition, Freeman, 2007.

Lu C, Shimaoka M, Ferzly M, Oxvig C, Takagi J, Springer TA. An isolated, surface-expressed I domain of the integrin aLb2 is sufficient for strong adhesive function when locked in the open conformation with a disulfide bond, *PNAS* **98**(5): 2387-92, 2001.

Lu P, Weaver VM, and Werb Z. The extracellular matrix: A dynamic niche in cancer progression. *J. Cell Biol.* **196**(4): 395-406, 2012.

Luedtke NW, Dexter RJ, Fried DB, and Schepartz A. Surveying polypeptide and protein domain conformation and association with FlAsH and ReAsH. *Nat Chem Biol.* **3**(12): 779-84, 2007.

Luo BH, Carman CV, Springer TA. Structural basis of integrin regulation and signaling. *Annu Rev Immunol.* **25**: 619-47, 2007.

Ma Z, Mao Z, and Gao C. Surface modification and property analysis of biomedical polymers used for tissue engineering. *Colloids Surf B Biointerfaces.* **60**(2): 137-57, 2007.

Machleidt T, Robers M, and Hanson GT. Protein labeling with FlAsH and ReAsH. *Methods Mol Biol.* **356**: 209-20, 2007.

Madani F, Lind J, Damberg P, Adams SR, Tsien RY, and Gräslund AO. Hairpin structure of a biarsenical-tetracysteine motif determined by NMR spectroscopy. *J Am Chem Soc.* **131**(13): 4613-5, 2009.

Maheshwari G, Brown G, Lauffenburger DA, Wells A, and Griffith LG. Cell adhesion and motility depend on nanoscale RGD clustering. *J Cell Sci.* **113** (10): 1677-86, 2000.

Mao Y, and Schwarzbauer JE. Fibronectin fibrillogenesis, a cell-mediated matrix assembly process. *Matrix Biol.* **24**(6): 389-99, 2005.

Marciano DK, Denda S, and Reichardt LF. Methods for identifying novel integrin ligands. *Methods Enzymol.* **426**: 223-37, 2007.

Martin BR, Giepmans BN, Adams SR, and Tsien RY. Mammalian cell-based optimization of the biarsenical-binding tetracysteine motif for improved fluorescence and affinity. *Nat Biotechnol.* **23**(10): 1308-14, 2005.

Maskarinec SA, and Tirrell DA. Protein engineering approaches to biomaterials design. *Curr Opin Biotechnol.* **16**(4): 422-6, 2005.

Massague J. How Cells Read TGFβ Signals, *Nat. Rev. Molec. Cell. Bio* **1**(3): 169-78, 2000.

Messent AJ, Tuckwell DS, Knäuper V, Humphries MJ, Murphy G, Gavrilovic J. Effects of collagenase-cleavage of type I collagen on α2β1 integrin-mediated cell adhesion. *J Cell Sci.* **111**(8): 1127-35, 1998.

Miller JS, Stevens KR, Yang MT, Baker BM, Nguyen DH, Cohen DM, Toro E, Chen AA, Galie PA, Yu X, Chaturvedi R, Bhatia SN, and Chen CS. Rapid casting of patterned vascular networks for perfusable engineered three-dimensional tissues. *Nat. Mater* **11**(9): 768-74, 2012.

Monsigny M, Mayer R, and Roche AC. Sugar-lectin interactions: sugar clusters, lectin multivalency and avidity. *Carbohydr Lett.* **4**(1): 35-52, 2000.

Murphy M, Jason-Moller L, and Bruno J. Using Biacore to Measure the Binding Kinetics of an Antibody-Antigen Interaction, *Current Protocols in Protein Science*, Unit 19.14, Willey and Sons, 2006.

Myszka DG. Kinetic analysis of macromolecular interactions using surface plasmon resonance

biosensors. *Curr Opin Biotechnol.* **8**(1): 50-7, 1997.

Myszka DG, He X, Dembo M, Morton TA, and Goldstein B. Extending the range of rate constants available from BIACORE: interpreting mass transport-influenced binding data. *Biophys J.* **75**(2): 583-94, 1998.

Naba A, Clauser KR, Hoersch S, Liu H, Carr SA, and Hynes RO. The matrisome: in silico definition and in vivo characterization by proteomics of normal and tumor extracellular matrices. *Mol Cell Proteomics* **11**(4): M111.014647, 2012.

Nelson CM, and Tien J. Microstructured extracellular matrices in tissue engineering and development. *Curr Opin Biotechnol.* **17**(5): 518-23, 2006.

Nocedal J, and Wright S. *Numerical Optimization*, Springer, 2006.

Nolte M, Pepinsky RB, Venyaminov SYu, Koteliansky V, Gotwals PJ, Karpusas M. Crystal structure of the α1β1 integrin I-domain: insights into integrin I-domain function. *FEBS Letters* **452**(3): 379-85, 1999.

Nykvist P, Tu H, Ivaska J, Käpylä J, Pihlajaniemi T, Heino J. Distinct recognition of collagen subtypes by α1β1 and α2β1 integrins. α1β1 mediates cell adhesion to type XIII collagen. *J Biol Chem.* **275**(11): 8255-61, 2000.

Nymalm Y, Puranen JS, Nyholm TK, Käpylä J, Kidron H, Pentikäinen OT, Airenne TT, Heino J, Slotte JP, Johnson MS, Salminen TA. Jararhagin-derived RKKH Peptides Induce Structural Changes in alpha1I Domain of Human Integrin α1β1, *J. Biol. Chem.* **279**(9): 7962-70, 2004.

O'Brien FJ, Harley B A, Yannas IV, and Gibson LJ. Influence of freezing rate on pore structure in freeze-dried collagen-GAG scaffolds, *Biomaterials* **25**6): 1077-86, 2004. (doi: 10.1016/S0142-9612(03)00630-6)

O'Brien FJ, Harley BA, Yannas IV, and Gibson LJ. The effect of pore size on cell adhesion in collagen-GAG scaffolds, *Biomaterials* **26**(4): 433-41, 2005. (DOI: 10.1016/ j.biomaterials.2004.02.052.)

Okuyama K, Bächinger HP, Mizuno K, Boudko S, Engel J, Berisio R, and Vitagliano L. Re: Microfibrillar structure of type I collagen in situ. *Acta Crystallogr D Biol Crystallogr.* **65**(Pt 9):1007-8; author reply 1009-10, 2009. (doi: 10.1107/S0907444909023051).

Olumi AF, Grossfeld GD, Hayward SW, Carroll PR, Tlsty TD, Cunha GR. Carcinoma-associated fibroblasts direct tumor progression of initiated human prostatic epithelium. *Cancer Res.* **59**(19): 5002-11, 1999.

Onley DJ, Knight CG, Tuckwell DS, Barnes MJ, Farndale RW. Micromolar Ca2+ concentrations are essential for Mg2+-dependent binding of collagen by the integrin α2bβ1 in human platelets. *J Biol Chem.* **275**(32): 24560-4, 2000.

Orgel JP, Irving TC, Miller A, and Wess TJ. Microfibrillar structure of type I collagen in situ. *Proc Natl Acad Sci U S A.* **103**(24):9001-5, 2006.

Orlando G, Wood KJ, Stratta RJ, Yoo JJ, Atala A, and Soker S. Regenerative medicine and organ transplantation: past, present, and future. *Transplantation.* **91**(12): 1310-7, 2011. (doi: 10.1097/TP.0b013e318219ebb5).

Pace CN, Vajdos F, Fee L, Grimsley G, and Gray T. How to measure and predict the molar

absorption coefficient of a protein, *Prot. Sci.* **4**(11): 2411-23, 1995.

Pain R. Determining the CD spectrum of a Protein, *Current Protocols in Protein Science*, Unit 7.6, John Willey and Sons, 2004.

Park H, Hanson GT, Duff SR, Selvin PR. Nanometre localization of single ReAsH molecules. *J Microsc.* **216**(3): 199-205, 2004.

Paszek MJ, Zahir N, Johnson KR, Lakins JN, Rozenberg GI, Gefen A, Reinhart-King CA, Margulies SS, Dembo M, Boettiger D, Hammer DA, and Weaver VM. Tensional homeostasis and the malignant phenotype. *Cancer Cell* **8**(3): 241-54, 2005. (doi:10.1016/j.ccr.2005.08.010)

Pawley JB. (editor) *Handbook of biological confocal microscopy*, third edition. Berlin, Springer, 2006.

Perret S, Eble JA, Siljander PR, Merle C, Farndale RW, Theisen M, Ruggiero F. Prolyl hydroxylation of collagen type I is required for efficient binding to integrin alpha 1 beta 1 and platelet glycoprotein VI but not to alpha 2 beta 1. *J Biol Chem.* **278**(32): 29873-9, 2003.

Pilcher BK, Dumin JA, Sudbeck BD, Krane SM, Welgus HG, Parks WC. The activity of collagenase-1 is required for keratinocyte migration on a type I collagen matrix. *J Cell Biol.* **137**(6): 1445-57, 1997.

Plow EF, Haas TA, Zhang L, Loftus J, and Smith JW. Ligand binding to integrins. *J Biol Chem.* **275**(29): 21785-8, 2000.

Pomorski A, Krężel A. Exploration of biarsenical chemistry--challenges in protein research. *Chembiochem.* **12**(8): 1152-67, 2011.

Provenzano PP, Eliceiri KW, Campbell JM, Inman DR, White JG, and Keely PJ. Collagen reorganization at the tumor-stromal interface facilitates local invasion. *BMC Med.* **4**(1): 38, 2006. (doi:10.1186/1741-7015-4-38).

Racine-Samson L, Rockey DC, Bissell DM. The role of α1β1 Integrin in Wound Contraction. A quantitative analysis of liver myofibroblasts in vivo and in primary culture. *J. Biol. Chem.* **272**(49): 30911-7, 1997.

Radivojac P. *Circular Dichroism*, Structural Bioinformatics Lecture Slides, Indiana University Bloomington, (http://www.informatics.indiana.edu/predrag/classes/2008springi619/week4_m.pdf), 2008.

Ratner BD, and Bryant SJ. Biomaterials: where we have been and where we are going. *Annu Rev Biomed Eng.* **6**:41-75, 2004.

Raymond K, Deugnier MA, Faraldo MM, and Glukhova MA. Adhesion within the stem cell niches. *Curr Opin Cell Biol.* **21**(5): 623-9, 2009. (doi: 10.1016/j.ceb.2009.05.004).

Raynal N, Hamaia SW, Siljander PR, Maddox B, Peachey AR, Fernandez R, Foley LJ, Slatter DA, Jarvis GE, Farndale RW. Use of synthetic peptides to locate novel integrin α2β1-binding motifs in human collagen III. *J Biol Chem.* **281**(7): 3821-31, 2006.

Rehm H. *Protein biochemistry and proteomics*, Academic press, 2006.

Rial DV, and Ceccarelli EA. Removal of DnaK contamination during fusion protein purifications. *Protein Expr Purif.* **25**(3): 503-7, 2002.

Ricard-Blum S. The collagen family. *Cold Spring Harb Perspect Biol.* **3**(1): a004978, 2011. (doi: 10.1101/cshperspect.a004978).

Rich RL, Deivanayagam CC, Owens RT, Carson M, Höök A, Moore D, Symersky J, Yang VW, Narayana SV, and Höök M. Trench-shaped binding sites promote multiple classes of interactions between collagen and the adherence receptors, α1β1 integrin and Staphylococcus aureus cna MSCRAMM, *J. Biol. Chem.* **274**(35): 24906-13, 1999.

Rich RL, Kreikemeyer B, Owens RT, LaBrenz S, Narayana SV, Weinstock GM, Murray BE, and Höök M. Ace is a collagen-binding MSCRAMM from *Enterococcus faecalis*. *J Biol Chem.* **274**(38): 26939-45, 1999b.

Riikonen T, Westermarck J, Koivisto L, Broberg A, Kähäri VM, Heino J. Integrin α2β1 Is a Positive Regulator of Collagenase (MMP-1) and Collagen α1(I) Gene Expression. *J. Biol. Chem.* **270**(22): 13548-52, 1995.

Ruoslahti E. The Walter Herbert Lecture. Control of cell motility and tumour invasion by extracellular matrix interactions. *Br J Cancer.* **66**(2): 239-42, 1992.

Rust MJ, Bates M, and Zhuang X. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nat Methods* **3**(10): 793-5, 2006.

Sands RW, and Mooney DJ. Polymers to direct cell fate by controlling the microenvironment. *Curr Opin Biotechnol.* **18**(5): 448-53, 2007.

Schiro JA, Chan BM, Roswit WT, Kassner PD, Pentland AP, Hemler ME, Eisen AZ, Kupper TS. Integrin α2β1 (VLA-2) Mediates Reorganization and Contraction of Collagen Matrices by Human Cells, *Cell* **67**(2): 403-10, 1991.

Schuck P. Reliable determination of binding affinity and kinetics using surface plasmon resonance biosensors. *Curr Opin Biotechnol.* **8**(4): 498-502, 1997.

Schuck P., Boyd LF, and Andersen PS. Measuring Protein Interactions by Optical Biosensors, *Current Protocols in Protein Science*, Unit 20.2, Willey and Sons, 1999.

Serini G, Bochaton-Piallat ML, Ropraz P, Geinoz A, Borsi L, Zardi L, and Gabbiani G. The fibronectin domain ED-A is crucial for myofibroblastic phenotype induction by transforming growth factor-beta1. *J Cell Biol.* **142**(3): 873-81, 1998.

Sharon N., and Lis H. *Lectins*. Second edition. Springer, 2007.

Shen Q, Wang Y, Kokovay E, Lin G, Chuang SM, Goderie SK, Roysam B, and Temple S. Adult SVZ stem cells lie in a vascular niche: a quantitative analysis of niche cell-cell interactions. *Cell Stem Cell.* **3**(3): 289-300, 2008. (doi: 10.1016/j.stem.2008.07.026).

Shi M, Pedchenko V, Greer BH, Van Horn WD, Santoro SA, Sanders CR, Hudson BG, Eichman BF, Zent R, Pozzi A. Enhancing integrin α1 inserted (I) domain affinity to ligand potentiates integrin α1β1-mediated down-regulation of collagen synthesis. *J Biol Chem.* **287**(42): 35139-52, 2012.

Shimaoka M, Lu C, Palframan RT, von Andrian UH, McCormack A, Takagi J, Springer TA. Reversibly locking a protein fold in an active conformation with a disulfide bond: Integrin aL I domains with high affinity and antagonist activity in vivo, *PNAS* **98**(11): 6009-14, 2001.

Shimaoka M, Takagi J, Springer TA. Conformational regulation of integrin structure and

function. *Annu Rev Biophys Biomol Struct.* **31**: 485-516, 2002.

Shimaoka M, Xiao T, Liu JH, Yang Y, Dong Y, Jun CD, McCormack A, Zhang R, Joachimiak A, Takagi J, Wang JH, Springer TA. Structures of the aL I domain and its complex with ICAM-1 reveal a shape-shifting pathway for integrin regulation, *Cell* **112**(1): 99-111, 2003.

Siljander PR, Hamaia S, Peachey AR, Slatter DA, Smethurst PA, Ouwehand WH, Knight CG, Farndale RW. Integrin activation state determines selectivity for novel recognition sites in fibrillar collagens. *J Biol Chem.* **279**(46): 47763-72, 2004.

Simonian M.H. Spectrophotometric Determination of Protein Concentration, *Current Protocols in Food Analytical Chemistry*, Unit B1.3, John Willey and Sons, 2002.

Sjölander S, and Urbaniczky C. Integrated fluid handling system for biomolecular interaction analysis. *Anal Chem.* **63**(20): 2338-45, 1991.

Smith C, Estavillo D, Emsley J, Bankston LA, Liddington RC, Cruz MA. Mapping the collagen-binding site in the I domain of the glycoprotein Ia/IIa (integrin alpha(2)beta(1)). *J Biol Chem.* **275**(6): 4205-9, 2000.

Soller EC, Tzeranis DS, Miu K, So PT, and Yannas IV. Common features of optimal collagen scaffolds that disrupt wound contraction and enhance regeneration both in peripheral nerves and in skin. *Biomaterials* **33**(19): 4783-91 2012. (doi:10.1016/j.biomaterials. 2012.03.068).

Spagnuolo CC, Vermeij RJ, and Jares-Erijman EA. Improved photostable FRET-competent biarsenical-tetracysteine probes based on fluorinated fluoresceins. *J Am Chem Soc.* **128**(37): 12040-1, 2006.

Sweeney SM, Orgel JP, Fertala A, McAuliffe JD, Turner KR, Di Lullo GA, Chen S, Antipova O, Perumal S, Ala-Kokko L, Forlino A, Cabral WA, Barnes AM, Marini JC, and San Antonio JD. Candidate cell and matrix interaction domains on the collagen fibril, the predominant protein of vertebrates. *J Biol Chem.* **283**(30): 21187-97, 2008. (doi: 10.1074/jbc.M709319200).

Talior-Volodarsky I, Connelly KA, Arora PD, Gullberg D, and McCulloch CA. α11 integrin stimulates myofibroblast differentiation in diabetic cardiomyopathy. *Cardiovasc Res.* **96**(2): 265-75, 2012. (doi: 10.1093/cvr/cvs259).

Taylor ME, and Drickamer K. Paradigms for glycan-binding receptors in cell adhesion. Curr Opin Cell Biol. **19**(5): 572-7, 2007.

Tiger CF, Fougerousse F, Grundstrom G, Velling T, Gullberg D. α11β1 integrin is a receptor for interstitial collagens involved in cell migration and collagen reorganization on mesenchymal nonmuscle cells. *Dev. Biol.* **237**(1): 116–29, 2001.

Tuckwell D, Calderwood DA, Green LJ, Humphries MJ. Integrin α2 I-domain is a binding site for collagens, *J Cell Sci.* **108**(4): 1629-37, 1995.

Tulla M, Pentikäinen OT, Viitasalo T, Käpylä J, Impola U, Nykvist P, Nissinen L, Johnson MS, Heino J. Selective Binding of Collagen Subtypes by Integrin α1I, α2I, and α10I Domains, *J. Biol. Chem.* **276**(51): 48206-12, 2001.

Tulla M, Lahti M, Puranen JS, Brandt AM, Käpylä J, Domogatskaya A, Salminen TA, Tryggvason K, Johnson MS, Heino J. Effects of conformational activation of integrin α1I

and α2I domains on selective recognition of laminin and collagen subtypes. *Exp. Cell Res.* **314**(8): 1734-43, 2008.

Uygun BE, Soto-Gutierrez A, Yagi H, Izamis ML, Guzzardi MA, Shulman C, Milwid J, Kobayashi N, Tilles A, Berthiaume F, Hertl M, Nahmias Y, Yarmush ML, and Uygun K. Organ reengineering through development of a transplantable recellularized liver graft using decellularized liver matrix. *Nat Med.* **16**(7): 814-20, 2010. (doi: 10.1038/nm.2170).

Vakonakis I, and Campbell ID. Extracellular matrix: from atomic resolution to ultrastructure. *Curr Opin Cell Biol.* **19**(5): 578-83, 2007.

Valenick LV, and Schwarzbauer JE. Ligand density and integrin repertoire regulate cellular response to LPA. *Matrix Biol.* **25**(4): 223-31, 2006. (doi:10.1016/j.matbio.2006.01.002).

Varki A. Selectin ligands. Proc Natl Acad Sci U S A. **91**(16): 7390-7, 1994.

Vogel V, and Sheetz M. Local force and geometry sensing regulate cell functions. *Nat Rev Mol Cell Biol.* **7**(4):265-75, 2006.

Vorup-Jensen T, Waldron TT, Astrof N, Shimaoka M, Springer TA. The connection between metal ion affinity and ligand affinity in integrin I domains. *Biochim Biophys Acta.* **1774**(9): 1148-55, 2007.

Wang T, Yan P, Squier TC, and Mayer MU. Prospecting the proteome: identification of naturally occurring binding motifs for biarsenical probes. *Chembiochem.* **8**(16): 1937-40, 2007.

White ES, Baralle FE, and Muro AF. New insights into form and function of fibronectin splice variants. *J Pathol.* **216**(1): 1-14, 2008. (doi: 10.1002/path.2388).

Wilson K, and Walker J. *Principles and techniques of practical biochemistry*, fifth edition, Cambridge University Press, 2000.

Wombacher R, and Cornish VW. Chemical tags: applications in live cell fluorescence imaging. *J Biophotonics.* **4**(6): 391-402, 2011.

Xu Y, Gurusiddappa S, Rich RL, Owens RT, Keene DR, Mayne R, Höök A, and Höök M. Multiple Binding Sites in Collagen Type I for the Integrins α1β1 and α2β1, *J. Biol. Chem.* **275**(50): 38981-9, 2000.

Xu H, Bihan D, Chang F, Huang PH, Farndale RW, and Leitinger B. Discoidin domain receptors promote α1β1- and α2β1-integrin mediated cell adhesion to collagen by enhancing integrin activation. *PLoS One* **7**(12):e52209, 2012. (doi: 10.1371/journal.pone.0052209).

Yannas IV. Collagen and Gelatin in the Solid State. *J. Macromol. Sci. Revs. Macromol. Chem.* **C7**(1): 49-104, 1972.

Yannas IV. *Tissue and organ regeneration in adults*. Springer, 2001.

Yannas IV, and Burke JF. Design of an artificial skin. I. Basic design principles. *J Biomed Mater Res.* **14**(1): 65-81, 1980.

Yannas IV, Burke JF, Gordon PL, Huang C, and Rubenstein RH. Design of an artificial skin. II. Control of chemical composition. *J Biomed Mater Res.***14**(2): 107-32, 1980.

Yannas IV, Burke JF, Orgill DP, and Skrabut EM. Wound tissue can utilize a polymeric template to synthesize a functional extension of skin. *Science* **215**(4529): 174-6, 1982. (doi:10.1126/science.7031899)

Yannas IV, Lee E, Orgill DP, Skrabut EM, Murphy GF. Synthesis and characterization of a model extracellular matrix that induces partial regeneration of adult mammalian skin. *Proc Natl Acad Sci U S A* **86**(3): 933-7, 1989. (doi:10.1073/pnas.86.3.933)

Yannas IV, Tzeranis DS, Harley BA, and So PT. Biologically active collagen-based scaffolds: advances in processing and characterization. *Philos Trans A Math Phys Eng Sci.* **368**(1917): 2123-39, 2010. (doi: 10.1098/rsta.2010.0015).

Zhang J, Campbell RE, Ting AY, and Tsien RY. Creating new fluorescent probes for cell biology. *Nat Rev Mol Cell Biol.* **3**(12): 906-18, 2002.

Zhang WM, Kapyla J, Puranen JS, Knight CG, Tiger CF, Pentikainen OT, Johnson MS, Farndale RW, Heino J, and Gullberg D. α11β1 Integrin Recognizes the GFOGER Sequence in Interstitial Collagens. *J. Biol. Chem.* **278**(9): 7270-7, 2003.

Zhu GG, Risteli L, Mäkinen M, Risteli J, Kauppila A, and Stenbäck F. Immunohistochemical study of type I collagen and type I pN-collagen in benign and malignant ovarian neoplasms. *Cancer* **75**(4): 1010-7, 1995.

Zürn A, Klenk C, Zabel U, Reiner S, Lohse MJ, and Hoffmann C. Site-specific, orthogonal labeling of proteins in intact cells with two small biarsenical fluorophores. *Bioconjug Chem.* **21**(5): 853-9, 2010.

Chapter 6: Conclusions

6.1 Conclusions

In this thesis we combine the use of advanced imaging, probabilistic image processing, materials science, and biology to conduct three studies that focus on how biomaterials regulate the wound healing outcome in peripheral nerve wound healing. Special emphasis is placed on the regulation of contractile phenotypes by the physicochemical properties of the matrix. This is an exciting research area due to the huge potential impact of quantifying, understanding and exploiting cell-matrix interactions in the development of novel biomaterial-based treatments that can induce regeneration in injured adult organs.

Chapter 2 is not directly involved to a problem related to peripheral nerve regeneration. However, it provides the necessary image processing tools that are used to process the imaging data utilized in the remaining three chapters.

- Provides an image processing pipeline that converts raw spectral microscopy data (not necessary multi-photon) into regions corresponding to particular physical entities of specific chemical composition.
- The algorithm can handle complex low-signal images and can be utilized by users that are not expert in optics or signal processing. It fills a gap in the area of high-content fluorescence imaging, since none of the available image processing toolkits is designed to process low-signal spectral images.
- It presents analytic relationships for the variance and bias of spectral unmixing, the key calculation in processing spectral data, suggests their range of validity, and provides guideline for the experimental design of high content spectral imaging experiments.

Chapter 3

- Describes an imaging-based methodology for studying signal transduction pathways in cells that interact with 3D matrices. It is based on quantifying single cells using imaging assays of their morphology and their binding pattern with the surrounding microenvironment (matrix, cells).
- The methodology is applied to a pilot study of TGFβ isoform signaling via the SMAD pathway in fibroblasts inside porous collagen scaffolds similar to the ones utilized in chapters 4 and 5.
- The results of the pilot study suggest that, apart from the canonical SMAD proteins (SMAD2,3), the non-canonical SMADs (SMAD1,5) affect the response of cells to TGFβ. Specifically, SMAD1 seems to affects the response to TGFβ1 more than TGFβ3; while the TGFBR1 receptor ALK1 and to a lesser extent SMAD5 have the opposite effect. The involvement of the non-canonical SMADs may lead TGFβ1 and TGFβ3 to induce different effects in cells.

Chapter 4

- describes an imaging study of the early (1 and 2 weeks post-injury) wound healing response in two groups of severely injured peripheral nerves treated with either regeneratively "active" scaffolds D or regeneratively "inactive" scaffolds E respectively.
- Imaging results show that the response observed 1 and 2 weeks post-injury follows the trend observed by [Soller et al. 2012] at 9 weeks. In nerves treated with the active scaffold "D" the capsule is thin, contains less αSMA-positive cells and is in contact with the inner lumen of the scaffold. In nerves treated with the inactive scaffold "E" the capsule is thicker and makes fragile contact with the scaffold.

Chapter 5

- Describes a new methodology for quantifying the surface chemistry (surface density of adhesion ligands for particular adhesion receptors) in 3D matrices (biomaterials, ECM).
- The methodology is based on fluorescently-labeled soluble markers that emulate the corresponding adhesion receptor. It provides a quantitative description of the insoluble environment felt by cells.
- The methodology is applied to quantify the ligands of the two major collagen-binding integrins $\alpha_1\beta_1$, $\alpha_2\beta_1$ on the surface of collagen scaffolds "D" and "E" utilized in the animal study of chapter 4, using as markers fluorescently-labeled I Domains of the corresponding integrin.
- Results show that the regeneratively active scaffolds "D" have significantly more ligands compared to the regeneratively inactive scaffolds "E". This difference is attributed to the chemical cross-linking procedure applied to "E" which attacks the carboxyl groups that are necessary components of α1β1, α2β1 ligands.

Even though chapters 3, 4 and 5 may not seem closely related, they all focus on the critical effects of mechanical (contractile) forces occurring during wound healing. Specifically, Chapter 3 focuses on a signaling pathway known to induce contractile phenotypes. Chapter 4 shows that, in nerves grafted with "inactive" scaffold "E" the cells of the capsule tissue that surrounds the nerve express more contractile phenotypes and bind the inner lumen of the scaffold in a more fragile way. In contrast, in nerves grafted with "active" scaffold "D" the expression of contractile phenotypes around the nerve is weaker and there is much more adhesion of cells in the inner lumen of the scaffold. Finally, Chapter 5 shows that the "active" scaffold "D" exposes cells to a much higher density of α1β1, α2β1 ligands for adhesion compared to the inactive scaffold "E". These ligands are required by cells in order to adhere to the matrix and to apply mechanical forces to the scaffold.

6.2 Future Work

The image processing pipeline described in Chapter 2 can be improved in several ways:

- Implement the code in an efficient programming language such as C or Java and utilize GPUs to enhance its computational efficiency. Embed the image processing code in the microscope control software and generate "smart" microscopes that can image large area samples by picking autonomously the regions of interest.

- Convert the pipeline into an ImageJ plug-in and make it freely available to investigators of various spectral microscopy modalities.
- Utilize spectral-FLIM detection for processing extremely hard cases where spectral imaging by itself cannot resolve the components accurately.
- The implementation of spectral fluorescence imaging in deep tissue imaging *in vivo* may require modification of the image processing algorithms, or implementation of optical tricks to handle the deteriorating effects of emission light scattering in the spectral response.

The image informatics methodology described in Chapter 3 can be used to study the role of individual ECM molecules or adhesion receptors in cell-scaffold interactions

- perturbation of the cell-matrix system can be implemented by changing the chemical composition of the matrix, adding or removing ECM components, or cytokines known to bind to ECM components. This approach can be utilized as a screening of ECM molecules for the development of new improved scaffolds of more complex chemical composition
- 3D imaging experiments are relative slow. High throughput imaging modalities can speedup imaging but this is probably not enough. The application of image informatics in large-scale cell signaling studies or screenings requires extensive automation such as sample handling, microfluidics, and image acquisition that would enable consistent staining, and user-free imaging of batches of samples.
- the prediction capacity of the approach can be improved by describing single cells by more imaging assays, and by developing appropriate statistical analysis methods that can interpret complex large-scale results.

The observations of the imaging study described in Chapter 4 can be extended in several ways:

- results suggest that scaffolds D and E regulate differently the elementary processes taking place during the first week (coagulation, fibrin matrix formation, and cell migration into the fibrin matrix). It may be reasonable to conduct a study that focuses on how scaffold properties (particularly surface chemistry) regulate the formation of the capsule during the first week after injury.
- the ability of nonlinear optics to distinguish banded collagen (synthesized by cells) from non-banded collagen (banding lost due to collagen fibril swelling in acid during laboratory processing) could be useful in developing *in vitro* assays for studying collagen fibrilogenesis.
- The study presented in Chapter 4 is *ex vivo*. An intravital multi-photon imaging study of peripheral nerve wound healing in the presence of a scaffold can provide information that is not possible to collect by other means, and answer questions related to the mechanism of capsule formation and how its formation is affected by the presence of biomaterials of various physic-chemical properties. Intrvital imaging combined with molecular targeting of agents that upregulate and downregulate wound contraction could provide solid evidence about a causal relationship between wound contraction and wound healing outcome.

The methodology for quantifying the surface chemistry of 3D matrices described in Chapter 5 can be extended in several ways:

- It can be applied to quantify the density of α1β1 and α2β1 ligands in different biomaterials, providing a standard way to compare them and quantify their effects to cells.
- It can be applied to quantify the density of α1β1 and α2β1 ligands in tissue ECM after decellularization. Experiments could focus on how the progression of particular diseases (e.g. fibrosis, cancer) affect the surface chemistry of the matrix felt by cells
- It can be applied to quantify the density of ligands of other kinds of adhesion receptors. This requires the design, expression and characterization of an appropriate fluorescent marker. In some cases this is straightforward (e.g. α10β1 and α11β1). In other interesting cases (e.g. ligands of α3β1, α5β1, α8β1) this may be harder and require some iteration
- The developed methodology can have a major impact if it is modified and combined with emerging 3D printing tools in order to modify the surface chemistry of a 3D biomaterial at will.

6.3 Literature Cited

Aebischer P, Guénard V, Valentini RF. The morphology of regenerating peripheral nerves is modulated by the surface microgeometry of polymeric guidance channels. Brain Res. 531(1-2):211-8, 1990.

Harley BA, Spilker MH, Wu JW, Asano K, Hsu HP, Spector M, and Yannas IV. Optimal degradation rate for collagen chambers used for regeneration of peripheral nerves over long gaps. Cells Tissues Organs 176(1-3):153-65, 2004.

Ikeguchi R, Kakinoki R, Matsumoto T, Tsuji H, Ishikawa J, Nakamura T. Rat nerve regeneration through a silicone chamber implanted with negative carbon ions. Brain Res Dev Brain Res. **140**(1):127-31, 2003.

Jenq CB, and Coggeshall RE. Nerve regeneration through holey silicone tubes. *Brain Res.* **361**(1-2): 233-41, 1985b.

Jenq CB, Jenq LL, and Coggeshall RE. Nerve regeneration changes with filters of different pore size. *Exp Neurol.* **97**(3): 662-71, 1987.

Soller EC, Tzeranis DS, Miu K, So PT, Yannas IV. Common features of optimal collagen scaffolds that disrupt wound contraction and enhance regeneration both in peripheral nerves and in skin. Biomaterials 33(19): 4783-91, 2012.

Appendix A: Instrument Description

This Appendix provides a brief description of the spectral single-photon counting multi-photon microscope utilized in this thesis.

A1. Optical System

The optical system consists of the optical components that deliver the ultrafast mode-locked infrared excitation beam into the sample, collect the emission and detect it via PMT sensors. The system provides three different PMT sensors.

- A 16-channel MA-PMT (Hamamatsu R59000P-00-L16) used for spectral imaging
- A fast PMT (Hamamatsu R7400P) used for fluorescence lifetime imaging (FLIM) or single-channel intensity imaging.
- A high-efficiency Photon counting head (Hamamatsu H7421-40) used for fluorescence correlation spectroscopy (FCS) or single-channel intensity imaging.

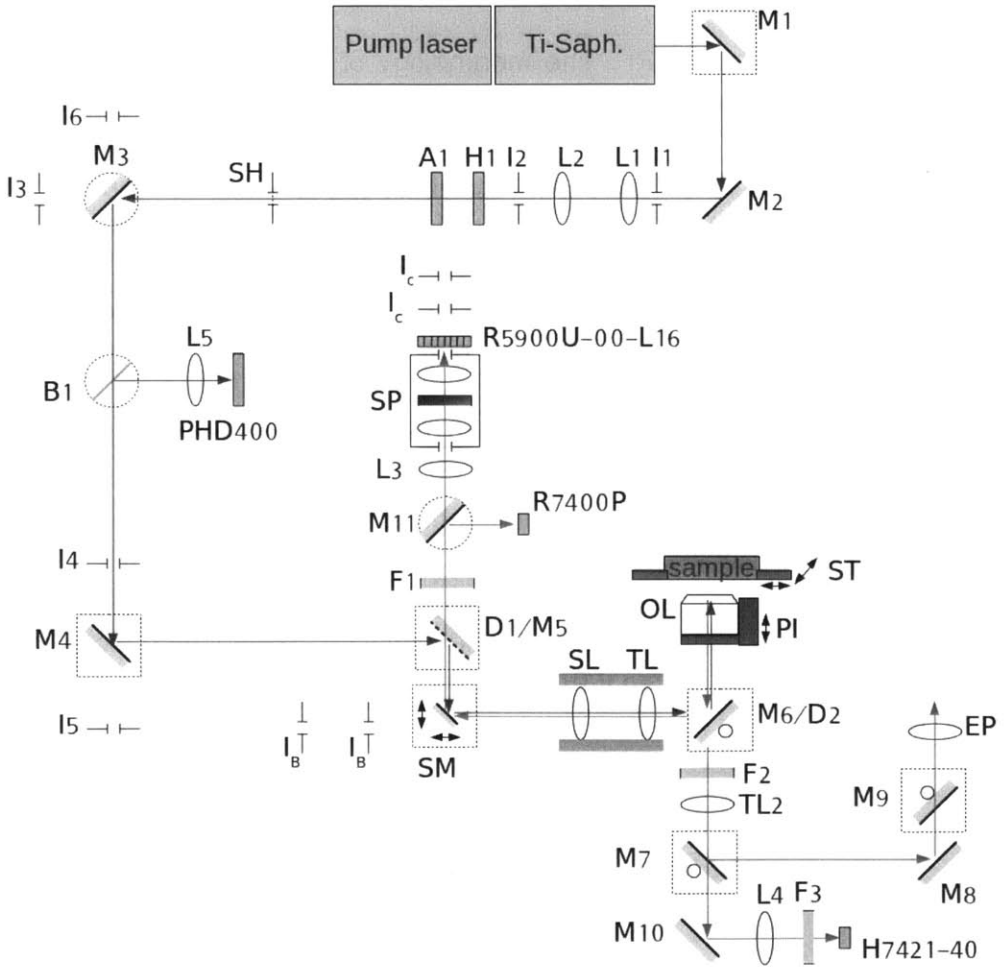


Figure A1.1: Schematic of the optical system. Red: illumination laser beam. Green: emission.

Component	Description & catalogue number
Pump laser	Millenia 10W laser, Spectra Physics
Ti-saph.	Tsunami titanium-sapphire laser, Spectra Physics
M1,M2,M3	Broadband ultrafast mirror 1" 700-930nm, Newport 10B20UF.25
M4,M5,M10,M11	Broadband metallic mirror 1" 480-20000nm, Newport 10D20ER.2
M6	Enhanced silver reflective mirror, Chroma Tech. 21010
M7,M8,M9	Metallic mirrors, part of the Zeiss Axiovert S100 microscope
L1	Plano-concave lens 1" f=-25mm, 650-1000nm, Newport KPC043AR.16
L2	Plano-convex lens 1" f=62.9mm, 650-1000nm, Newport KPX085AR.16
L3,L4	Plano-convex lens 1" f=25mm, 430-700nm, Newport KPX076AR.14
L5	Plano-convex lens 1" f=25mm, 430-700nm , Newport KPX076AR.14
SL	Scan lens, aspherical biconvex 1", f=25mm, Zeiss 444232-9902C
TL	Tube lens, plano-convex 1", f=125mm, 430-700nm, Newport KPX097AR.14
TL2	Tube lens for scanned detection configuration, f=165 mm, Zeiss
EP	Eyepiece, Zeiss
OL	C-apochromat 40× 1.2NA 0.28mm WD water immersion, Zeiss 441757-9970-000 Fluar 10× 0.5NA 1.9mm WD, Zeiss 440135-0000-000 Fluar 20× 0.75NA 0.6mm WD, Zeiss 440145-0000-000
I1...I9, IB, IC	Adjustable iris for system alignment
D1	Dichroic mirror T:350-670 R:720-1080, Chroma Tech. T700DCSPXRUV
D2	Dichroic mirror, Chroma Tech. 675DCSX
F1	Bandpass filter (394-680nm), Chroma Tech. ET680SP-2p8
F2	Bandpass filter, Chroma Tech. E700SP2P8 Bandpass filter BG39, schott
F3	Bandpass filter, Chroma Tech. E700SP2P8
H1	λ/2 plate 1", 780nm, Newport 10RP02-28
A1	Glan-Thomson polarizer 650-1000nm, Newport 10GT04AR.16
B1	Beamsplitter
SH	Shutter, melles griot
SM	x-y galvanometric scanning mirrors, Cambridge Technology 6350
PI	Piezoelectric actuator 100µm, Physik instrumente P-721.00
ST	Motorized x-y stage, Prior ES111
SP	1/8m f=120mm F/3.7 spectrograph, Oriel MS125 600 lines/mm λ_B=400nm ruled grating, Newport 77414
F-PD	Fast photodiode, Beckle & Hickle PHD-400
MA-PMT	16channel multialkali MA-PMT 300-600 nm, Hamamatsu R5900P-00-L16
PCH	Photon counting head GaAsP PMT 300-720nm, Hamamatsu H7421-40
F-PMT	Fast bialkali PMT 300-650nm, Hamamatsu R7400P

Table A1-1: Information about the optical components shown in the optics schematic of Fig. A1-1.

A2. Electronics System

The electronic system consists of three main parts. The first part are power electronics and drivers that control the various actuators of the microscope (scanning mirrors, x-y stage, piezoelectric objective actuator) based on signals provided by the microscope control GUI. The second part are electronics that process (condition, single-photon counting) the electric signal

provided by the three PMTs of the microscope. The third part are voltage regulators that provide appropriate voltage to various components, necessary for their operation.

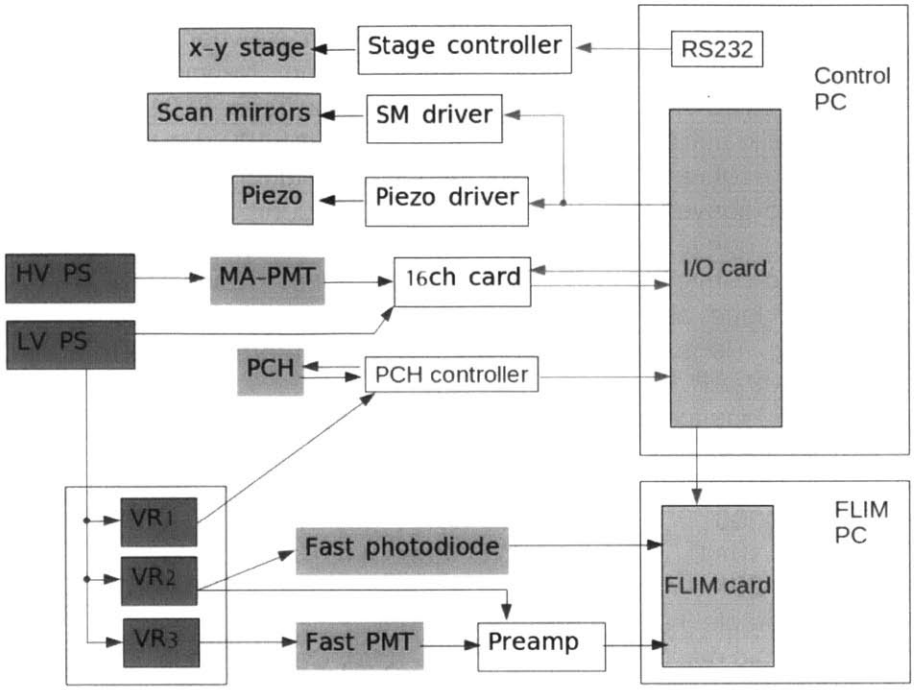


Figure A2.1: Electronics schematic. Red lines: DC power supply. Green lines: digital signals. Blue lines: analog signals (50Ω), Black lines: analog driver voltage. Green boxes: sensors. Red boxes: power supplies & regulator.

Component	Description & catalogue number
SM	x-y galvanometric scanning mirrors, Cambridge Technology 6350
SM controller	x-y galvanometric scanning mirror controller, Cambridge Technology 6350
PI	Piezoelectric actuator 100µm, Physik instrumente P-721.00
PI controller	Custom-made piezoelectric actuator power supply (0-100V), So lab
ST	Motorized x-y stage, Prior ES111
ST controller	Motorized x-y stage controller, Prior
F-PD	Fast photodiode, Beckle & Hickle PHD-400
MA-PMT	16channel multialkali MA-PMT 300-600 nm, Hamamatsu R5900P-00-L16
16ch card	Custom-made 16-channel counting card (Buehler et al. 2005), So lab
PCH	Photon counting head GaAsP PMT 300-720nm, Hamamatsu H7421-40
PCH controller	Photon counting head controller, Hamamatsu M9011
F-PMT	Fast bialkali PMT 300-650nm, Hamamatsu R7400P
Preamplifier	Preamplifier 26 dB 1.6GHz, Beckle & Hickle HFAC-26
HV PS	High voltage power supply (-800V)
LV PS	Low voltage power supply (12-14 V)
VR1	Voltage regulator 12V 3A, National semiconductor LM1085 IT-12
VR2	Voltage regulator 12V 400 mA, Micrel MIC2920A-12WT
VR3	Voltage regulator 9V 500 mA, ST microelectronicsLF90CV
I/O card	Custom made input/output PCI card. Contains an FPGA chip that reads inputs and communicates with the microscope GUI. So lab
FLIM card	Time-correlated single photon counting module, Beckle & Hickle SPC 730

Table A2-1: Information about the electronic components shown in the schematic of Fig. A2-1.

A3. Microscope Control and Acquisition Software

The operation of the microscope and the acquisition of images is controlled by appropriate software and firmware.

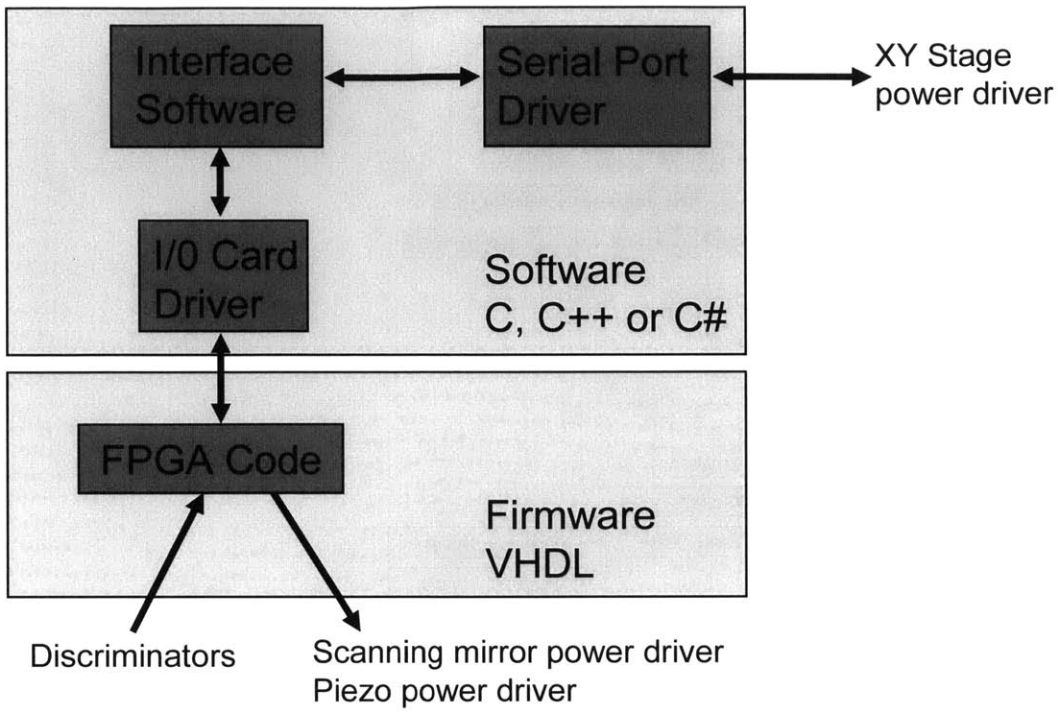


Figure A3.1: Overview of the software that controls the microscope and acquires data from the 16-channel MA-PMT.

The software that controls the microscope has four parts:

1. **Interface Software (confocal.exe):** Provides a GUI for user input and image display. Creates high-level commands for the z-piezo, the z-y stage and the scanning mirrors. Reads imaging data from I/O card. It is written in visual C++. The source code for the confocal application consists of many files, most of which are class definitions and declarations for the GUI: AcqMode.cpp, ConfocalFrame.cpp, PPSheet.cpp, Proppage_Acquisition.cpp, Proppage_Detector.cpp, Proppage_FCS.cpp, Proppage_IntgOpt.cpp, Proppage_PriorCont.cpp, Proppage_SaveOpt.cpp, Proppage_Scanner.cpp, Proppage_TRACK.cpp, and Toolbar_Acq.cpp and the corresponding header .h files.
   The user can run the confocal application in four different imaging modalities (four different ways to communicate with the FPGA chip, read and display the data): intensity imaging, particle tracking, pulse correlation and FCS. Classes named confocal… are related to intensity imaging, classes named track… to particle tracking, classes named pulse… are related to pulse correlation, and classes named FCS… are related to FCS. The AcqMode class corresponds to the dialog box where the user chooses which of the four to use.
   The confocal application uses five threads (intensity imaging mode). The first and most important is the CConfocalApp that probably corresponds to the confocal program itself. The second thread is CDataThread, probably related to data handling. The third is CViewThread that is probably related to data displaying. The last two classes are specific to the imaging modality used and are "children" of the previous two classes. For example, in the case of intensity imaging class CConfocalThread inherits from CDataThread and class CConfocalThreadView inherits from CViewThread.

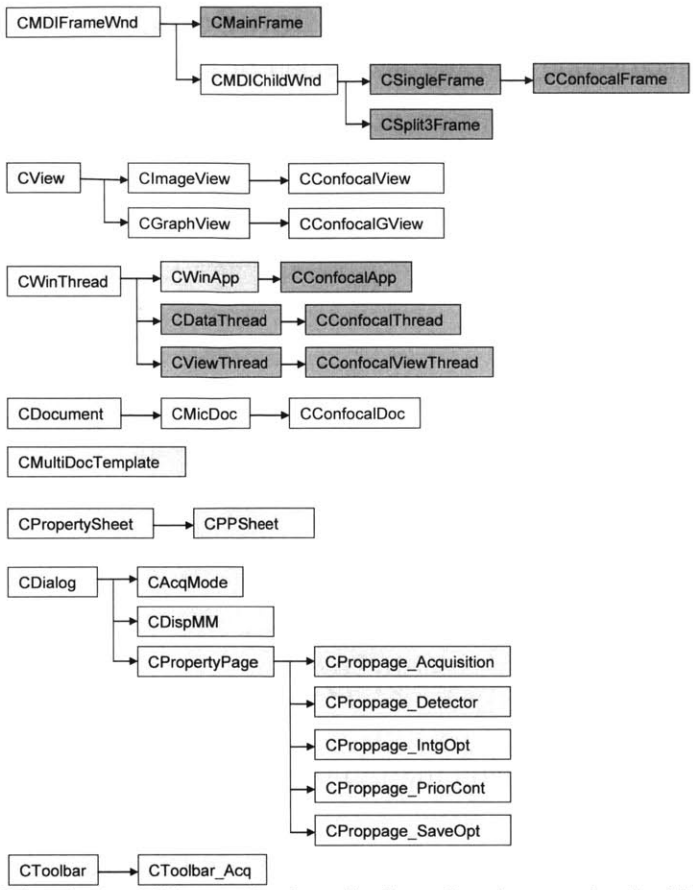


Figure A3.2: Hierarchy of the classes of the confocal application. Grey boxes: standard Microsoft foundation library classes. Yellow boxes: classes defined in the confocal application related to GUI functions. Green boxes: threads. Red boxes: classes defined in window handling.

2. **I/O Card Driver (cscan.dll and cscan.sys)**: Controls the communication between the computer and the I/O card through the PCI bus. The I/O card driver consists of two parts. The first part is the "cscan.dll", a DLL that controls the communication between the FPGA chip on the I/O card and the microscope GUI software (confocal). It is located on the folder "cscan" and was generated by a software tool called DriverAgent once provided by the company Numega (the company doesn't exist anymore). The second part is the "cscan.sys" that is located on the folder "cscan\cscan_agent". This is the software that handles the low-level communication between the computer and the FPGA chip.

3. **Serial Port Driver (SerialDevice.dll)**: Implements data exchange through the serial port of the computer. The "SerialDevice.dll" dynamically-link library that controls the serial port of the computer and provides functions for controlling the motion of the XY stage of the microscope (also known as the "prior" stage). The functions that control the stage have name format Prior… and their definitions are located in the file SerialDevice.cpp.

4. **FPGA Code:** i) Receives high-level commands from the interface software through the I/O Card Driver, translates these commands into low-level electronic control signals and sends these control signals to the power drivers of the piezo, XY stage and scanning mirrors, ii) receives the incoming digital electronic signal from the discriminators, translates them into "photon counts" and transmits this result to the interface software through the "I/O card driver". It is written in VHDL language. There are many versions of this program, depending on the type of sensor used.

Appendix B: Instrument Characterization

This Chapter provides protocols to characterize the instrument utilized in the thesis.

B1. Pixel Size Calibration

Pixel size P (in μm) depends on the objective magnification, and on the range of the scanning mirror motion (controlled by the "res" field of the confocal GUI).

Protocol

- Image a ronchi ruler of known spacing D (lines/μm). The lines of the ruler should be parallel to the y axis of the image
- Calculate the 2D FFT on the image. The magnitude of the 2D FFT should contain harmonics position on the k_x axis (spatial frequency along x axis). The first harmonic should be n pixels away from the DC frequency ($k_x = k_y = 0$). This corresponds to a spatial frequency of n/P μm^{-1} when then pixel size P is expressed in μm. This spatial frequency equals the frequency of the Runchi ruler D μm^{-1}, therefore the pixel size is calculated as:

  $$P = nD$$

Results

Res	32	16	8	4
40×	0.432	0.216	0.108	0.0054
10×	1.728	0.864	0.432	0.216

Table B1-1: Pixel size (μm) as a function of the objective magnification and the range of motion of the scanning mirrors (described by the "res" parameter).

Res	32	16	8	4
40×	110.6	55.3	27.7	13.8
10×	442.4	221.2	110.6	55.3

Table B1-2: Field of view (μm) of a 256×256 image as a function of the objective magnification and the range of motion of the scanning mirrors (described by the "res" parameter).

B2. Point Spread Function Measurement

This protocol describes a procedure to measure the point spread function (PSF) of the multi-photon microscope. The PSF is fit into a Gaussian beam model, where the square of excitation light intensity around the focus is modeled as

$$I^2(r,z) = I_0(1+(\frac{z}{z_R})^2)^{-2}\exp(\frac{-4r^2}{w^2(z)})$$

where $z_R \frac{\lambda}{\pi} = w_0^2$

Protocol

- Dilute 1:2000 0.02μm yellow-green fluorescent latex beads (F8787, invitrogen) in 2% agarose (dilute first 1:50 in H2O, then 1:500 in H20, then dilute in 2% agarose).
- Image beads at low power (2 mW in the sample, 25 mW after analyzer), 1 kHz sampling, 250 nm z-steps, 13.75 μm field of view (0.05371 μm pixel size).
- Pick image regions around single beads. Sum the intensity profiles of single beads over different z planes. Identify the focus plane and fit axial data to a Lorentzian profile (obtained by summing $I^2(r,z)$ over r:

$$I^2(z) = \alpha + \beta \cdot (1 + (\frac{z}{z_R})^2)^{-1}$$

- Identify the centroid location of the bead, sum $I^2(r,z)$ over z and then fit into:

$$I^2(r) = \alpha + \beta \cdot \exp(-(\frac{r}{w_r})^2)$$

Results

When using the C-apochromat 40× 1.2NA water-immersion objective, typical results are w_0 = 0.405 to 0.42 μm, z_R = 0.667 to 0.71 μm, and w_r = 0.1968μm. The PSF can be improved (reduce w_0) by overfilling more the objective BFP.

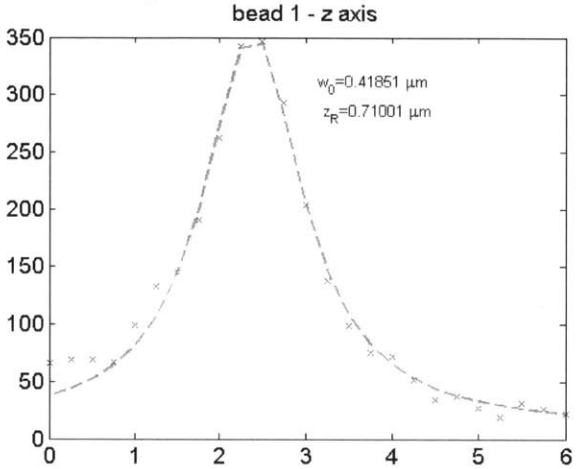


Figure B2.1: Typical axial PSF measurement obtained using a 40× 1.2NA objective. The measured PSF theoretically corresponds to a NA less than 1.2.

B3. Calibrating the Gains of the 16-Channel Discriminators

This protocol describes a procedure to measure the relative gains of the 16 PMT– single photon counting channels of the instrument. Due to slight variations in the PMT channels and the single-photon counting circuitry, the same emission signal is expected to produce slightly different photon counts by different channels. The objective of this protocol is to quantify this difference.

Protocol

- Use the SH emission of a collagen I sample to find spectrograph knob position so that SH signal is centered at each channel. These knob positions are called "centered" positions.
- Image a 300 μM fluorescein sample at all knob "centered" positions.
- Compare how the signal at each channel varies at different "centered" knob positions

Results

Figure B3.1 shows representative results. Each row contains the emission of fluorescein for a particular calibration of the spectrograph knob position. The calibration is described by the PMT channel that contains the SHG emission. The emission spectrum of fluorescein is visible in each row, e.g. when SHG is located at channel 1 (775nm excitation), then fluorescein emission peaks at channel 10. Each column therefore also contains the emission spectrum of fluorescein, as detected by a single channel. The emission detected by different channels differs by a multiplicative factor which is used to quantify the gain.

Gain is independent of the objective magnification. Identical results were obtained by 10× and 40× objectives, however the "centered" knob positions for each objective are slightly different. It is important to use a mono-chromatic emission (such as the SH emission) to describe the spectral calibration of the emission optics.

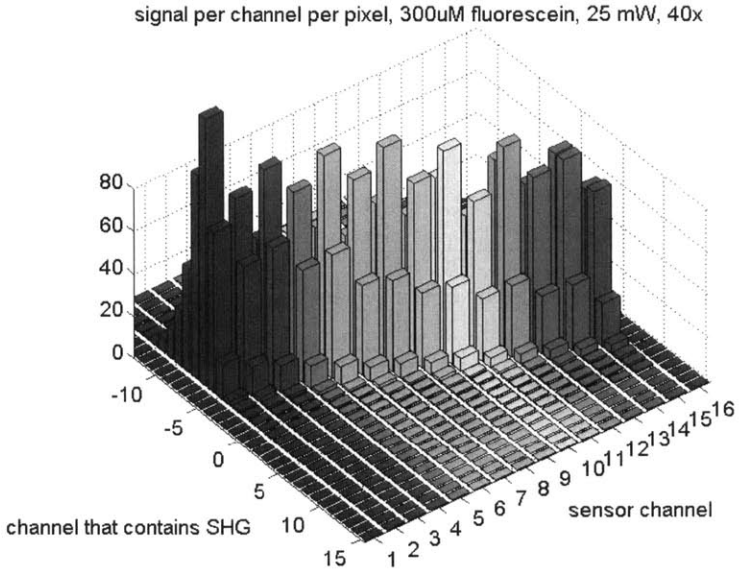


Figure B3.1: Typical data used to estimate PMT-discriminator gain.

Channel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Gain	1	0.96	0.99	0.93	0.94	0.95	1.00	0.95	0.97	0.94	1.02	0.97	1.00	0.95	0.95	0.94

Table B3-1: Gains for the R5900P-00-L16 MA-PMT sensor.

B4. Spectral Response Measurement

This protocol provides a way to quantify the spectral response of the instrument, which describes how a monochromatic emission generates photon counts to a neighborhood of channels centered around the channel that should ideally provide all photon counts based on geometric optics.

Protocol

- Image a collagen I sample at the "centered" knob positions (see Section B3 above).
- Quantify the spectral response based on the signal one particular channel at different "centered" knob positions. This signal should peak when the SH is centered in this channel, and should fade as the SH emission is centered in a different channel. The spectral response is symmetric.

Results

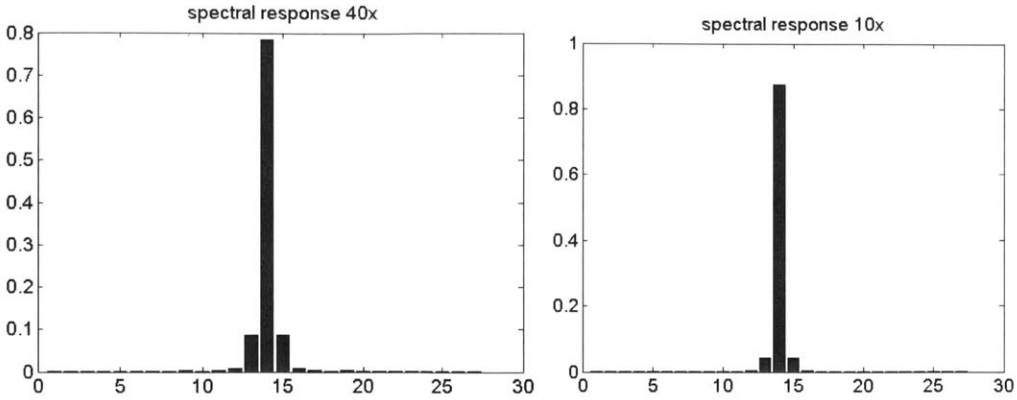


Figure B4.1: Spectral response of the emission optics system using a 40× objective (left), or tthe 10× objective (right).

	0	±1	±2	±3	±4
10× objective	0.8971	0.0437	0.0045	0.0019	0.0014
40× objective	0.8047	0.0870	0.0071	0.0025	0.0010

Table B4-1: Spectral response of the detection system using a 10× or a 40× objective.

Appendix C: Spectrograph Design

C1. System Description

The existing spectrograph available in the So Lab (oriel 77400) consists of [Buehler et al. 2005]:

- input concave mirror MS1, f_s = 120 mm.
- diffraction grating DG (model oriel 77400, grating spacing 600 lines/mm, d = 1667 nm, blazing wavelength λ_B = 400nm, blazing angle θ_B = 6.89 deg, max efficiency η_{max}=0.85).
- output concave mirror MS2, f_s = 120 mm.

It is of interest to evaluate if the diffraction grating should be substituted in order to optimize the spectrograph performance in the wavelength region of interest [380 nm, 620 nm].

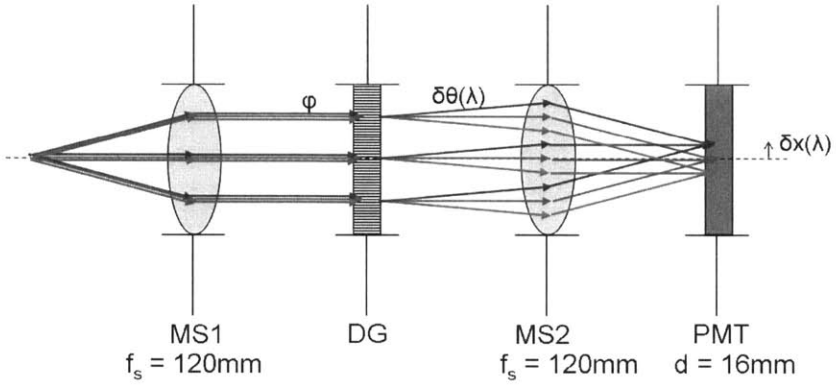


Figure C1.1: Schematic of the spectrograph and MA-PMT assembly.

C2. Light Diffraction by a Grating

When a ray of monochromatic light of wavelength λ is incident on a diffraction grating (width W, ruling spacing d = G^{-1}, brazing angle b) at incident angle φ, then it will be reflected at several directions. The field of the light that is reflected towards angle θ is given by the following convolution [James 2007]:

$$\Gamma(p) = W \cdot \sin c(\pi W p) \otimes [\sin c(\pi \frac{p-q}{G}) \cdot \Xi_G(p)]$$

where $p = \frac{\sin(\varphi)+\sin(\theta)}{\lambda}$, $q = \frac{2\sin(b)}{\lambda}$, $\Xi_G(x) = \sum_{n=-\infty}^{\infty} \delta(x - nG)$ and ⊗ describes the convolution operator.

Light will be reflected towards discrete directions (called "orders"), specified by the pulsetrain function $\Xi_G(x)$. For a particular incident angle φ (for sign nomenclature check [Palmer and Lowen 2005, James 2007]) the direction of the m-th order is given by:

$$p = mG \;\Rightarrow\; \sin(\theta) = mG\lambda - \sin(\varphi)$$

This direction depends only on φ, λ and G. It does not depend on the grazing angle b. The case m=0 (zero order) corresponds to reflection. In spectrographs the mode of interest is m=1.

The efficiency of the m-th mode is defined as the percentage of the incident light intensity that it reflected towards mode m, and can be calculated analytically as:

$$\eta_m = \frac{(\sin c(\pi(\frac{p-q}{G}))^2}{\sum_{pG=-\infty}^{\infty}(\sin c(\pi(\frac{p-q}{G}))^2} = \frac{(\sin c(\pi(m-\frac{q}{G}))^2}{\sum_{m=-\infty}^{\infty}(\sin c(\pi(m-\frac{q}{G}))^2} = \frac{(\sin c(\pi(m-\frac{\lambda_B}{\lambda}))^2}{\sum_{m=-\infty}^{\infty}(\sin c(\pi(m-\frac{\lambda_B}{\lambda}))^2}$$

Comments:

- The efficiency of each mode depends on the wavelength of the incident light λ
- At one particular wavelength (brazing wavelength) all light is diffracted into order

$$\lambda_B = \frac{2\sin(b)}{G} \Rightarrow q=G$$

- The efficiency of the first order as a function of λ equals:

$$\eta_1 = \eta_{\max} \frac{(\sin c(\pi(1-\frac{\lambda_B}{\lambda}))^2}{\sum_{m=-5}^{5}(\sin c(\pi(m-\frac{\lambda_B}{\lambda}))^2}$$

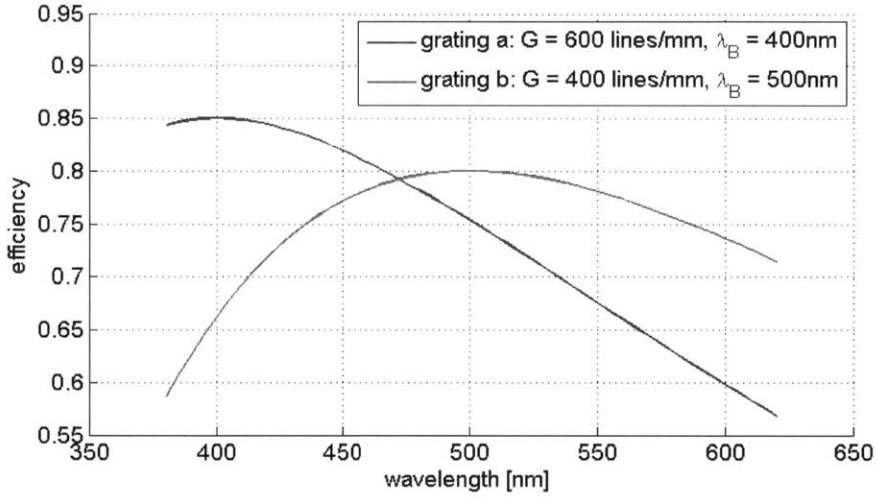


Figure C2.1: Efficiency of the first order for the diffraction grating of the instrument (a) and an alternative diffraction grating (b) in the wavelength region of interest.

C3. Nominal Beam and Detected Range of the Spectrum

For the particular instrument considered here, the wavelength region of interest lies between 380 and 620 nm. The nominal ray corresponds to a beam of wavelength λ_0 = 500 nm (the center wavelength of the wavelength region of interest) that is focused on axis on the input plane of the spectrograph entrance slit. The beam is collimated by mirror MS1, gets incident on the diffraction grating at angle φ_0, its first order is diffracted at angle θ_0 (deviation angle $K_0 = \theta_0 - \varphi_0$), and then is focused by mirror MS2 into the center of the MA-PMT area. Since the exact value of $K_0 = \theta_0 - \varphi_0$ is not known exactly (varies slightly after each instrument alignment), the nominal beam is calculated for three values K_0 = {20, 25, 30} deg. For each value of K_0, the corresponding value of φ_0 is calculated from:

$$\sin(K_0 + \varphi_0) + \sin(\varphi_0) = G\lambda_0$$

For an input beam of arbitrary wavelength λ incident on the grating at the same angle φ_0, the angle θ(λ) where the 1st order of wavelength λ is reflected is calculated as:

$$\theta(\lambda) = \sin^{-1}(G\lambda - \sin(\varphi_0)) \qquad \lambda \in [\lambda_{\min} \quad \lambda_{\max}] = [380 \quad 620] nm$$

$$\theta_{\min} : \theta(\lambda_{\min})$$

$$\theta_{\max} : \theta(\lambda_{\max})$$

The min and max values of θ correspond to the min and max values of the wavelengths of interest λ_{min} = 380 nm and λ_{max} = 620 nm. Light of wavelength λ different than the nominal λ_0 will be reflected to angle θ(λ) and will be focused by lens MS2 on the PMT surface at distance δx away from the position where the nominal ray λ_0 is focused.

$$\delta x(\lambda) = f_s \tan(\theta(\lambda) - \theta_0)$$

$$\delta x_{\min} : \delta x(\lambda_{\min})$$

$$\delta x_{\max} : \delta x(\lambda_{\max})$$

The spectrum range $[\lambda_l \quad \lambda_u]$ that can be imaged by the PMT (when λ_0 is focused on the center of the PMT) can be calculated based on the total width of the PMT sensor W_p =16mm (the width of each PMT channel is c_p = 1mm) and the focal length f_s =120 mm of mirror MS2:

$$\lambda_l = G^{-1}(\sin(\varphi_0) + \sin(\theta_0 - \tan^{-1}(W_P / 2f_s)))$$

$$\lambda_u = G^{-1}(\sin(\varphi_0) + \sin(\theta_0 + \tan^{-1}(W_P / 2f_s)))$$

The range of wavelengt $[\lambda_l^j \quad \lambda_u^j]$ that is detected by each channel can be calculated by finding the wavelengths λ_i that are focused on positions $[x_l^j \quad x_u^j]$ on the i-th channel of the PMT

$$\lambda_l^j = G^{-1}(\sin(\varphi_0) + \sin(\theta_0 + \tan^{-1}(x_l^j / f_s))) \quad , \; x_l^j = -W_c/2 + (j-1)\cdot p_c \quad , \text{j = 1, 2, ..., 16}$$

$$\lambda_u^j = G^{-1}(\sin(\varphi_0) + \sin(\theta_0 + \tan^{-1}(x_u^j / f_s))) \quad , \; x_u^j = -W_c/2 + j \cdot p_c \quad , \text{j = 1, 2, ..., 16}$$

The spectral width that is detected by sensor channel j equals $SR_j = \lambda_u^j - \lambda_l^j$

Nominal beam			*Detected spectrum*		
K_0 (deg)	φ_0 (deg)	θ_0 (deg)	θ_{min} ... θ_{max} (deg)	λ_l ... λ_u [nm]	[SR_1, SR_8, SR_{16}] (nm/pixel)
20	-1.24	18.76	14.46 ... 23.18	393.9 ... 603.8	13.4, 13.2, 12.8
25	-3.66	21.33	16.98 ... 25.85	395.0 ... 601.5	13.2, 13.0, 12.6
30	-6.06	23.94	19.48 ... 28.52	397.8 ... 600.4	13.0, 12.7, 12.3

Table C3-1: Nominal beam (λ_0 = 500 nm), detected spectrum range ([λ_l, λ_u]), and spectral width per channel (SR_i) for the instrument utilized in this thesis, for three different values of the deflection angle K_0.

The detected spectral width per channel is less for red channels compared to blue channels, and it depends slightly on K_0.

C4. Deviation Due to Off-axis Focusing on the Input Plane

Nominal ray calculations are valid when the emission light is focused on the optic axis on the input slit of the spectrograph. When the emission light is focused δy_i away from the optic axis, the resulting incident angle on the diffraction grating will be

$$\varphi^* = \varphi_0 + \delta\varphi = \varphi_0 - {\delta y_i}/{f_s}$$

The resulting reflection angle of the 1st order for wavelength λ is:

$$\theta^* = \sin^{-1}(G\lambda - \sin(\varphi^*))$$

$$\theta^*(\lambda) = \theta(\lambda) + \delta\theta(\lambda) \Rightarrow \qquad \sin(\delta\theta) = -\frac{\cos(\varphi_0)}{\cos(\theta(\lambda))}\sin(\delta\varphi) = \frac{\cos(\varphi_0)}{\cos(\theta(\lambda))}\sin({\delta y_i}/{f_s})$$

or approximately:

$$\delta\theta = \frac{\cos(\varphi_0)}{\cos(\theta(\lambda))}\frac{\delta y_i}{f_s}$$

The resulting position displacement δy_s of the focus point for the beam of wavelength λ on the PMT surface is:

$$\delta y_s = f_s(\tan(\theta^*(\lambda)) - \tan(\theta(\lambda)))$$

Which can be approximated by:

$$\delta y_s \approx f_s \delta\theta(\lambda) = \frac{\cos(\varphi_0)}{\cos(\theta(\lambda))}\delta y_i$$

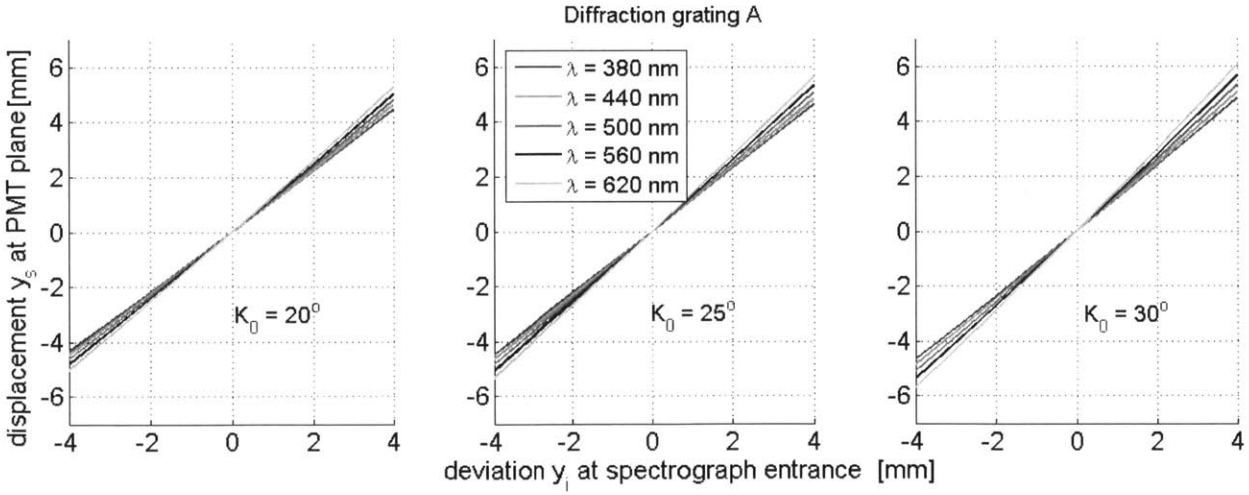


Figure C4.1: Deviation of focus y_s on the PMT plane as a function of off-axis position y_i for various wavelengths λ.

Comments:

- the deviation y_s on the PMT plane is larger than the deviation y_i in the spectrograph entrance, because usually $|\varphi_0| < |\theta(\lambda)|$.
- y_s is approximately a linear function of y_i
- The focus of red emission is shifted more than the focus of blue emission. This means that in microscopes of "scanned" configuration, the spectrum region that will be detected by each

channel will depend on the pixel position. The proportionality constant K of $y_s \approx K \cdot y_i$ depends on the grating and on the nominal deviation angle K_0 (greater K_0 results in greater K), and can vary significant as a function of the system setup and calibrartion.

C5. Literature Cited

Buehler C, Kim K H, Greuter U, Schlumpf N, and So PTC. Single-photon counting Multi-Color Multi-Photon Fluorescence Microscope, *J. of Fluorescence* **15**(1): 41-51, 2005.

James J. *Spectrograph design fundamentals*, Cambridge university press, Cambridge, 2007.

Palmer C, and Loewen E. *Diffraction grating handbook*, Sixth Edition, Newport Corporation, 2005.

Appendix D: Maximum Likelihood Solution of the Poisson Mixing Model by the Expectation-Maximization Algorithm

The expectation-maximization (EM) algorithm finds the maximum of a log-likelihood function $L(\underline{\lambda} \mid \underline{y})$ through an iterative two-step process [Dempster et al. 1977]. It is assumed that the observed "incomplete" data $\underline{y}$ corresponding to random variables $\underline{\mathrm{Y}}$ arise from a set of unobserved "hidden" random variables $\underline{\mathrm{X}}$. The complete "data" $\underline{Z} = [\underline{X}, \underline{Y}]$ are described through the pdf $p_Z(\underline{X}, \underline{Y}; \underline{\theta})$, where $\underline{\theta}$ is the vector of parameters to be estimated via ML. The "expectation" step of EM calculates the expectation $Q(\underline{\theta}, \hat{\underline{\theta}})$ of $\ln(p_Z(\underline{X}, \underline{Y}; \underline{\theta}))$ with respect to the hidden random variables $\underline{\mathrm{X}}$ conditioned on the measurement $\underline{y}$ and the current estimate of the parameters $\hat{\underline{\theta}}$

$$Q(\underline{\theta}, \hat{\underline{\theta}}) = E[\ln(p_Z(\underline{Z}; \underline{\theta})) \mid \underline{y}, \hat{\underline{\theta}}] = \int_X \ln(p_Z(\underline{Z}; \underline{\theta})) p_X(\underline{x} \mid \underline{y}; \hat{\underline{\theta}}) dx$$

The "maximization" step of EM finds an updated estimate for $\underline{\theta}$ by maximizing $Q(\theta, \hat{\theta})$. The EM iteration generates a sequence of updates for $\hat{\underline{\theta}}^{(k)}$ that converges on the ML estimate of $\hat{\underline{\theta}}$.

The derivation of the EM solution of the Poisson mixing model described in Chapter 2 follows the derivation described in [Candy 2009]. The observed variables $\underline{\mathrm{Y}}$ are the photon counts detected on the sensor channels, while the hidden variables are the number of photons $\underline{N} = [N_1 \quad N_N]^T$ emitted by each source. The Poisson mixing model provides the complete likelihood of the source emission rate vector $\underline{\lambda}$ equals:

$$p_Z^{NL}(\underline{Z}; \underline{\lambda}) = \prod_{i=1}^{N} \prod_{j=1}^{M} \{ \frac{\exp(-\lambda_{ij}) \lambda_{ij}^{n_{ij}}}{n_{ij}!} \}$$

n_{ij} is the number of photons emitted by source i and detected by detector j (such that $\sum_{j=1}^{M} n_{ij} = n_i$, and $\sum_{i=1}^{N} n_{ij} = y_j$). The corresponding random variable N_{ij} is a Poisson process of rate $\lambda_{ij} = s_{ji} \lambda_i$. The complete log-likelihood Λ_Z is:

$$\Lambda_Z = \ln(p_Z^{NL}(\underline{Z}; \underline{\lambda})) = -\|\underline{\lambda}\|_1 + \sum_{i=1}^{N} \sum_{j=1}^{M} \{ n_{ij} \ln(s_{ji} \lambda_i) \} - \sum_{i=1}^{N} \sum_{j=1}^{M} \{ \ln(n_{ij}!) \}$$

which can be written in the form of an exponential family [Wornell 2010]:

$$\Lambda_Z = -\alpha(\underline{\lambda}) + \underline{c}^T(\underline{\lambda}) \underline{t}(\underline{z}) + \beta(\underline{z})$$

$$\alpha(\underline{\lambda}) = \|\underline{\lambda}\|_1 = \sum_{i=1}^{N} \lambda_i \,,\ \beta(\underline{z}) = -\sum_{i=1}^{N}\sum_{j=1}^{M} \{\ln(n_{ij}!)\} \,,\ \underline{c}(\underline{\lambda}) = \left[\ln(\lambda_{11}) \quad \ldots \quad \ln(\lambda_{NM})\right]^T$$

$$\underline{t}(\underline{z}) = \left[n_{11} \quad \ldots \quad n_{NM}\right]^T$$

In this case, the M-step becomes

$$Q(\underline{\lambda}, \hat{\underline{\lambda}}) = E[\Lambda_Z \mid \underline{y}, \hat{\underline{\lambda}}] = -\alpha(\underline{\lambda}) + \underline{c}^T(\underline{\lambda}) \cdot E[\underline{t}(\underline{z}) \mid \underline{y}, \hat{\underline{\lambda}}] + E[\beta(\underline{z}) \mid \underline{y}, \hat{\underline{\lambda}}]$$

Since the last term is not a function of $\underline{\lambda}$, it will not affect the E-step, and can be dropped:

$$\hat{Q}(\underline{\lambda}, \hat{\underline{\lambda}}) = -\alpha(\underline{\lambda}) + \underline{c}^T(\underline{\lambda}) \cdot E[\underline{t}(\underline{z}) \mid \underline{y}, \hat{\underline{\lambda}}] = -\sum_{i=1}^{N} \lambda_i + \sum_{i=1}^{N}\sum_{j=1}^{M} \{\ln(s_{ij}\lambda_i) \cdot E[n_{ij} \mid \underline{y}, \hat{\underline{\lambda}}]\} .$$

In the E-step, the next estimate for $\underline{\lambda}$ is chosen as $\hat{\underline{\lambda}}^{(k+1)} = \arg\max_{\underline{\lambda}} \hat{Q}(\underline{\lambda}, \hat{\underline{\lambda}}^{(k)})$:

$$\left.\frac{\partial \hat{Q}(\underline{\lambda}, \hat{\underline{\lambda}}^{(k)})}{\partial \lambda i}\right|_{\underline{\lambda}=\underline{\lambda}^{(k+1)}} = -1 + \sum_{j=1}^{M} \{s_{ij} \cdot \frac{E[n_{ij} \mid \underline{y}, \hat{\underline{\lambda}}^{(k)}]}{s_{ij}\lambda_i^{(k+1)}}\} = -1 + \frac{1}{\lambda_i^{(k+1)}} \sum_{j=1}^{M} \{E[n_{ij} \mid \underline{y}, \hat{\underline{\lambda}}^{(k)}]\} = 0 \Rightarrow$$

$$\lambda_i^{(k+1)} = \sum_{j=1}^{M} \{E[n_{ij} \mid \underline{y}, \hat{\underline{\lambda}}^{(k)}]\} \qquad , \text{i = 1,2,...,N}$$

The expected value of n_{ji} given the measurement y_j in channel j equals [Shepp and Vardi 1982]

$$E[n_{ij} \mid \underline{y}, \hat{\underline{\lambda}}] = y_j \frac{s_{ji}\lambda_i^{(k)}}{\sum_{l=1}^{N} s_{jl}\lambda_l^{(k)}}$$

Therefore the EM algorithm becomes:

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} \sum_{j=1}^{M} \{y_j \frac{s_{ji}}{\sum_{l=1}^{N} s_{jl}\lambda_l^{(k)}}\}$$

Or in matrix form

$$\underline{\lambda}_{NL}^{(k+1)} = diag(\underline{\lambda}_{i,NL}^{(k)}) \cdot \underline{S}^T \cdot diag(d_j(\underline{\lambda}_{i,NL}^{(k)}))^{-1} \cdot \underline{y}$$

D1. Literature Cited

Candy JV. *Bayesian Signal Processing. Classical, Modern and Particle Filtering Methods*, Wiley, 2009.

Dempster AP, Laird NM, and Rubin DB. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. Royal Statist. Soc. B.* **39**(1): 1-38, 1977.

Shepp LA, and Vardi Y. Maximum Likelihood Reconstruction for Emission Tomography, *IEEE Trans. Med. Imag.* **1**(2): 113-122, 1982.

Wornell GW. *Lecture Notes for Course 6.437* (Inference and Information). Massachusetts Institute of Technology, 2010.

Appendix E: Fabrication of Porous Collagen Scaffolds

E1. Overview

The procedures for fabricating porous collagen scaffolds applied in this study are based on the freeze-drying process, and follow the methodology established in the Yannas Lab over the past 30 years [Dagalakis et al. 1980; Yannas and Burke 1980; Yannas et al. 1980; Harley 2006; Yannas et al. 2010]. The outcome of the fabrication process is a dry scaffold sheet that can be used in two major kinds of experiments: i) *in vivo* experiments of inducing skin regeneration in animals where the scaffold is used as a graft for severe skin injuries [Yannas et al. 1982, 1989; Troxel 1994; Yannas 1998, 2001], ii) *in vitro* studies of cell-matrix interactions, where the scaffold is seeded with cells and is used as an analog of tissue ECM [Freyman et al. 2001a, 2001b; Zaleskas et al. 2001, 2004; Corin 2005; Soller 2005; O'Brien et al. 2005, 2007; Harley et al. 2007a, 2008; Keogh et al. 2010; Caliari et al. 2011, 2012; Murphy et al. 2012]. Scaffold fabrication consists of four major processes:

- **Preparation of collagen suspension**: Microfibrillar collagen I fibers and other components (in this study glycosaminoglycans (GAG) or fibronectin isoforms) are dissolved in 0.05M acetic acid. The resulting suspension contains collagen particles and air bubbles.
- **Freeze-drying**: The collagen suspension is poured into a mold that provides the shape of the final dry scaffold. The aqueous part is removed by first freezing the suspension, and then removing the resulting ice crystals via sublimation. What remains is a dry porous sponge-like material whose struts contain the solid material (collagen, GAG, ECM proteins) that was trapped between ice crystals during freezing.
- **Scaffold Cross-linking**: Formation of additional intra-molecular bonds in the scaffold in order to improve resistance to proteinase degradation and increased the material stiffness. Cross-linking can be implemented by thermal (dehydro-thermal treatment) or chemical (chemical cross-linkers) means.
- **Sample preparation**: *In vivo* applications of collagen scaffolds in skin and peripheral nerve regeneration require scaffolds of relative large dimensions (on the order of cm), and planar or tubular geometry respectively. In contrast *in vitro* applications of scaffolds for studying cell-scaffold interactions require multiple copies of small samples (size around 5-8mm), which are prepared by cutting scaffold sheets.

The following sections describe each process in more detail.

E2. Suspension Preparation

Suspension Types

Four types of collagen-based suspensions were fabricated by homogenizing collagen fibers in 0.05M acetic acid using a dispersing element (IKA Works, Inc., Wilmington, NC). The collagen mass fraction in all four suspensions is approximately 0.5%. Collagen suspension (denoted as "0.5C100") consists of banded micro-fibrillar collagen particles. Gelatin suspension (denoted as

"0.5G100") consists of gelatinized collagen (made using the same collagen source as collagen suspension). Gelatinization of collagen is implemented by keeping the suspension during homogenization at elevated temperature. Collagen-gelatin suspensions (denoted as "0.5CxGy" where x and y correspond to the percentage contribution of collagen and gelatin suspensions correspondingly) are mixtures of collagen suspension and gelatin suspension of various proportions. Collagen-GAG suspension (denoted as "0.5CG100") consists of co-precipitant particles of microfibrillar collagen I and chondroitin-6-sulfate and is the standard suspension described in [Harley 2006] and is the closest match for the material used in skin regeneration studies [Yannas et al. 1982, 1989].

Homogenization

During suspension homogenization, raw collagen I fibers (and potentially other ECM components) is dissolved into 0.05M acetic acid pH 3.2. Table E2-1 summarizes the blending protocol parameters for preparing a 50 ml suspension batch. The baseline is the collagen suspension fabrication protocol. First, m_{Coll} = 250 mg collagen I (Integra LifeSciences, Plainsboro, NJ) is dissolved into V_1 = 50 ml 0.05M acetic acid inside a glass beaker. The suspension is homogenized by blending for t_b = 12.5 min using a dispersing element spinning at 15000 rpm. During blending the suspension is kept inside iced water and its temperature is monitored so that it does not exceed 10°C. The resulting collagen suspension is partially transparent, and the most viscous from all four suspension types.

The fabrication of the gelatin suspension differs in that during blending the suspension is heated at 65°C using a dry heater underneath the glass beaker. Gelatin suspension is more transparent and much less viscous compared to collagen suspension. The preparation of collagen-gelatin suspension consists of blending appropriate volumes of collagen suspension and gelatin suspension for 3 min while monitoring the suspension so that its temperature does not exceed 10°C. The properties of collagen-gelatin suspensions depend on the composition (contribution of collagen and gelatin parts).

The preparation of collagen-glycosaminoglycan (GAG) suspension is divided in two steps. Initially m_{Coll} = 250 mg collagen is blended in V_1 = 41.6 ml 0.05M acetic acid for $t_b/2$ = 6.25 min. Then m_{GAG} = 22.2 mg chondroitin-6-sulfate GAG (Sigma-Aldrich Chemical Co., St. Louis, MO) is added drop-wise to the suspension while blending. The collagen-GAG suspension is blended for $t_b/2$ minutes more. The presence of GAG lowers the suspension viscosity and reduces viscosity-induced heat dissipation. Collagen-GAG suspension is less viscous than pure collagen suspension, a bit more viscous compared to gelatin suspension, and has a white milky-like color (unlike collagen and gelatin suspensions that look partially transparent and appear turbid). The size of collagen I particles in a collagen-GAG suspension lie in 40 – 150 μm size [Lee 1994].

It is possible to prepare suspensions of volume other than 50 ml. In this case the values of t_b, m_{coll}, m_{GAG}, V_1, V_2 need to scale analogous to the suspension volume.

Suspension	Abbreviation	t_b [min]	T_b [°C]	m_{Coll} [mg]	V_1 [mL]	m_{GAG} [mg]	V_2 [mL]
Collagen	0.5C100	12.5	<10	250	50	-	-
Gelatin	0.5G100		65			-	-
Collagen-gelatin	0.5CxGy*		<10			-	-
Collagen-GAG	0.5CG100		<10		41.6	22.2	8.4

Table E2-1: Parameters for preparing collagen-based suspension (0.5% collagen mass fraction). *x and y correspond to the percentage contribution of collagen and gelatin in the suspension (x+y = 100)

Degassing

After homogenization, all suspensions contain trapped air bubbles, whose size ranges from the micron level to centimeter level. It is necessary to remove the large-size bubbles in order to prevent suspension inhomogeneity and the formation of large irregular-sized pores in the final scaffold. Degassing is implemented by transferring the suspension into a large-surface Erlenmeyer flask and apply 15 Pa vacuum for 90 min while keeping the flask in iced water.

The effect of applying vacuum for removing air bubbles from a suspension depend on: i) vacuum applied, ii) degassing duration, vacuum-suspension surface area (need to use an Erlenmeyer flask of volume at least 10 times the volume of the suspension), and iv) the suspension viscosity. Since the four suspensions described above (collagen, gelatin, collagen-gelatin, collagen-GAG) have different viscosities, it is expected that the same vacuum treatment should remove air bubbles more effectively in low-viscosity suspensions (gelatin-rich, collagen-GAG) than high-viscosity suspensions (collagen-rich). However at no experiment was conduct to verify this claim and it relies on empirical observations from the turbidity of the degassed suspensions.

Degassing should not be extensive (very high vacuum, long time) otherwise this interferes with the porosity of the final freeze-dried scaffold. Extensive degassing in collagen and gelatin suspensions is inferred by significant reduction in the turbidity of the suspension (caused by the removal of tiny air bubbles).

After degassing, suspensions can be used at 4°C until use (within 1-2 months). At this temperature, all suspensions remain stable (no visible change in color, turbidity, viscosity was observed) with the exception of the gelatin suspension, which progressively gels at 4°C.

Doping Collagen Suspension with Fibronectin Fragments

Collagen scaffolds can be thought of as a simple analog of the extracellular matrix (ECM) found in tissues. Although in many tissues collagen I is the main component of ECM, the composition of tissue ECM is much more rich, and contains multiple additional components such as other collagen isoforms (e.g. collagen III), other scaffold proteins (e.g. fibronectin), glycosaminoglycans and proteoglycans [Hay 1991]. It makes sense therefore to try to fabricate more complex scaffolds, made of collagen I plus other key ECM molecules. The preparation of collagen-GAG scaffolds can be thought of as a step in this direction [Yannas et al. 1980].

In this study, 0.5% mass fraction collagen suspension is doped with fibronectin isoforms, expressed in e-coli and purified via metal ion affinity chromatography (poly-His tag) [Bloom et al. 1999]. Due to the high cost of purchasing or purifying ECM components, it is impractical to prepare suspensions of large enough volume (more than >20ml) that can be homogenized using a dispersing element as described above. In this case, it is necessary to prepare the suspension using a different strategy.

Small batches (360µl) of fibronectin-doped collagen suspension were prepared by adding 288µl 0.5% collagen suspension (0.5C100), 36 µl solution of purified fibronectin isoforms, and 36µl 0.05M acetic acid. The doped suspension is homogenized by repetitive pipeting. The mass fraction of the resulting fibronectin-doped collagen suspension is 0.4%. The fibronectin mass corresponds to approximately 2% of the total solid mass. It is important to reconstitute the purified fibronectin isoforms in 0.05M acetic acid before mixing with the collagen suspension. Since the isolectric point (pH of minimum solubility) is approximately 5.2 [Tooney et al. 1983] (lies between the pH of the initial buffer (PBS) and the final 0.05M acetic acid) there is danger of fibronectin precipitation during buffer exchange. To avoid fibronectin precipitation around the isolectric point, the purified fibronectin fragments are diluted in PBS (density less than 0.3 mg/ml) before changing buffer, and then are concentrated using an AMICON 3 kDA MWCO centrifugal filter unit (Millipore, Billerica, MA). The resulting 360 µl fibronectin-doped collagen suspension has too small volume to be used with the ordinary 125x125 mm stainless steel freeze-drying pan. Instead it is used to fabricate six cylindrical scaffold samples of 5mm diameter and 3mm height using miniature 5mm-diamater freeze-drying pans cut in an elastomer sheet.

E3. Freeze-Drying

Freeze-drying consists of two major steps. In the freezing step the collagen suspension (suspension) is cooled. This causes the majority of the water content to form ice crystals. These ice crystals are surrounded by a glassy phase that contains the collagen particles. In the drying step, the water content is removed via sublimation by applying high vacuum. The final outcome consists of a porous sponge-like material consisting of struts (correspond to the glassy collagen phase that was trapped between ice crystals after freezing) and pores (correspond to the ice crystals that were removed by sublimation).

In practice, freeze-drying is implemented by pouring the collagen suspension in a mold, and applying an appropriate temperature/pressure profile in a lyophilizer. The structural parameters of the resulting scaffold (pore diameter, homogeneity) depend on the physicochemical and thermodynamic properties of the collagen suspension, the temperature and pressure profile applied, and the thermal properties of the mold used [Schoof et al. 2001; Zmora et al. 2002]. A more detailed description of the freeze-drying process for the fabrication of porous collagen scaffolds can be found in [Harley 2006].

Freezing

In the freezing step the collagen suspension is cooled so that it solidifies. Specifically, as the suspension temperature T_s drops below the acetic acid solution melting temperature T_m (a function of the acetic acid mole fraction η_{HAc}; initially $\eta_{HAc}(0)\approx 0.001$), ice crystal nuclei start to form. The rate of ice nuclei formation depends on suspension undercooling T_s - T_m and the

presence of impurities in the system. When the suspension undercooling is enough, the newly-formed ice nuclei survive and grow and form ice grains (crystals). As water molecules form ice crystals, the remaining acetic acid and any acetic-acid soluble "impurities" (in this case collagen particles) are excluded from the ice and stay in a liquid phase. As freezing proceeds (more ice crystals form, and existing ice crystals grow) the remaining liquid phase contains collagen particles in increasingly concentrated acetic acid. The thermodynamics of the pure water-acetic acid system suggest that as η_{HAc} increases T_m drops (Fig. E3-1 left) and that the lowest temperature that water and acetic acid can survive in liquid phase is -25°C. However, the freezing response of a collagen suspension deviates from the response of the pure water - acetic acid system because the collagen particles interfere with the formation of a crystal solid phase. Instead, as the temperature drops, when the suspension temperature reaches the glass transition temperature T_g of the collagen-acetic acid system, the remaining collagen suspension in the boundaries of ice grains solidify into a glassy phase (vitrification) [Tang and Pikal 2004]. The glass transition temperature T_g of the collagen particle suspension in acetic acid used in this application is unknown.

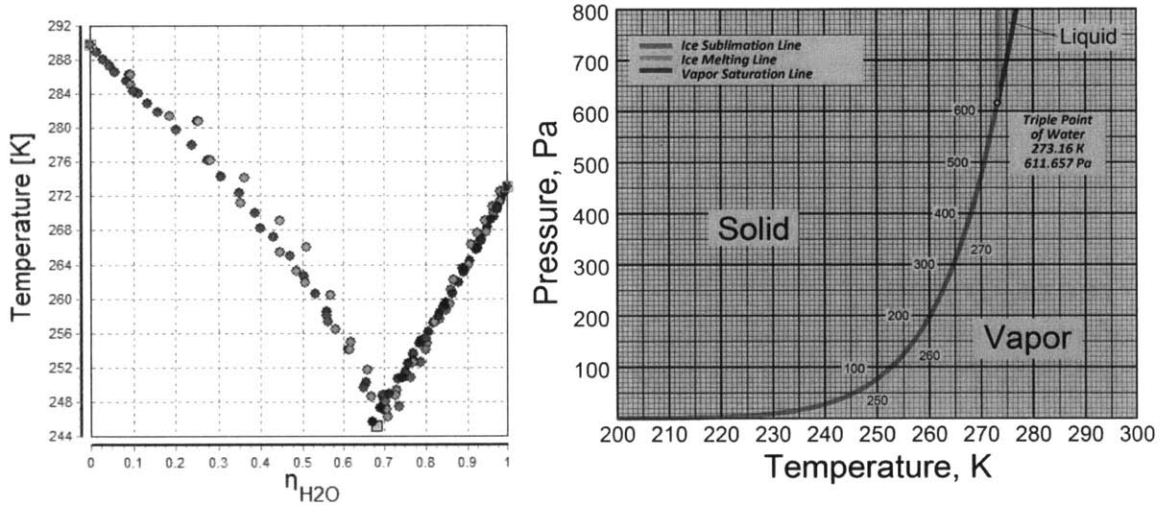


Figure E3.1: Thermodynamic properties that affect the fabrication of porous collagen scaffolds via freeze-drying. Left: Phase diagram (solidus temperature as a function of mole fraction of water η_{H2O}) of the acetic acid – water system. A eutectic point is present at (η_{H2O} = 0.7, T = 248K). Data kindly provided by the Dortmund Data Bank (http://www.ddbst.com). Right: Phase diagram (temperature, pressure) of water at low pressures and temperatures. Source: http://en.wikipedia.org/wiki/Water_(data_page)

In this study, the freezing process follows the protocols and practice described in [O'Brien et al. 2004; Harley 2006]. Collagen suspension is poured in 18-gauge 304 stainless steel molds (thickness: 1.27mm, heat capacity: 503 $J \cdot kg^{-1} \cdot K^{-1}$, thermal conductivity: 20 $W \cdot m^{-1} \cdot K^{-1}$). The suspension is then frozen in a lyophilizer (VirTis, Gardiner, NY) by controlling the lyophilizer shelf's temperature. The shelf temperature starts initially at room temperature, then is help constant at -5°C for 15 min (ensure homogenous temperature in the whole suspension; this small supercooling is not enough to induce ice crystal formation), and is set to ramp to final temperature T_f = -40°C with a rate of -1K/min. For this particular thermal system, this rate was found to result in fabricating scaffolds of homogenous mean pore diameter, in contrast to quenching (very rapid cooling) that produces scaffolds of inhomogenous pore diameter.

After the shelf temperature reaches the final shelf temperature T_f = -40°C, the shelf temperature is kept at T_f = -40°C for 1h more. This step, named "annealing" in [Harley 2006], provides thermal energy and time so that neighboring ice crystals merge, therefore increasing the mean pore diameter of the scaffold (coarsening). [Tang and Pikal 2004] mentions that annealing will take place as long as the frozen suspension temperature raises above T_g (the glass transition temperature of the collagen-acetic acid solution system just before vitrification).

The pore microstructure of the resulting scaffolds depends significantly on the thermal treatment (time profile of the temperature of the lyophilizer shelf) during the freezing step. It was found that by varying the final shelf temperature T_f (while keeping the shelf cooling rate at -1K/min) it was possible to control the mean pore diameter of the scaffold [O'Brien et al. 2004]. It was suggested that the scaffold pore microstructure is modulated mainly by ice crystal coarsening [Harley 2006]. However, experimental practice suggests that there are other factors that also affect decisively the resulting pore microstructure of the scaffold. In particular, unpublished observations of this thesis suggest that the presence of air bubbles in the suspension (not removed by degassing) affects significantly the resulting scaffold mean pore diameter. For given collagen suspension composition, using suspension that has undergone extensive degassing (high vacuum 3 hours, on ice) leads to scaffolds of much smaller mean pore size compared to using suspension degassed according to the protocol described above. The presence of much fewer air bubbles in the extensively degassed suspension was inferred by the significantly reduced turbidity of the suspension. Furthermore, experimental observations suggest that suspension viscosity and chemical composition also affect the resulting pore microstructure (for given thermal treatment during freezing). Viscosity affects the ability of degassing treatments to remove bubbles. Suspension composition and viscosity are expected to affect the kinetics of ice crystal formation (nucleation, growth, and coarsening) as well as the glass transition temperature. The biggest difference was observed between scaffolds made of 0.5C100 suspension and 0.5 G100 suspension, since gelatin slurry has much less viscosity compared to collagen. This study did not pursue further to quantify in detail the effects of suspension composition and viscosity in the resulting scaffold pore microstructure.

Drying

In the drying step the vast majority of the water content of the original collagen suspension is removed via sublimation. In pharmaceutical applications, drying consists of two steps. During primary drying, the majority of the water (water that froze into ice crystals) is removed by carefully applying high vacuum in low temperature. During secondary drying, most of the remaining water (water that did not froze and is located in the scaffold struts) is removed by applying high vacuum in elevated temperature [Tang and Pikal 2004].

In the established protocol for fabricating porous collagen scaffolds, primary drying was conducted by applying vacuum <300 mTorr at 0°C for 17 hours, while secondary drying was skipped [Harley 2006]. According to this practice, after annealing the lyophilizer pressure drops to 300 mTorr, and then temperature rises from T_f= -40°C to 0°C. This quick crossing of the water sublimation line is expected to induce sudden sublimation in the frozen suspension. This sudden change of phase is the corresponding phenomenon to the sudden freeze of water induced by rapid cooling of water (quenching).

Based on scientific literature, the temperature of primary drying T_p needs to be chosen carefully. If T_p is close to the sublimation temperature T_s, then the sublimation rate is slow (major drawback in pharmaceutical applications because drying is the longest and most energy-consuming part of the freeze-drying process). On the other hand, elevated T_p induces fast sublimation (limited by the mass transfer capability of the lyophilizer), however when T_p exceeds a critical "macroscopic collapse temperature" T_c, then the remaining solid structure (in this case the struts of the collagen scaffold) will collapse [Tang and Pikal 2004]. The collapse temperature T_c, the temperature above which the freeze-dried product loses macroscopic structure, is usually 2°C higher than the glass transition temperature T_g of the collagen-acetic acid solution just before vitrification [Tang and Pikal 2004].

In this study, the primary drying protocol is slightly different compared to [Harley 2006]. After freezing and annealing, the shelf temperature drops to -50°C (thermocouples placed on the mold show that the resulting temperature of the frozen suspension is around -40°C). Sblimation is then gradually induced by first applying 150 mTorr pressure, and then ramping the shelf temperature to T_p = -20°C within one hour (the sublimation temperature of water at 150 mTorr is approximately -36 °C, Table E3-1). Primary drying is conducted at -20°C, 150 mTorr for 12 hours. Secondary drying is conducted by applying 150 mTorr at 0°C for 12 hours more.

T [oC]	-65	-60	-55	-50	-45	-40	-35	-30	-25	-20	-15	-10
P [Pa]	0.54	1.08	2.08	3.92	7.17	12.79	22.27	37.90	63.13	103.6	165.0	259.7
P [mT]	4.0	8.1	15.6	29.4	53.8	95.9	167.0	284.3	473.5	777.1	1238	1948

Table E3-1: Saturated pressure (in Pa and mTorr) of vapor over ice at various temperatures [Sonntag and Heinze 1982]. Data correspond to the low-pressure part of Fig E3.1. right.

Based on the literature, secondary drying of amorphous freeze-dry products is a slow process that takes place at elevated temperatures T_s (e.g. 40°C for 6h). Secondary drying should take place only when primary drying is over (all ice crystals have been sublimated), which can be detected by either measuring the moisture inside the lyophilizer chamber or can be inferred by monitoring the mold temperature (ice sublimation causes mold cooling). The rise of the shelf temperature from T_p to T_s should be slow (0.1°C/min) to avoid collapse of the freeze-dried product [Tang and Pikal 2004]. In this study secondary drying took place at 0°C for 12h. In the last step of freeze drying, the shelf temperature is set equal to room temperature, so that when the scaffold is removed from the lyophilizer, no moisture condensation is collected on the scaffold struts.

E4. Scaffold Cross-Linking

After freeze-drying, the scaffold is cross-linked in order to increase its resistance to degradation (by enzymes secrete by cells) and to increase its mechanical stiffness [Harley et al. 2007b]. Cross-linking involves the formation of covalent bonds between different molecules in the scaffold struts. In this study, cross-linking is implemented by two methods:

- Dehydro-thermal treatment (DHT): the absence of moisture drives intermolecular condensation reactions in the scaffold struts that lead to scaffold cross-linking. DHT is implemented by applying vacuum (50 mTorr) at elevated temperature (105°C to 120°C) [Yannas and Tobolsky 1967]. Longer DHT duration and higher DHT temperature increase the resulting stiffness of collagen-GAG scaffolds [Harley et al. 2007; Haugh et al. 2009].

Experimental results suggest that increasing DHT temperature is more effective compared to increasing DHT duration (the vast majority of crosslinking takes place within the first 24h), however DHT temperature higher than 150°C should be avoided because it induces collagen denaturation. DHT treatment has the advantage that it provides a dry and sterile scaffold. However its ability to increase the material stiffness is limited. Specifically, DHT can increase the compressive and tensile moduli of collagen-GAG scaffolds 2-fold and 3.8 fold respectively [Haugh et al. 2009]. In this study, all kinds of scaffolds (collagen, gelatin, collagen-gelatin hybrids, collagen-GAG, collagen-fibronectin) were cross-linked by DHT treatment at 120°C for 48 hours. This DHT treatment was chosen because it was the one that provided tubular collagen scaffolds of "optimal" ability to induce regeneration in peripheral nerves [Harley et al. 2004; Soller et al. 2012].

- Chemical crosslinking: In this method covalent intermolecular bonds in the collagen struts are produced using small molecules that are reactive towards particular chemical groups present in the scaffold molecules. Such small molecules (cross-linkers) are used widely in biochemistry for bioconjugation [Hermanson 2008]. In this study, chemical cross-linking was implemented using an carboxyl-amine coupling conjugation chemistry consisting of the crosslinkers 1-Ethyl-3-(3-dimethylaminopropyl)carbodiimide hydrochloride (EDC) and N-Hydroxysulfosuccinimide (NHS). This is a method widely applied in previous studies of collagen-GAG scaffolds [Harley 2006]. The degree of scaffold crosslinking by EDC-NHS chemistry can be modulated by controlling the cross-linker amount to collagen amount and the duration of the cross-linking reaction. EDC molecules produce intermediate groups reactive to primary amines, which can be quenched using ethanolamine.

E5. Scaffold Sample Preparation

The outcome of the freeze-drying process is a dry scaffold sheet. On the other hand, cell-scaffold interaction experiments or biomaterial characterization experiments need much smaller cylindrical samples of scaffolds. These samples are cut from the original sheet using biopsy punches (Miltex).

E6. Differences Compared to Previous Fabrication Practice

The protocols described in this Appendix are based on established protocols for fabricating porous collagen scaffolds described in [Harley 2006; Yannas et al. 2010], which have been modified in several ways to achieve particular goals:

First, the volume of suspension preparation batch was reduced approximately 20 times. The standard practice for fabricating sheet scaffolds was to prepare suspension batches of 720 ml volume and 0.5% collagen mass fraction, store the suspension at 4°C, and use 65 ml suspension to fabricate each 125x125x3 mm sheet [Harley 2006]. Downscaling the batch size from 720 ml to 30-50 ml reduced the required collagen I mass per batch from 3600 mg to 150 - 250 mg. The motivation for downscaling suspension batch size was the cost and availability of raw collagen I. The raw collagen I used in this work was a proprietary microfibrillar collagen I provided to the Yannas lab by Integra Life Sciences in 2000. This collagen I supply is not available commercially, and the quantity of raw collagen I provided to the Yannas lab by Integra was limited, making necessary to minimize its consumption. Many alternative commercially available collagen products are tropocollagen preparations (e.g. Sigma C7661, BD 354231 and

354236, Santa Cruz Biotech Sc-29009, R&D 3447-020-01, Millipore 08-115) that cost $1 - $5 per mg. The lack of fibrillar organization and their high cost, makes the use of these products for scaffold fabrication questionable or impractical. Other commercial collagen products (Sigma C3511, C9791, C9879, USB 13815) are purified from bovine skin or tendon tissues (therefore are likely to contain collagen fibrils) and cost much less ($0.02 – $2 per mg). Although some of these materials have been used in published studies to fabricate porous scaffolds (for example [Klapperich and Bertozzi 2004; Javorski and Klapperich 2006]), trial suspension preparations using these products resulted in suspensions of significantly different properties (viscosity, color) and in scaffolds of significantly different structure (presence of impurities, strut geometry) compared to the scaffolds fabricated using Integra collagen. Based on the observed differences in suspension viscosity and color, possible reasons why several alternative fibrillar collagen products fail to replicate the outcomes of Integra micro-fibrillar collagen are the presence of salts or other impurities, and gelatinization of collagen, both by-products of the collagen purification process. Nevertheless, we haven't been able to test other commercially available microfibrillar collagen I supplies, for example Geistlich (Geistlich Pharma AG, Wolhusen, Switzerland) or Devro (Devro Pty Ltd, Bathurst, NSW, Australia) [Kanungo and Gibson 2010].

Second, the dimensions of the fabricated scaffolds were reduced. This reduction was motivated by several reasons: i) match suspension requirements with the new collagen suspension batch size and consume less suspension, ii) reduce collagen waste: collagen scaffolds should be used within 1-2 months after their fabrication in order to prevent degradation, iii) the need to fabricate a large variety of similar but not identical scaffolds (e.g. making scaffold libraries). The available stainless steel molds provide 125x125mm scaffold sheets. Each such sheet can provide either 150 8-mm diameter scaffold samples or 220 5-mm diameter scaffold samples. In many cases, the demand of collagen scaffold samples is much smaller, leading to degradation (and disposal) of the uncut fabricated sheet. Scaffold dimensions were reduced in two ways:

- Reduce the mold size. This was achieved by dividing the original 125x125 stainless steel pan into four equal quadrants using an adhesive elastomer as a spacer. Each quadrant of the pan was approximately 58x58 mm wide. The resulting 58x58 sheet can provide approximately 25 8mm-diameter or 64 5mm-diameter cylindrical samples, that is adequate samples for many experimental procedures.
- Reduce the thickness of the scaffold sheet. This is achieved by using less collagen suspension per pan area. In this study the required suspension volume was calculated assuming a sheet thickness of 3mm. The required suspension volume per 58x58x3 sheet is approximately 10 ml. Many applications require or would be benefitted by even thinner scaffolds. For example, fluorescent imaging of cells inside scaffolds takes place usually within 100μm from the scaffold surface. Thinner scaffolds would make ligand density measurements easier since they accelerate the diffusion of proteins inside scaffolds, therefore reduce required incubation duration. However, fabricating scaffolds thinner than 2 mm using the existing stainless steel pans was quite challenging, because thin scaffolds are hard to remove them from the mold without stretching or tearing, and are hard to cut using biopsy punches. The alternative idea of using a 2-piece mold (where a 1mm-thick suspension is sandwiched between two heat-conductive surfaces) was tried with significant success (homogenous-looking scaffolds) but was not pursued further because the mold

properties affect significantly the mean pore diameter of the produced scaffolds and there was no time for a detailed optimization of the freeze-drying protocol.

- Fabricate individual cylindrical scaffold samples rather than fabricating sheets. This approach could reduce significantly raw material waste (the leftover scaffold sheet after cutting cylindrical samples with a punch). Fabricating scaffold samples in miniature molds minimizes the consumption of ECM raw materials. For example, fabricating a 5mm-diameter 3mm-thick scaffold sample requires 60μl collagen suspension, with contains 300 μg collagen (for 0.5% collagen mass fraction). Doping such a scaffold with an additional ECM components by would require 15 μg ECM protein (5% doping), an amount that is feasible to purchase or express recombinately. This idea was implemented by covering the surface of a stainless steel pan with a 3mm-thick elastomer that contains holes of either 5mm or 8mm diameter. These holes acted like miniature molds from freeze-drying scaffolds. This approach was used to fabricate 5mm-diameter 3mm-thick cylindrical scaffold samples doped with recombinant fibronectin fragments (due to the difficulty in purifying fibronectin fragments) of 0.4% mass fraction. This approach suffers from the "skin" effect, which is observed during freeze drying. Specifically, apart from the pan side of the scaffold, all other sides seem to be covered by a "skin" layer of reduced porosity that makes scaffolds less permeable to cells or chemical solutions.

E7. Literature Cited

Bloom L, Ingham KC, and Hynes RO. Fibronectin Regulates Assembly of Actin Filaments and Focal Contacts in Cultured Cells via the Heparinbinding Site in Repeat III13. *Mol. Biol. Cell* **10**(5): 1521-36, 1999.

Caliari SR, Harley BA. The effect of anisotropic collagen-GAG scaffolds and growth factor supplementation on tendon cell recruitment, alignment, and metabolic activity. *Biomaterials* **32**(23): 5330-40, 2011.

Caliari SR, Weisgerber DW, Ramirez MA, Kelkhoff DO, Harley BA. The influence of collagen-glycosaminoglycan scaffold relative density and microstructural anisotropy on tenocyte bioactivity and transcriptomic stability. *J Mech Behav Biomed Mater.* **11**: 27-40, 2012.

Corin KA, *Inhibition of myofibroblast contraction*, S.M. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 2005.

Dagalakis N, Flink J, Stasikelis P, Burke JF, and Yannas IV, Design of artificial skin. III. Control of pore structure, *J. Biomed. Mat. Res.* **14**: 511-28, 1980.

Freyman TM, *Development of an In Vitro Model of Contraction by Fibroblasts*, PhD Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 2001.

Freyman TM, Yannas IV, Pek YS, Yokoo R, Gibson LJ. Micromechanics of fibroblast contraction of a collagen-GAG matrix. *Exp Cell Res.* **269**(1): 140-53, 2001a.

Freyman TM, Yannas IV, Yokoo R, Gibson LJ. Fibroblast contraction of a collagen-GAG matrix. Biomaterials. **22**(21): 2883-91, 2001b.

Harley BA, *Cell-Matrix Interactions : Collagen-GAG Scaffold Fabrication, Characterization, and Measurement of Cell Migratory and Contractile Behavior via Confocal Microscopy*, PhD Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 2006.

Harley BA, Spilker MH, Wu JW, Asano K, Hsu HP, Spector M, and Yannas IV, Optimal degradation rate for collagen chambers used for regeneration of peripheral nerves over long gaps. *Cells Tissues Organs* **176**(1-3):153-65, 2004.

Harley BA, Freyman TM, Wong MQ, Gibson LJ, A New Technique for Calculating Individual Dermal Fibroblast Contractile Forces Generated within Collagen-GAG Scaffolds, *Biophys. J.* **93**(8): 2911-22, 2007a.

Harley BA, Leung JH, Silva EC, and Gibson LJ, Mechanical characterization of collagen-glycosaminoglycan scaffolds, *Acta Biomater.* **3**(4): 463-74, 2007b.

Harley BA, Kim HD, Zaman MH, Yannas IV, Lauffenburger DA, and Gibson LJ, Microarchitecture of Three-Dimensional Scaffolds Influences Cell Migration Behavior via Junction Interactions, *Biophys. J.* **95**(8): 4013-24, 2008.

Haugh MG, Jaasma MJ, and O'Brien FJ, The effect of dehydrothermal treatment on the mechanical and structural properties of collagen-GAG scaffolds, *J Biomed Mater Res A.* **89**(2):363-9, 2009.

Hay ED, *Cell Biology of Extracellular Matrix*, Second Edition, Plenum Press, 1991.

Hermanson GT, *Bioconjugate Techniques*, Second Edition, Academic Press, 2008.

Hiemenz PC, and Lodge TP, *Polymer Chemistry*, Second Edition, CRC Press, 2007.

Javorski J, and Klapperich CM, Fibroblast remodeling activity at two- and three-dimensional collagen-glycosaminoglycan interfaces, *Biomaterials* **27**(23): 4212-20, 2006.

Kanungo BP, and Gibson LJ. Density-property relationships in collagen-glycosaminoglycan scaffolds. *Acta Biomater.* **6**(2):344-53, 2010.

Keogh MB, O'Brien FJ, and Daly JS. Substrate stiffness and contractile behaviour modulate the functional maturation of osteoblasts on a collagen-GAG scaffold. *Acta Biomater.* **6**(11): 4305-13, 2010.

Klapperich CM, and Bertozzi CR, Global gene expression of cells attached to a tissue engineering scaffold, *Biomaterials* **25**(25): 5631-41, 2004.

Lee JB, *The effect of mixing conditions on the distribution of collagen-GAG particle size*, B.S. Thesis, Department of Mechanical Engineering, MIT, 1994.

Murphy CM, Matsiko A, Haugh MG, Gleeson JP, O'Brien FJ. Mesenchymal stem cell fate is regulated by the composition and mechanical properties of collagen-glycosaminoglycan scaffolds, *J Mech Behav Biomed Mater.* **11**:53-62, 2012.

O'Brien FJ, Harley BA, Yannas IV, and Gibson LJ. Influence of freezing rate on pore structure in freeze-dried collagen-GAG scaffolds, *Biomaterials* **25** (6): 1077-86, 2004.

O'Brien FJ, Harley BA, Yannas IV, and Gibson LJ, The effect of pore size on cell adhesion in collagen-GAG scaffolds, *Biomaterials* **26**(4): 433-41, 2005.

O'Brien FJ, Harley BA, Waller MA, Yannas IV, Gibson LJ, and Prendergast PJ. The effect of pore size on permeability and cell attachment in collagen scaffolds for tissue engineering. *Technol Health Care* **15**(1): 3-17, 2007.

Schoof H, Apel J, Heschel I, and Rau G. Control of pore structure and size in freeze-dried collagen sponges. *J Biomed Mater Res.* **58**(4): 352-7, 2001.

Soller EC, *In vitro pharmacological inhibition of myofibroblast differentiation and force generation in a collagen-GAG matrix*, S.M. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 2005.

Soller EC, Tzeranis DS, Miu K, So PT, Yannas IV. Common features of optimal collagen scaffolds that disrupt wound contraction and enhance regeneration both in peripheral nerves and in skin, *Biomaterials* **33**(19): 4783-91, 2012.

Sonntag D, und Heinze D, *Sättigungsdampfdruck- und Sättigungsdampfdichtetafeln für Wasser und Eis*, 1. Aufl., VEB Deutscher Verlag für Grundstoffindustrie, 1982.

Tang X, and Pikal MJ. Design of freeze-drying processes for pharmaceuticals: practical advice. *Pharm Res.* **21**(2): 191-200, 2004.

Tooney NM, Mosesson MW, Amrani DL, Hainfeld JF, Wall JS. Solution and surface effects on plasma fibronectin structure. *J Cell Biol.* **97**(6): 1686-92, 1983.

Troxel KS, *Delay of skin wound contraction by porous collagen-GAG matrices*, PhD Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 1994.

Yannas IV, *Tissue and Organ Regeneration in Adults*, Springer, 2001.

Yannas IV, Studies on the biological activity of the dermal regeneration template, *Wound Rep. Reg* **6**: 518-24, 1998.

Yannas IV, and Tobolsky AV, Cross linking of gelatine by dehydration, *Nature* **215**(100): 509-10, 1967.

Yannas IV, and Burke JF, Design of artificial skin. I. Basic design principles, *J. Biom. Mat. Res.* **14**: 65-81, 1980.

Yannas IV, Burke JF, Gordon PL, Huang C, and Rubenstein RH, Design of artificial skin. II. Control of chemical composition, *J. Biomed. Mat. Res.* **14**: 107-31, 1980.

Yannas IV, Burke JF, Orgill DP, Skrabut EM, Wound Tissue Can Utilize a Polymeric Template to Synthesize a Functional Extension of Skin, *Science* **215**(4529): 174–6, 1982.

Yannas IV, Lee E, Orgill DP, Skrabut EM, and Murphy GF, Synthesis and Characterization of a model extracellular-matrix that induces partial regeneration of adult mammalian skin, *Proc Natl Acad Sci U.S.A.* **86**: 933-7, 1989.

Yannas IV, Tzeranis DS, Harley BA, and So PTC, Biologically active collagen-based scaffolds: advances in processing and characterization. *Philos Transact A Math Phys Eng Sci.* **368**(1917): 2123-39, 2010.

Zaleskas JM, Kinner B, Freyman TM, Yannas IV, Gibson LJ, and Spector M, Growth Factor Regulation of Smooth Muscle Actin Expression and Contraction of Human Articular Chondrocytes and Meniscal Cells in A Collagen-GAG Matrix. *Exp. Cell Res.* **270**:21-32, 2001.

Zaleskas JM, Kinner B, Freyman TM, Yannas IV, Gibson LJ, Spector M. Contractile forces generated by articular chondrocytes in collagen-glycosaminoglycan matrices. *Biomaterials* **25**(7-8): 1299-308, 2004.

Zmora S, Glicklis R, and Cohen S. Tailoring the pore architecture in 3-D alginate scaffolds by controlling the freezing regime during fabrication. *Biomaterials* **23**(20): 4087-94, 2002.

Appendix F: Scaffold Fabrication Protocols

Contents

F.1. Preparation of Collagen-Based Suspensions

Based on [Harley 2006]

This protocol describes the preparation of 0.5% (mass collagen/suspension mass) 50 ml collagen, gelatin, collagen-gelatin or collagen-GAG suspension (suspension).

Supplies

- Type I microfibrillar bovine tendon collagen (Integra LifeSciences, Inc., Plainsboro, NJ); stored at 4°C.
- Distilled, deionized water.
- Glacial Acetic Acid (Mallinckrodt Chemical Co., Paris, KY).
- Chondroitin 6-sulfate from shark cartilage (Cat. No. C-4384, Sigma-Aldrich Chemical Co., St. Louis, MO); stored at 4°C.

Equipment

- Dispersing element (Ultra Turrax T18 Overhead blender, IKA Works, Inc., Wilmington, NC)
- Vacuum pump.

Procedure

Part I: Preparation

- Prepare 60 ml 0.05M acetic acid solution (for suspension preparation).
  - Add 174 ul glacial acid to 60 ml ddH_2O (2.9 µl acetic acid per ml ddH_2O).
  - Cool the acetic acid at 4°C for 15 minutes.
- Clean the mixing chamber and the dispersion element components with 0.05 M acetic acid and wipe them with kimwipes to remove dust and remaining content from the previous run.
- Incubate the dispersing element at 4°C for 10 min inside an autoclave pouch.

Part II: Suspension Homogenization

The following protocols describe the preparation of 50 ml collagen suspension (a), or gelatin suspension (b), or collagen-gelatin suspension (c), or collagen-GAG suspension (d). To prepare a different suspension volume, then scale V_1, V_2, m_{Coll}, t_b accordingly.

Collagen Suspension Preparation

- Transfer V_1 = 50 ml 0.05M acetic acid into a 100 ml beaker.
  - Keep the beaker inside a bucker filled with iced water for the remaining steps.
- Add m_{Coll} = 250 mg micro-fibrillar bovine tendon collagen I to V_1 ml 0.05M acetic acid.
- Incubate suspension at 4°C for ten minutes so that collagen swells in the acetic acid.
- Assemble the dispersing element.
- Blend the collagen suspension at 15,000 rpm (setting 3.25) for t_b = 12.5 minutes.
  - Monitor the suspension temperature. When the suspension temperature reaches 10°C, stop blending. Resume when the suspension temperature drops to 7°C.
  - Soon collagen fibers get suspended into the acetic acid, and the suspension becomes viscous and heats up quickly.

Gelatin Suspension Preparation

- Transfer V_1 = 50 ml 0.05M acetic acid into a 100 ml beaker
- Add m_{Coll} = 250 mg micro-fibrillar bovine tendon collagen I to V_1 ml 0.05M acetic acid.

- Incubate suspension at room temperature for ten minutes so that collagen swells in the acetic acid.
- Place the beaker on top of a dry heater. Set the heater temperature so that the suspension temperature reaches 65 °C.
- Assemble the dispersing element.
- Blend the collagen suspension at 15,000 rpm (setting 3.25) for t_b = 12.5 minutes.
  - Collagen fibers get suspended in acetic acid. The suspension doesn't get too viscous.

Collagen-Gelatin Suspension Preparation

- Prepare a collagen suspension and a gelatin suspension as described in sections (a, b).
- To prepare 20 ml collagen-gelatin suspension of X% collagen composition (and Y=100-X % gelatin composition), add 0.2X ml collagen suspension and 20-0.2X ml gelatin suspension in a 50 ml glass beaker.
  - Keep the beaker inside a bucker filled with iced water for the remaining steps.
- Assemble the dispersing element.
- Blend the collagen suspension at 15,000 rpm (setting 3.25) for t_b = 3 minutes.
  - Monitor the suspension temperature. When the suspension temperature reaches 10°C, stop blending. Resume when the suspension temperature drops to 7°C.

Collagen-GAG Suspension Preparation

- Transfer V_1 = 41.6 ml 0.05M acetic acid into a 100 ml beaker
  - Keep the beaker inside a bucker filled with iced water for the remaining steps.
- Add m_{Coll} = 250 mg microfibrillar bovine tendon collagen I to V_1 ml 0.05M acetic acid.
- Incubate suspension at 4°C for ten minutes so that collagen swells in the acetic acid.
- Assemble the dispersing element.
- Blend the collagen suspension at 15,000 rpm (setting 3.25) for $t_b/2$= 6.25 min.
  - Monitor the suspension temperature. When the suspension temperature reaches 10°C, stop blending. Resume when the suspension temperature drops to 7°C.
  - Soon collagen fibers get suspended into the acetic acid, and the suspension becomes viscous and heats up quickly.
- Prepare GAG solution: dissolve m_{GAG} = 22.2 mg chondroitin-6-sulfate in V_2 = 8.4 ml 0.05M acetic acid.
- Use a 1 ml pipette to transfer drop-wise the GAG solution to the collagen suspension while bending the suspension at 15,000 rpm (setting 3.25).
  - Rate: 8 ml/min or 133 ul/sec (add one drop from a 1 ml pipette per second).
  - Need to add GAG slowly to avoid the formation of a massive collagen-GAG precipitate.
- Blend the collagen-GAG suspension $t_b/2$= 6.25 min at 15,000 rpm (setting 3.25).
  - The addition of GAG decreases the viscosity of the suspension, which now heats up less. The suspension temperature should therefore remain below 10°C as long as the beaker is inside iced water.

Part III: Degassing and Storage

- Transfer the suspension from the beaker into an ice-cold 500 ml Erlenmeyer flask using a serological pipette
  - Use a flask whose volume is 10 times the suspension volume.
- Seal the flask top and place flask inside a bucket that contains iced water. Keep the flask inside the iced water during degassing.
- Apply 15 Pa vacuum for 90 minutes.
- Use a serological pipette to transport the suspension in a sterile 50 ml conical tube.

- Transfer the suspension into clean molds for freeze-drying, or store the suspension at 4°C.
    - Use the suspension within 1 month.

F.2. Collagen-GAG Scaffold Fabrication

Supplies

- Type I collagen-glycosaminoglycan suspension (67.25 ml/sheet)
- 0.05 M acetic acid

Equipment

- 5" x 5", 18 gauge 304 stainless steel pan (VirTis Inc., Gardiner, NY)
- Genesis freeze dryer (VirTis)

Procedure

- Prepare the freeze-dryer
    - Remove drain tubing from condenser drain.
    - Make sure both windows can be tightly closed
- Clean the stainless steel pan with 0.05 M acetic acid and wipe its inside with Kim-Wipes to remove all dust and any remaining CG content from previous runs. Allow the pan to air dry.
    - Do not touch the inside of the pan with your bare hands. Use gloves.
- Pipet 10 ml of degassed suspension into each quarter of the pan.
- Clean the shelves with 70% ethanol. Place the pan on the shelves.
    - Make sure there is good contact between the pan and the shelf.
- Close tightly the chamber door.
- Choose the "freeze-dry screen" tab and either enter the freeze-drying program, or click the "load recipe" button and load a freeze-drying program. The program utilized in this thesis is shown in Fig F2.1.
    - This freeze-drying program is based on "program 2" of [Harley 2006] that fabricates scaffolds of approximately 100 μm mean pore diameter.
- Click "start cycle" and then "yes" to get freeze-drying starting.
- When secondary drying is done, click "cancel current cycle" and then "yes" to stop the freeze-drying run.
- Click the "semi-auto functions" tab and turn on the "pressure release" switch. When the chamber pressure reaches atmospheric pressure, open the chamber door and transfer the pan inside a bio-hood. Use sterile forces to transfer the dry scaffold sheet inside aluminum foil packets. Either store them in a dessicator, or cross-link them by DHT.
- In the "semi-auto functions" tab, turn off the "pressure release" switch, and then turn on the "defrost" switch to thaw the ice trapped in the condenser. Connect the drain tube in the condenser drain.
    - Defrosting lasts usually 15 min. Defrosting stops automatically within 1h after its start.

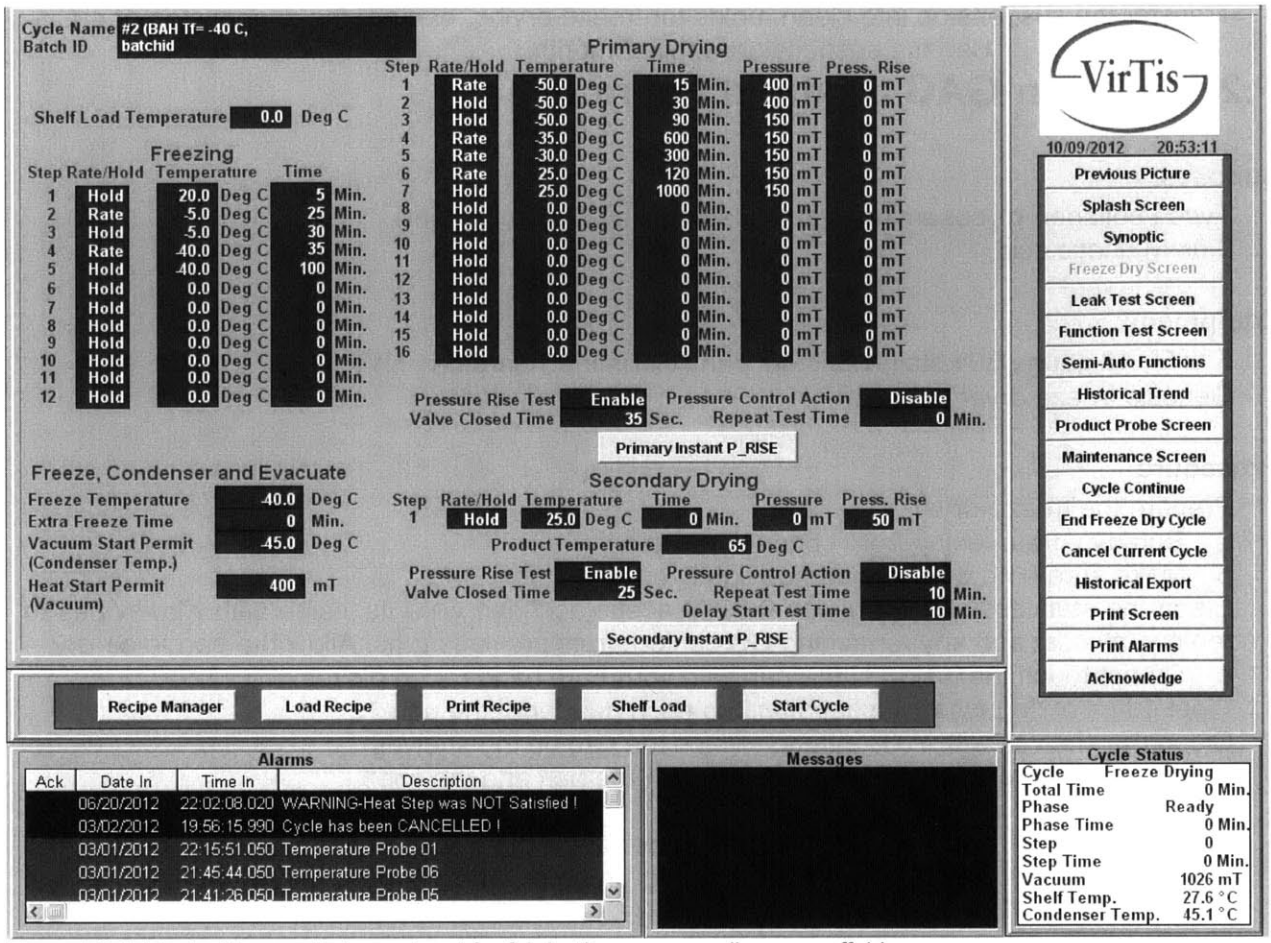


Figure F2.1: Typical freeze-drying protocol for fabricating porous collagen scaffolds.

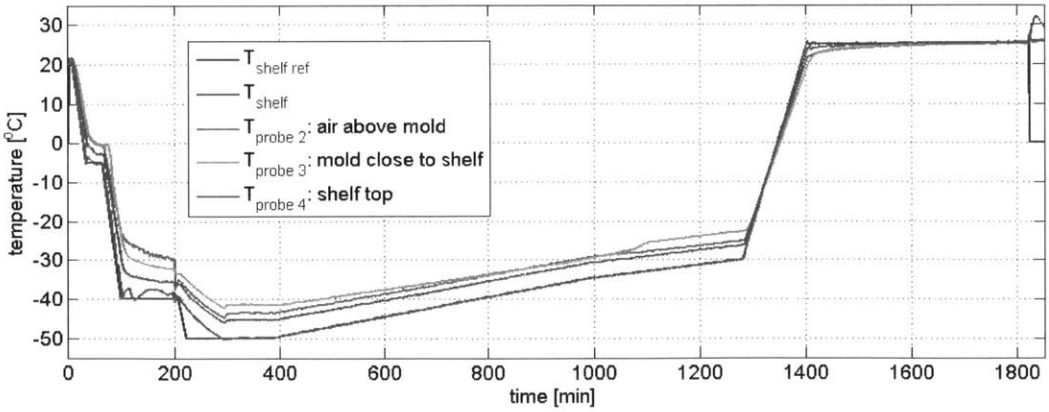


Figure F2.2: Temperature profiles of the shelf (reference and actual), the air above the pan, the stainless steel mold close to the shelf, and the shelf top surface. This response was measured when using the protocol described in Fig. F2.1. The steps of the process (equilibration, freezing, annealing, primary drying, secondary drying) can be distinguished based on the response of temperature at different locations.

F.3. Dehydrothermal Cross-linking of Collagen Scaffolds

Supplies

- 1-ethyl-3-(3-dimethylaminopropyl) carbodiimide (EDAC, Cat. No. E-7750, Sigma-Aldrich Chemical Co., St. Louis, MO); store in a dessicator at -20oC.
- N-hydroxysuccinimide (NHS, Cat. No. H-7377, Sigma-Aldrich); store in a dessicator at room temperature.
- Polystyrene cell culture 6-well plate (BD Falcon Multiwell Flat-Bottom Plates, Cat. No. 62406-161, VWR Scientific, Inc., Bridgeport, NJ)
- Sterile Phosphate Buffered Saline (PBS, Cat. No. 20012-043, Invitrogen, Inc., Carlsbad, CA)

Equipment

- Isotemp vacuum oven (Model 201, Fisher Scientific, Hanover Park, IL).

Procedure

- 1-2 hours before starting DHT, verify that the oven temperature is correct. Otherwise set to desired temperature.
- Immediately after freeze drying, place porous collagen scaffold sheets inside clean aluminum packets. Leave packet open.
- Place the packets inside the vacuum oven.
- Quickly close and tighten oven door, turn off the "vent" switch, turn on the "vacuum" switch, and turn on the vacuum pump.
- After the appropriate DHT treatment duration, turn off the vacuum pump, turn off the "vacuum" switch, turn on the "vent" switch.
- Wait until the pressure inside the oven becomes equal to the atmospheric. Then quickly open the oven door, close the aluminum packets, and store them in a dessicator.

F.4. Scaffold Cross-linking by EDAC-NHS

Based on [Harley 2006]

Supplies

- 1-ethyl-3-(3-dimethylaminopropyl) carbodiimide (EDAC, E-7750, Sigma-Aldrich Chemical Co., St. Louis, MO); store in a dessicator at -20oC.
- N-hydroxysuccinimide (NHS, H-7377, Sigma-Aldrich); store in a dessicator at room temperature.
- Sterile Phosphate Buffered Saline (PBS, 20012-043, Invitrogen, Inc., Carlsbad, CA)

Equipment

- 5, 6, or 8 mm diameter dermal biopsy punch (52441, 52442, 52443, Miltex)

Procedure

- Cut the scaffold samples to be crosslinked using a 5, 6, or 8 mm dermal biopsy punch.
- Estimate (assume 0.5% mass fraction) the dry mass of the scaffold samples $M_{scaffold}$ to be crosslinked as

  $$M_{scaffold} = f \cdot \rho \cdot V_{scaffold}$$

  Where f=0.005 is the mass fraction, $\rho \approx 1$mg/µl is water density, and $V_{scaffold}$ is the scaffold volume. For a 0.5% 8mm-diameter 3mm-thick scaffold sample, $M_{scaffold} \approx 377$µg
- Calculate the required amount of EDAC, and NHS reagent to cross-link each scaffold. Calculations assume 1.2 mmol of carboxylic acid (COOH) groups per gram of collagen [Olde Damink et al. 1996], and 5:2:1 EDAC:NHS:COOH reaction ratio (all units are in mg):

  $$M_{EDAC} = 5 \cdot M_{scaffold} \cdot 0.0012 \cdot 191.7 = 1.15 \cdot M_{scaffold}$$
  $$M_{NHS} = 2 \cdot M_{scaffold} \cdot 0.0012 \cdot 116 = 0.2784 \cdot M_{scaffold}$$
- Determine required EDAC-NHS reaction volume $V_r = 410 \cdot M_{scaffold}$, where $M_{scaffold}$ is in mg and V_r in µl.
- Prepare $V_r/4$ µl 4x working solution of EDAC in H_2O (57.6 mM), and $V_r/4$ µl 4x working solution of NHS in H_2O (22.2 mM) for each scaffold.
  - Thaw EDC to room temperature first.
  - Prepare the EDC and NHS solutions fresh!
- Transfer scaffolds in sterile clean PCR tubes (2 scaffolds per tube)
  - Hydrate the scaffolds at 4°C in $V_r/2$ sterile water for 15 minutes.
- Prepare 2x EDAC-NHS working solution by mixing 4x working solutions of EDAC and NHS.
- Add $V_r/2$ of 2x EDAC-NHS working solution per scaffold, mix using a pipette, and incubate 1h at room temperature.
- Use a pipette to remove the EDAC-NHS working solution. Dispose by EHS standards.
- Add V_r quench solution (0.1M ethanolamine or 50 mM Tris HCl pH 7.4) per scaffold. Incubate 30 min at room temperature.
- Use a pipette to remove the quench solution.
- Wash scaffolds with excess PBS at 4°C. 3 times for 30 min each and one overnight.
- Store scaffolds in PBS at 4°C until use.

Appendix G: Cell Culture Protocols

Contents

G.1. Mammalian Cell Culture on Flasks

Cells are grown in cell culture flasks or dish. As cells proliferate cell density increases and if left untreated the culture becomes confluent (cells neighbor each other). It is usually necessary to detach cells from the flask and transport them into new flasks at lower cell density before the culture reaches confluency. This protocol describes how to detach cells from cell culture flasks and transfer them into new sterile culture flasks.

Supplies

- PBS, without calcium and magnesium, sterile (Gibco 14190-1440), stored at 4°C.
- DMEM++ cell culture medium (DMEM plus 10% FBS, 1% penicillin/streptomycin)
  - DMEM, high glucose, no sodium pyruvate (Invitrogen 11965092)
  - Fetal bovine serum (Invitrogen 16000036), stored at -20°C until added to DMEM.
  - Penicilin/streptomycin (Invitrogen 15140148), stored at -20°C until just before added to DMEM.
  - Prepare 555ml DMEM by adding one 50ml shot of FBS and 5ml penicillin/streptomycin to a 500ml bottle of DMEM inside a bio-hood.
- Trypsin 0.05% with 0.02% EDTA (Gibco 25300-054), thaw bottle, make 10ml aliquots, store at -20°C long term, thaw one aliquot and store at 4°C for a few days before use.
- Pasteur pipettes, sterilized (VWR 14672-380).
- Serological pipettes, 5 mL, 10 mL (VWR).
- Pipettes and pipette tips, 200 µL, 1 mL.
- T25 or T75 Cell culture flasks with vented cap (BD 353108 , BD 353136).
- Trypan blue (Gibco 15250-061).

Equipment

- Water bath (Yannas Lab 3-315, So Lab NE47-235)
- Biological Hood (Yannas Lab 3-315, So Lab NE47-205)
- Cell incubator connected to CO2 tanks (Yannas Lab 3-315, So Lab NE47-205)
- Upright light microscope (Yannas Lab 3-315, So Lab NE47-216)
- Hemocytometer (Yannas Lab 3-315, So Lab NE47-205)
- Table centrifuge (Yannas Lab 3-315, So Lab NE47-216)

Safety Precautions

- Keep the cell incubator clean:
  - Always spray hands with 70% ethanol before opening the cell incubator.
  - Wipe any items with 70% ethanol before transfer inside the incubator.
  - Open the incubator door gently, for as small angle as possible, and as quickly possible.
  - Each user needs to keep his/her cells together in a separate part of the incubator.
- Keep the bio-hood clean:
  - Always use gloves in the bio-hood.
  - Wipe working area with 70% ethanol before and after use.
  - Spray gloved hands with 70% ethanol before inserting hands in hood.
  - Transfer the minimum number of items necessary in the hood. Always spray items to be transferred inside the bio-hood with 70% ethanol. Avoid blocking the air-flow.
  - Clean any spills immediately (especially ones involving cell culture medium).
- Check that the cell incubator works properly:
  - CO2 supply: Check the two CO2 tanks periodically.

- Humidity: there is sterile water on a pan located in bottom shelf of the incubator. This water keeps the incubator humidified. Need to add 1L of autoclaved sterile water per month.

- Keep cells in a sterile environment:
  - Do not open cell culture flasks outside the biohood.
  - Do not open sterile solutions outside the bio-hood.
  - If growth is noticed in a cell culture, bleach and dispose the flask. Do not place back in incubator or biohood,
- Contamination
  - If the pH of the culture medium in a cell culture flask gets acidic (pH indicator color changes from red to yellow) this is a sign of contamination.
  - If cell culture medium looks turbid, this is a sign of contamination.
  - Bleach and dispose contaminated flasks immediately. Do not place back in biohood or incubator. If contamination persists then dispose all cells, disinfect biohood & incubator and start a new cell culture.
- When handling liquids & cell suspensions:
  - Do not touch surfaces that contain cells with liquid-handling instruments (tips, pasteur pipettes, serological pipettes) from touching when the same instrument will come in contact with a liquid that should remain sterile.
  - Minimize contact between liquid-handling instruments (pipettes or pasteur pipettes) and glass-ware, flasks, bottles.
  - Minimize the exposure of sterile liquids to air inside the bio-hood.
  - Minimize the time where your hands are located above open bottles inside the bio-hood.

Procedure

Preparation

- Prepare the bio-hood: switch light from "UV" to "light", wipe surface with 70% ethanol.
- Open the Pasteur pipette metal box, remove one Pasteur pipette (without touching the others), and place the Pasteur pipette on the vacuum tube.
- Warm PBS, trypsin/EDTA and cell medium aliquots in a water bath at 37°C, 10 min.
  - Make sure there is enough amount of sterile PBS, trypsin/EDTA and cell culture medium.
- Transfer the culture flasks (ideally one representative) to be splitted from the cell incubator on the microscope stage. Image the culture flask and check the confluency of the cell culture.
  - Confluency: the percentage of the culture flask area that is covered by cells.
- Proceed in cell splitting when the cell culture flasks are approximately 70-80% confluent.

Detach cells

- Transfer cell culture flasks from the cell incubator into the bio-hood.
- Aspirate the cell medium in the cell culture flasks using the Pasteur pipette connected to the vacuum pump.
- Use a serological pipette to add V_{PBS} ml PBS to each flask (Table G1-1). Roll gently. This wash removes residual cell culture medium that impedes trypsin/EDTA activity.
- Aspirate PBS using a Pasteur pipette.
- Use a serological pipette to add $V_{trypsin}$ ml trypsin/EDTA to each flask (Table G1-1).
- Incubate flasks for 5 minutes inside the 37°C cell incubator.
- Hit gently the flasks on the side to detach cells from the flask surface.
- Image the flasks in the microscope to check that cells have indeed detached from the flask.
  - Detached cells appear like spheres floating in the medium.

- If cells are not detached incubate 5 min more in the cell incubator, or add a bit more trypsin.
- Use a serological pipette to add $V_{medneutr}$ ml cell culture medium to each flask, see Table G1-1. This step neutralize trypsin/EDTA action. Extensive trypsin action can kill cells.
- Use a serological pipette to aspirate the detached cells from the flasks.
  - Aspirate and wash the flask surface 8 times, avoid creating bubbles.
  - Aspirate and respirate the cell suspension 8 times, avoid creating bubbles.
  - Aspirate the cell suspension and transport in 15 mL or 50 ml conical tubes.
  - Repeat procedure for all flasks.

	V_{PBS} [mL]	$V_{trypsin}$ [mL]	$V_{medneutr}$ mL]	$V_{medflask}$ [mL]
T25 flasks	1.5	0.5	1	3
T75 flasks	3	1.5	3	5

Table G1-7-1: Parameters for cell detachment.

Wash & count cells

- Centrifuge conical tubes that containing the cell suspension at 560×g (1700 rpm), 15 min, 25°C.
- During centrifugation prepare the flasks of the new culture passage:
  - Calculate number of flasks of new culture. For fibroblasts and smooth muscle cells, usually each flask can provide 2-3 new flasks.
  - Label the new flasks with date, cell type, passage number, and user name. The "passage" number is a indicator how many times a particular cell culture has been split. Every split increases passage number by 1.
  - Add $V_{medflask}$ cell culture medium to each flask (Table 1).
  - Wipe the flasks with 70% ethanol and transfer them in the cell incubator.
  - Dispose the flasks of the previous cell culture: add 10% bleach, incubate 20 minutes, dispose bleached medium in the sink, dispose used flasks as biohazard trash.
- When centrifugation is done, transfer the conical tubes in the biohood, and use a Pasteur pipette to aspirate the supernatant without disturbing the cell pellet.
- Use a 1ml pipettor to add 1 mL PBS to each centrifuge tube. Use the same pipette to break the cell pellet so that cells resuspend in PBS.
- Collect all cell suspensions (same cell type, passage number) into a single conical tube. Add 2-5 ml more PBS. Make sure to keep record of total volume of PBS $V_{cellPBS}$ added to cells.
- Mix the cell suspension well (use 1 ml pipettor) and then quickly use a 200ul pipettor to transfer a 40 µL aliquot of cell suspension into a PCR tube. This aliquote will be used for cell counting.
- Centrifuge the conical tube at 560×g (1700 rpm), 15 min, 25°C.
- During centrifugation, use the 40 µL cell suspension aliquot for cell counting:
  - If it is desired to estimate fraction of live cells, add 20 uL trypan blue dye to the cell aliquot & mix the suspension.
  - Clean the hemocytometer surfaces using 70% ethanol.
  - Transfer 20 µL of cell suspension into the two parts of the hemocytometer.
  - Count the total number of cells in the quadrants of the hemocytometer. Calculate the mean number M of cells counted on each hemocytometer quadrant.
  - Total number of cells N_{cells} in the cell suspension

    $$N_{cells} = d \cdot M \cdot V_{cellPBS} \cdot 10^4$$

    where d is the cell dilution. d=1 if no trypan blue is added, otherwise it is d=6/4 = 1.5
  - Each 100% confluent T75 flask provides 1.5-2 million fibroblasts. Each 100% confluent T25 flask provides 0.5-0.7 million fibroblasts.

- When centrifugation is done, transfer the conical tubes in the biohood, and use a Pasteur pipette to aspirate the supernatant, avoid disturbing the cell pellet.
- Use a 1ml pipettor to transfer N_{cells} mL cell culture medium to each pellet, and use the same pipettor to break the pellet. This results in 1000 cells/ul cell suspension.
  - E.g. if the cell pellet contains N_{cells} = 1E6 cells, add 1 ml medium.

Create new cell culture

- Transfer the flasks of the new culture from the incubator into the bio-hood.
- Use a pipette to transfer a proper amount of 1000 cells/ul cell suspension to each flask. The proper number of cells depends on the cell line. For fibroblasts, transport 250E3 cells to a T25 flask and 750E3 cells to a T75 flask.
- Spray the flasks with 70% ethanol and transfer back to the incubator.

Clean the bio-hood

- Aspirate 5-10 ml 10% bleach using the Pasteur pipette. Dispose the Pasteur pipette.
  - This will clean the vacuum tube and prevent growth.
- Cover the vacuum tube with aluminum foil.
- Wipe the cabinet with 70% ethanol, switch light from "light" to "UV", and close the sash.

G.2. Cryo-Preservation of Cell Suspensions

Supplies

- Recovery™ Cell Culture Freezing Medium (Invitrogen #12648-010)
- Cryovials (VWR #16001-114)
- PBS, without calcium and magnesium (Gibco #14190-1440)
- Trypsin 0.05% with EDTA (Gibco #25300-054)
- Cell culture medium (DMEM++): DMEM + 10% FCS +1% penicillin streptomysin
    - DMEM medium, high glucose, no sodium pyruvate (Gibco #11965-092)
    - Fetal Calf serum (Gibco #12483-012)
    - Penicilin and streptomycin (Gibco #15070-063)
- Pasteur pipettes, sterilized (VWR)
- Serologic pipettes, 5 mL, 10 mL (VWR)
- Pipettes and pipette tips, 200 µL, 1 mL (VWR)
- 15 mL centrifuge tubes (VWR)
- 200 µL eppendorf tubes (VWR)

Equipment

- Biological hood
- Warm water bath
- 4°C refrigerator, -20°C freezer, -80 °C freezer
- Liquid nitrogen storage tank and liquid nitrogen supply.

Procedure

This protocol is used to store cells in cryovials by cryo-preservation. Cells are suspended in a proper buffer, then frozen and stored in liquid nitrogen.

Preparation

- Thaw one aliquote of Recovery™ Cell Culture Freezing Medium and store at 4°C.
- Label cryovials (the cells of a 90% confluent T75 flask can provide 2-3 cryovials) with date, cell type, passage number, and usename.
    - KEEP CRYOVIALS STERILE AT ALL COSTS!!!

Detach cells from the cell culture flasks

- Use a Pasteur pipette to aspirate the cell culture medium from the flasks.
- Use a serological pipette to add α mL PBS to the flasks (Table G2-1). Shake gently.
- Aspirate PBS using a Pasteur pipette.
- Use a serological pipette to add β mL trypsin/EDTA to the flasks (Table G2-1).
- Incubate flasks for 3 minutes inside the 37°C incubator.
- Carefully hit the flasks on the side in order to detach cells. Check that cells have detached from the flask/disk in the light microscope. If cells are not detached, wait and add a little more trypsin/EDTA or repeat gentle hits. When cells detach then proceed.
- Use a serological pipette to add γ mL DMEM++ to the flasks in order to neutralize trypsin/EDTA (Table G2-1).
- Use a serologic pipette to aspirate and spray the cell suspension in each flask in order to wash detached cells from the flask surface. Repeat this 8-10 times, being careful not to create bubbles. Use the same pipette to aspirate and respire the cell culture from the

bottom corner of the flask. Repeat this 8-10 times, being careful not to create bubbles. Aspirate the cell suspension and transport into a 15 mL centrifuge tube.
    - Repeat this step for all flasks.
- Centrifuge the 15 mL centrifuge tubes (containing the cell suspension) at 1700 rpm, 15 min, 25 deg.
- Use a Pasteur pipette to aspirate supernatant, being careful not to aspirate the cell pellet.

Cryo-preservation of the cell suspension

- Resuspend cells in Recovery™ Cell Culture Freezing Medium (stored at 4°C).
    - Add 1 mL Cell Culture Freezing Medium per 1 million of cells .
    - Break pellet by pipetting gently.
- Transfer 1 mL cell suspension per cryovial.
- Store the cryovials at 4°C for 2 hours.
- Store the cryovials at -20 °C for 2 hours.
- Store the cryovials at -80 °C for 2 hours.
- Store the cryovials in liquid nitrogen (77 K) for long-term storage.

	α [mL PBS]	β [mL Trypsin/ EDTA]	γ [mL DMEM++]
T25 flasks	2	0.5	1
T75 flasks	4	1.5	3

Table G2-7-2: Culture parameters for cell cryo-preservation.

G.3. Thawing Cryo-Preserved Cells and Initiating a New Cell Culture

Supplies

- Cell suspension stored inside cryovials in a tank filled with liquid nitrogen. See
- DMEM++ cell culture medium for Fibroblasts (DMEM + 10% FCS + 1% penicillin/streptom)
  - DMEM medium, high glucose, no sodium pyruvate (Gibco #11965-092)
  - Fetal Bovine serum
  - Penicilin and streptomycin (Gibco #15070-063)
- Serologic pipettes, 5 mL, 10 mL (VWR #)
- Pipettes and pipette tips, 200 µL, 1 mL (VWR #)

Equipment

- Biological hood
- Warm water bath

Procedure

Cells are preserved over long time periods inside cryovials, stored inside liquid nitrogen. This protocol describes how to start a new cell culture from one of these cryovials.

Prepare cell culture flasks

- Turn on the warm water bath. Set the water bath temperature at 37 deg. Use the water bath to warm one bottle containing cell culture medium.
- Each cryovial is used to seed two or three T25 cell culture flasks.
- Label two or three sterile flasks with cell type, passage number, user name, and date.
- Add 3 mL cell medium to each flask.

Initiate the cell culture

- Remove one cryovial of proper cell type and passage number from the liquid nitrogen tank.
  - Use heat-protection gloves to avoid frost-bites that happen when skin comes in contact with liquid nitrogen.
- Spray the cryovial with 70% ethanol, quickly transfer it inside a bio-hood, unscrew the vial cup (to relieve pressure), and re-tighten the cryovial screw.
- Place the vial inside the warm water bath (set at 37 deg) for 30-60 sec.
  - Use a foam floater to prevent the water from getting into contact with the vial cap screw.
  - At the end of this short incubation the cell suspension is thawed.
- Use a pipette to mix and transport the thawed cell suspension into the culture flasks.
- Swirl gently so that cells are distributed uniformly in the medium inside the flasks.
- CHANGE ALL THE MEDIA AFTER THREE HOURS OR ON THE FOLLOWING DAY THE LATEST.

G.4. Seeding Cells in Porous Collagen-Based Scaffolds

This protocol describes the seeding of eukaryotic cells inside porous collagen-based scaffolds inside agarose-coated 6-well plates.

Supplies

- 3mm-thick sheet of porous collagen-based scaffold, sterile, fabricated by freeze-drying, cross-linked by DHT, stored desiccated inside aluminum foil in room temperature.
- 6-well polystyrene plate (BD Falcon #353046).
- 0.2% agarose solution.
- PBS, sterile (Gibco #14190-144), stored at 4°C.
- DMEM++ Cell culture medium, stored at 4°C (DMEM, 10% FBS, 1% penicillin streptomycin)
  - DMEM medium, high glucose, no sodium pyruvate (Invitrogen #11965092)
  - Fetal bovine serum (Invitrogen #16000036)
  - Penicilin and streptomycin (Invitrogen #15140148)
- Biopsy punch, 8mm diameter (Miltex #33-37) or 5mm diameter (miltex #33-35).

Equipment

- Biological Hood (Yannas Lab, 3-315).
- Forceps (sterile) (Yannas Lab, 3-315).
- Cell incubator (Yannas Lab, 3-315).
- Mini-centrifuge, eppendorf 5415C (Yannas Lab, 3-315).

Procedure

Safety Precautions

Check the safety precautions mentioned in the "Mammalian Cell Culture on Flasks" protocol.

Coat the wells of a 6-well plate with agarose

- Take one 6-well plate out of its sterile cover and place it quickly inside the hood
- Add 800 uL 0.2% agarose on each well.
- Shake the plate mildly so the agarose solution is distributed evenly on the wells.
- Incubate the 6-well plate inside the bio-hood overnight (at least 8 hours).
  - The purpose of this incubation is to dry the agarose solution, remove the water and let the agarose coating on the wells.
  - Do not place the cover on the 6-well plate. The cover will slow down air-drying.
- The wells of the plate now are covered with agarose. Agarose coating prevents cells from attaching on the plate and favors their attachment on the scaffold.

Prepare cylindrical scaffold samples

- Transfer the collagen scaffold sheet (inside its aluminum foil cover) from the desiccator into the bio-hood.
- Use a biopsy punch to cut 8mm-diameter samples from scaffold.
  - If necessary store the samples in petri dish, store in a desiccator.
- Transfer the collagen scaffold sheet back to the desiccator.

Prepare the cell suspension

- Detach cells from a flask, wash in PBS, and resuspend them in cell culture medium to a final concentration of 1000 cells/ul (see "cell splitting protocol")

- Need approximately 50000 cells for each 8mm –diameter scaffold and 30000 cells for each 5mm-diameter scaffold.

Seed cells into the scaffold

- Use a 200 uL pipette to place a drop of cell suspension an agarose-coated well.
  - The drop size is 50 ul for 8mm scaffolds and 30 ul for 5mm scaffolds.
  - Add 1-3 drops per well, the drops should be far away or else they will unify.
- Use sterile forceps to place a scaffold just above the drop. The "pan side" (least shiny) of the scaffold should be facing the drop.
- As the scaffold and the cell suspension drop get slowly into contact, the cell suspension is aspirated inside the scaffold due to capillary action. Quickly release the scaffold from the forceps.
- Within 30 sec most of the cell suspension drop will be aspirated inside the scaffold.
- Repeat the seeding procedure for all scaffolds to be seeded.

Scaffold incubation

- Place the cover of the 6-well plate on the plate, wipe with 70% ethanol and transfer into the cell incubator.
- Incubate the plate 1h at 37°C, 5% CO_2.
  - This provides enough time for cells to bind to the scaffold.
- After the 1h incubation, transfer the 6-well plate in the bio-hood.
- Add 1 mL warmed (37°C) cell culture medium per seeded scaffold in each well.
- Incubate the cell-seeded scaffolds at 37°C, 5% CO_2. Change medium every 3 days.

G.5. Extract Cells from Porous Collagen Scaffolds Using Collagenase

Supplies

- Collagenase I, 200 Units/mg, stored at 4 deg. (Worthington # LS004196)
- Tris-HCl buffer, 1 M, pH 7.0, 1L (Boston Bioproducts # BM 310)
- Eppendorf tubes (VWR)
- Pasteur pipette, sterile
- NaCl
- CaCl2 (Boston Bioproducts #BM-625)
- PBS, sterile (Gibco #14190-144)
- DMEM++ cell culture medium for fibroblasts (Invitrogen #11965 plus 10% FBS plus 1% penicillin/streptomycin)

Equipment

- Biological Hood (Yannas Lab, 3-315)
- Forceps (sterile) (Yannas Lab, 3-315)
- Mini-centrifuge, eppendorf 5415C (Yannas Lab, 3-315)

Procedure

Prepare Collagenase Stock Buffer

- Prepare "Tris-HCl buffer": 50 mM Tris-HCl pH 7, 200 mM NaCl, 5 mM $CaCl_2$.
- Prepare collagenase stock solution by dissolving 2 mg of collagenase I per mL of Tris-HCl buffer (resulting in 0.4 U/μL). Aliquote and store at -20 deg.

Prepare Working Enzymatic Solution

- Thaw one aliquote of collagenase stock buffer.
- For each 8 mm seeded scaffold to be digested, prepare 80 μL of "working enzymatic solution" inside an eppendorf tube by mixing 40 μL collagenase stock solution with 40 μL of cell culture medium. Each eppendorf tube contains 16 U of collagenase I.

Scaffold Digestion

- Get the seeded scaffolds (inside the wells of a 6-well plate) out of the 37 deg incubator, place them inside the bio-hood.
- Aspirate the medium from each well of the 6-well plate.
- Add 3 mL PBS to each well that contains a seeded CGS, shake gently for 20 seconds, wait three minture, aspirate PBS. Repeat wash twice.
- Use forceps to transport each scaffold from its own well in the 6-well plate into a separate eppendorf tube containing 80 μL collagenase working solution. Avoid closing the cap tightly. Cover with aluminum foil (to protect from the UV light).
- Incubate inside the hood at 37 deg for 2 hours.

Wash cells

- Mini-centrifuge (14000 rpm, 10 sec). Aspirate supernatant. Add 200 μL PBS, break pellet, mini-centrifuge (14000 rpm, 10 sec), aspirate PBS.
- Add a proper amount of cell culture medium (DMEM++ for fibroblastS), break pellet. The cell suspension.is ready for further use.

G.6. Cell Lysis

Jin Guo, Dimitrios Tzeranis

Supplies

- 2× RIPA: Boston Bioproducts BP-115, NE47-235, 4C refrigerator.
- 0.5 M EDTA: Boston Bioproducts BM-150, NE47-235, NE47-235, 4C refrigerator.
- 500 mM NaF, NE biolabs #P0759S, stored aliquoted at NE47-214, -20 C freezer
- 100 mM NaVO3, NE biolabs #P0758S, stored aliquoted at NE47-214, -20C freezer
- Halt 100× protease inhibitor cocktail, Pierce #78430, stored at NE47-235, 4C refrigerator.

Procedure

- Prepare lysis buffer (50mM Tris Hcl pH 7, 150mM NaCl, 5mM EDTA, 0.5% NP40, 10ul/ml protease inhibitors, 2mM NaVO3, 10mM NaF)
  - Need 25 µl per 5mm-diameter collagen scaffold, 100 µl per well of a 24-well plate.
  - Add 100 parts 2×RIPA, 88 parts dH2O, 2 parts 0.5M EDTA, 2 parts protease inhibitor, 4 parts 0.1M NaV03, 4 parts 0.5M NaF.
- Place each cell-seeded scaffold in a PCR tube, 1 CGS per well.
- Add 200uL PBS to each scaffold, incubate at room temperature for 5 minutes. Use a pipette to discard PBS. Repeat PBS wash 2 times more.
  - Make sure no residual PBS remains in tube walls after washes.
- Add 25ul lysis buffer per scaffold.
- Incubate 1.5 hour at 4°C while fast rotating
- Microcentrifuge the tubes for 1 min at 13000 rpm
- Aspirate and store supernatant at -20°C until downstream use.

Appendix H: Bacteria Culture Protocols

Contents

H.1. Ampicillin Stock Solution Recipe

This protocol describes the preparation of 2.65 mg/ml ampicillin stock solution.

Supplies

- Distilled deionized water ddH2O (Cellgro #)
- Ampicillin (Sigma #A9393 or #A9815; A9815 has much better solubility 50 mg/ml in water), stored at 4°C.
- Sterile 0.22 um filter (Pall corporation #)

Equipment

- Magnetic stirrer (Yannas Lab, 3-315)
- Bio-hood (Yannas Lab, 3-315)
- 4°C cold box (Yannas Lab, 3-315)

Procedure

- Identify necessary volume V of ampicillin solution

Culture size [ml]	80	100	400	500	600	1000
Necessary mass of ampicillin [mg]	4	5	20	25	30	50
Ampicillin 50 mg/ml stock (A9815) [ml]	0.08	0.1	0.4	0.5	0.6	1.0

- Preparation (inside a glass beaker):
    - A9393: Add 2.65V mg A9393ampicillin to V ml distilled deionized water.
    - A9815: Add 50V mg A9815 ampicillin to V ml distilled deionized water.
- Use a magnetic stirrer to dissolve ampicillin in water inside a 4°C cold box (takes ≥15 min).
- Filter-sterilize ampicillin solution inside a bio-hood using a 0.22 um filter.
- Divide the sterile solution into aliquots and store at -20°C.

H.2. LB-Agar-Ampicillin Plates Recipe

Supplies

- LB-Agar powder (USB #75851-250, VWR #101172-028)
- Sterile ampicillin stock solution (see "ampicillin stock solution recipe" protocol), 50 mg/ml, stored at -20°C.
- 100 mm glass Petri Dish or 100 mm single use Petri dish.
- Serological pipette (25 ml)

Equipment

- Autoclave (Yannas Lab, 3-315)
- Water bath (Yannas Lab, 3-315)
- Heat magnetic stirrer (Yannas Lab, 3-315)
- 4°C refrigerator (Yannas Lab, 3-315)

Procedure

- The following protocol refers to the coating of six to eight 100mm Petri dish. Use either glass of plastic Petri dish, each consists of a dish and a cover.

Prepare LB-agar solution

- Add 7 g LB-agar powder to 200 ml deionized water in a 250 ml bottle.
- Stir and heat until LB-Agar media is dissolved.
- Autoclave the LB-agar solution (liquid cycle) for 20 min.
- When the autoclave temperature cools to approximately 180F (82°C), transfer the beak in a water bath set at approximately 45°C.

Prepare the LB-agar-ampicillin solution

- Thaw one aliquot of 50mg/ml ampicillin stock solution (need 200ul).
- When the LB solution temperature stabilizes around 45°C, add 200 µl ampicillin stock solution, stir mildly to mix the solution and keep the beaker inside the 45°C water bath.
    - The final ampicillin concentration in the LB-agar-ampicillin solution is 50 µg/ml.

Coat the dish

- Label eight 100mm petri dish with name, date and content.
- Use a 25 ml serological pipette to add 24 ml LB-agar-ampicillin solution to each dish in a single shot.
    - Use the serological pipette to remove any bubbles formed on the plate.
- Let the plates cool in room temperature. Place the cover on top of the dish at some angle in order to prevent the condensated water from falling on the dish (lid covers the dish fully).
- Incubate the LB-agar-ampicillin coated plates at room temperature until the LB-agar-amp medium has solidified (takes approximately 30 min)
- Seal the dish/cover with parafilm, store them in a 4°C refrigerator upside-down.
- Use the petri dish within two weeks.

H.3. LB & LB-Ampicillin Medium Recipe

Supplies

- Luria Broth powder (USB #75854-250, VWR #101172-030).
- Glass bottles (VWR) or glass flasks.
- Sterile ampicillin stock solution (see "ampicillin stock solution recipe" protocol), 50 mg/ml, stored at -20°C .
- Serological pipettes (5 ml, 10 ml, 25 ml)

Equipment

- Autoclave (Yannas Lab, 3-315)
- Water bath (Yannas Lab, 3-315)
- Heat magnetic stirrer (Yannas Lab, 3-315)
- 4°C refrigerator (Yannas Lab, 3-315)

Procedure

- This protocol describes the preparation of LB or LB-amp medium inside glass flasks.
- Every time one opens a bottle containing LB medium, it is a good practice to:
  - minimize the exposure of the open bottle
  - flame the bottle lips using a Bunsen burner in the beginning and the end of the process.

Prepare LB medium

- Add V ml deionized water and LB powder inside a bottle or flask of proper size

Culture size [ml]	80	100	400	600	1000
LB powder [g]	1.6	2	8	12	20
Ampicillin stock 50 mg/ml [µl]	80	100	400	600	1000
Flask size [ml]	500	500	2000	2000	6000

- Steer the solution until the powder is completely dissolved in the water.
- Autoclave the solution (liquid cycle) for 20 min (make sure the cap of the flask is loose).
- To prepare LB medium, wait until the flask/bottle cools down to approximately 45°C, tighten bottle cap and store the sterile LB in 4°C.
- To prepare LB-ampicillin, proceed to next steps.

Prepare LB-ampicillin medium

- When the temperature of the autoclave cools to approximately 180F (82°C), transfer the beak in a water bath set at approximately 45°C.
- Thaw one aliquot of ampicillin stock solution for each batch (see table).
- When the temperature of the LB solution has stabilized around 45°C add ampicilin aliquote to the LB medium.
  - The final ampicillin concentration in the LB-ampicillin solution is 50 µg/ml.
- Tighten the bottle cap, cover with the aluminum foil, seal with para-film, and store at 4°C.
- Use the LB and LB-amp solutions within a month.

H.4. Culture Bacteria on Petri Dish

Supplies

- Petri Dish coated with LB-Agar or LB-Agar-Ampicillin (see "LB-Agar-Ampicillin Plate Recipe" protocol).
- Bacteria stored in 10% glycerol solution inside cryovials at -80°C.
- Inoculating loops (VWR #90001-100)

Equipment

- -80°C freezer (NE 47 second floor)
- Bunsen burner (Dedon Lab, NE 47 second floor)
- Bacteria incubator 37 ° C (Dedon Lab, NE 47 second floor)

Procedure

- Get one cryovial containing bacteria out from -80°C freezer (second drawer, top lane, NE 47 second floor).
- Turn on the bunsen burner and perform the rest of the steps under its flame to enhance sterility.
- Open the tube that contains the inoculation loops and flame its entrance. Get a sterile inoculating loop out of the tube. Flame the entrance again and close the tube.
- Use the sterile inoculating loop to touch the frozen bacteria suspension in the cryovial.
    - Do not thaw bacteria! Place cryovial back to -80°C quickly after streaking.
- Touch the loop to the surface of the plate, move the loop back and forth on the first quarter of the plate, moving towards the center of the dish.
- Streak the loop in the second quarter of the dish, start by crossing the previous streak once and continue by moving the loop back and forth on the second quarter of the plate, moving towards the center of the dish.
- Repeat the previous step in the third and fourth quadrants.
- Incubate the bacteria-containing Petri dish in a 37°C incubator for 12h.
    - The next day single colonies will be formed on the plate. In the parts of the plate where the distance between neighboring colonies is large enough, single colonies can be picked using an inoculation loop for suspension culture (see "Start a Suspension Culture of a Single Bacteria Colony" protocol).
- Petri dish can be sealed using paraffin and stored at 4°C for 2 weeks before bacteria die.

* For more information about streaking bacteria on LB-agar-plates, take a look in chapter 1 of "current protocols in molecular biology" and chapter 11 of "At the bench", by K. Barker.

H.5. Inoculate a Single Bacteria Colony into a Suspension Culture

Supplies

- Bacteria streaked in a plate coated with LB-agar-ampicillin (see "Culture bacteria on LB-Agar(-Ampicillin) Plates" protocol).
- Sterile LB or LB-Ampicillin medium (see "LB & LB-Ampicillin Medium Recipe" protocol).
- Inoculating loops (VWR #90001-100)
- Round-bottom test tubes, polypropylene, 14 ml (BD falcon #352059)
- Serological pipettes (5 ml or 10 ml).

Equipment

- Bunsen burner (Dedon Lab, NE 47 second floor)
- Shaking incubator (Dedon Lab, NE 47 second floor)

Procedure

Prepare Test Tubes

- Turn on the Bunsen burner and perform the rest of the steps under its flame to enhance sterility.
- Remove the cap of a test tube, and flame its entrance.
- Unscrew the cap of the bottle that contains the LB or LB-amp medium, flame its entrance, use a serological pipette to aspirate the growth medium, flame the entrance and place cap.
  - Use LB medium to culture bacteria that do not contain ampicillin resistance genes, for example during the preparation of competent bacteria
  - Use LB-amp medium to culture or select bacteria that contain ampicillin resistance genes.
- Uncap a test tube, flame its entrance, transfer the growth medium into the test tube, flame the tube entrance and place its cap back.
  - Repeat the previous three steps for each test tube.
  - The amount of medium that is less than 5 ml for a 14 ml test tube.

Pick a Single Colony and Inoculate in a Suspension Culture

- Transfer the LB-agar or LB-agar-ampicillin plates, which contain the bacteria colonies, from the 37°C incubator on the bench, under the flame of the Bunsen burner.
- Open the tube that contains the inoculation loops and flame its entrance. Get a sterile inoculating loop out of the tube. Flame the entrance again and close the tube.
- Use the sterile inoculating loop to touch gently a colony on the Petri dish.
- Remove the cap of a test tube that contains growth medium and flame its entrance.
- Place and move the loop inside the medium quickly.
- Flame the test tube entrance and place its cap back. Dispose the inoculation loop
- Incubate the suspension culture in a 37°C shaking incubator (250 rpm) overnight.

H.6. Bacteria Suspension Culture for Recombinant Protein Expression

Supplies

- Saturated suspension cultures of single bacteria colonies in LB-ampicilin medium inside 14 ml test tubes, incubated overnight at 37°C with 250 rpm shaking (see protocol "Inoculate a Single Bacteria Colony into a Suspension Culture").
- Sterile LB-Ampicillin medium (see "LB & LB-ampicillin medium recipe" protocol).
- IPTG (Sigma #I5502), in 0.2M aqueous solution. Stored in 1 ml aliquots at -20°C.
- 500 ml centrifuge tubes for JLA 10.5 Beckman centrifuge rotors (max 13400 g, Newman Lab, 68-380, or Dedon Lab, NE 47 second floor)
- 50 ml centrifuge tubes for JA 17 Beckman centrifuge rotors
- 5ml cuvettes (So Lab)
- 5 ml serological pipettes (VWR)

Equipment

- Bunsen burner (Dedon Lab, NE 47 second floor)
- Shaking incubator (Dedon Lab, NE 47 second floor)
- Spectrophotometer (Dedon Lab, NE 47 second floor)
- Floor centrifuge (Dedon Lab, NE 47 second floor).

Procedure

Preparation

- Prepare three 600 ml sterile LB-ampicillin medium solutions inside 2L glass flasks (see "LB & LB-ampicillin medium recipe" protocol). Store at 4°C until use.
- Each 600 ml bacteria culture requires 1.2 ml 0.2M IPTG stock solution (stored in 800 ul aliquots at -20°C)
- Grow one overnight 100 ml bacteria culture at 37°C, 250 rpm shaking.

Start the bacteria culture

- Turn on the Bunsen burner and perform the rest of the steps under its flame to enhance sterility.
- Transfer aseptically 30 ml of overnight saturated culture to each one of the 600 LB-amp solutions.

Monitor Bacteria Growth

- Incubate the three 600ml bacteria suspensions in a shaking incubator at 37°C, 250 rpm.
- After some initial lag time, the bacteria will start to grow exponentially. It is desired to prevent them from growing too much before inducing the protein expression, or else the amount of produced protein will be small.
- Monitor bacteria growth by measuring the 595 nm absorbance O.D. of the bacteria suspension.
    - Use a 1 ml pipette to sample 500 ul from each bacteria culture
    - Transfer this aliquote into a cuvette.
    - Use 500 ul sterile LB-amp medium as the blank sample.
    - Measure 595nm absorbance O.D. in a spectrophotometer (Dedon Lab, NE47).

Induce protein expression

- When the bacteria 595nm O.D. Is between 0.6 and 0.8, thaw and add 1.2 ml 0.2M IPTG stock solution (2 µl IPTG stock solution per ml bacteria culture).
  - The resulting IPTG concentration in the bacteria suspension is 0.4 mM.
- Incubate the bacteria suspension for 3h in a shaking incubator at 37°C, 250 rpm.

Pellet bacteria

- After the 3h protein expression incubation, take a 100 ul aliquot from each flask.
  - This is an aliquote of a 40x OD_{595}=0.75 bacteria culture which will be used to assess the expression of the recombinant proteins.
  - Spin down the aliquot (in a 1.5 ml eppendorf tube) 6500×g, 10 min, at 4°C.
  - Store the aliquot pellet at -20°C.
- Pour the three 600 ml bacteria cultures into four 500 ml plastic centrifugation bottles.
  - Don't transfer more than 400 ml bacteria suspension per bottle because it may leak during centrifugation.
  - Add 10% bleach to the remaining bacteria suspension. Discard after 20 min.
- Centrifuge the bacteria suspension at 6500×g, 10 min, 4°C in a Beckman floor centrifuge loaded with a Beckman JLA 10.5 rotor. Make sure:
  - the proper settings are loaded in the centrifuge controller (rotor type, acceleration, time, temperature).
  - the plastic JLA bottles have approximately the same weight (±20 mg error) Use a serological pipette to add/remove bacteria suspension or PBS.
  - the plastic JLA tubes are positioned inside metal JLA containers, the JLA container metal cap is tightened, and the JLA containers are positioned properly (counter-side) in the JLA rotor.
  - the JLA rotor cap is placed and securely tightened before centrifuging [!!].
- After centrifugation, the bacteria form a pellet in the bottom of the plastic JLA bottle. See the next steps, while the supernatant contains LB medium.
  - Some medium may have leaked outside the plastic centrifugation bottles. In this case bleach the JLA containers for 20 min, and then wipe with 70% EtOH.
- Pour the supernatant in the previously used 2L flasks (apart from 15 ml supernatant per 500ml bottle), add 10% bleach to the flask, and dispose in the sink after 20 min.
- Use the 10 ml remaining supernatant to break the pellet in each bottle using a 10 ml serological pipette.
- Transfer the four 10 ml bacteria suspension from the four 500ml centrifugation tubes into a single 50 ml JA centrifugation tube.
  - Bleach the plastic JLA bottles, and dispose the bleach in the sink after 20 min.
- Centrifuge the 40 ml bacteria suspension at 6500×g, 10 min, 4°C in a Beckman floor centrifuge loaded with a Beckman JA 17 rotor. Follow the above safety precautions.
- Pour the supernatant in the previously used 2L flasks, add bleach and dispose the flask context after 20 min.
- Either store the bacteria pellet in the 50ml centrifugation tube at -80°C, or proceed immediately to the protein purification protocol.

H.7. Prepare Bacteria Extracts for SDS-PAGE

Supplies

- Bacteria pellet from 100 ul aliquots of 40x OD_{595}=0.75 bacteria cultures, stored at -80°C.

Procedure

- Resuspend the bacteria pellet in 100 ul PBS.
- Add 24 ul bacteria pellet in PBS to 8 ul 4x Laemmli SDS reducing buffer.
- Heat at 90-95°C for five minutes and run SDS-PAGE gel.

H.8. Transform Competent Bacteria

Supplies

- E-coli bacteria (not either competent or not), stored at -80 deg.
- LB-Agar coated plates (see protocol "Preparation of LB-Agar-Ampicillin Plates")
- LB-Agar-ampicillin plates (see protocol "Preparation of LB-Agar-Ampicillin Plates")
- LB medium (see protocol "Preparation of Sterile LB-Ampicillin Medium")
- CaCl2 (Mallinckrodt #4160)
- PIPES (Sigma #P1851)
- Round-bottom test tubes, polypropylene, 14 mL (BD falcon #352059)
- 50 mL centrifuge bottles for JA17 Beckman centrifuge rotors (max 50000 g)
- Inoculating loops (VWR #90001-100)

Equipment

- Autoclave (Yannas Lab ,3-315)
- Spectrophotometer (Dedon Lab, 2nd floor NE47)
- -80 deg freezer (Dedon Lab, 2nd floor NE47)
- Ultracentrifuge (Dedon Lab, 2nd floor NE47)
- 37 deg bacteria incubator (Dedon Lab, 2nd floor NE47)
- 37 deg shaking bacteria incubator (Dedon Lab, 2nd floor NE47)

Procedure

Prepare Buffers for Making Bacteria Competent

- Prepare 110 mL $CaCl_2$ working solution (60 mM $CaCl_2$, 15% glycerol, 10 mM PIPES pH 7).
    - Add 13.2 mL 500 mM $CaCl_2$, 11 mL 100 mM PIPES pH 7, 16.5 mL glycerol, 69.3 mL sterile dH2O. Then autoclave (liquid cycle) and store at room temperature.
- Prepare 50 mL LB medium in a 100 mL bottle.
- Prepare 200 mL LB medium in a 1L flask.
- Prepare LB-agar plates (without ampicillin).

Bacteria Culture for Competent Bacteria Preparation

- Take a cryovial that contains untransformed, not competent E-coli out of the -80 freezer.
- Touch the frozen bacteria suspension with an inoculation loop and streak an LB-agar plate.
    - Place the bacteria cryovial back to -80 deg quickly before the suspension thaws.
- Incubate the LB-agar plate for 12-16h in a 37 deg incubator.
    - The next day colonies will form.
- Use an inoculation loop to pick a single colony and inoculate it into 4 mL sterile LB medium inside a 14 mL test tube.
- Incubate the test tubes overnight in a shaking incubator (37 deg, 250 rpm).
- Transfer 2 mL bacteria suspension in a 1L flask containing 200 mL sterile LB medium.
- Incubate in a shaking incubator (37 deg, 250 rpm) until the absorbance OD_{595}=0.4 of the bacteria suspension.
    - The O.D. can be measured by a spectrophotomerer (Dedon lab).

Prepare Competent Bacteria

- Place the $CaCl_2$ working solution on ice.

- Aliquote the bacteria culture in four 50 mL ice-cold JA-17 tubes. Incubate the tubes in ice for 10 minutes.
- Centrifuge the tubes at 1600×g, 7 min, 4 deg in an ultracentrifuge (Dedon lab).
- Aspirate and dispose the supernatants. Resuspend each pellet gently in 10 mL ice-cold $CaCl_2$ working solution, while the tubes are placed in ice.
- Centrifuge the 50 mL tubes at 1100×g, 5 min, 4 deg, in an ultracentrifuge (Dedon lab).
- Aspirate and dispose the supernatants. Resuspend each pellet gently with 10 mL ice-cold $CaCl_2$ working solution.
- Incubate the 50 mL JA-17 tubes in ice for 30 min.
- Centrifuge the tubes at 1100g, 5 min, 4 deg in an ultracentrifuge (Dedon lab).
- Aspirate and dispose the supernatants. Resuspend each pellet gently with 2 mL ice-cold $CaCl_2$ working solution. Make sure there are no bacteria chunks in the suspensions.
- Dispense bacteria into 250 uL aliquots inside 1.5 mL eppendorf tubes. Make sure the aliquots are kept in ice. Store aliquots immediately at -80 deg.

Transform Competent Bacteria

- Set the warm water bath to 42 deg,
- Aliquote 10 ng plasmid DNA in 20 uL dH2O inside an ice-cold 14 mL sterile round-bottom test tube Place the tube in ice. Add as negative control a tube that doesn't contain DNA.
- Rapidly thaw the competent bacteria suspensions by hand warming.
- Immediately after thawing, transfer 100 uL of the competent bacteria culture inside each DNA-containing tube. Mix by tapping. Place in ice for 20 min.
- Heat shock cells by placing the test tubes in a water bath set to 42 deg for 2 minutes.
- Add 1 mL of LB medium per test tube.
- Incubate for 1h in a shaking incubator (37 deg, shaking at 250 rpm).
- Transfer 50 uL of bacteria suspension on a LB-agar-ampicillin plate.
    - Store the remaining bacteria suspension at 4 deg until further use.
- Use an inoculating loop to spread the 50 ul bacteria suspension.
- Allow plate to dry in room temperature for 30 minutes.
- Incubate inverted plates overnight in a 37 deg incubator.
    - Only bacteria transformed with plasmids contain ampicillin-resistant genes and form colonies.

H.9. Store E-Coli Cultures

Supplies

- LB (USB #75854)
- LB-Agar (USB #75851)
- Ampicillin (Sigma #A9393)
- Glycerol (USB #16374)
- Pippetors and tips.
- Cryovials.

Equipment

- Autoclave
- -80 deg freezer
- Centrifuge

Procedure

- Prepare a 20% glycerol aqueous solution. Autoclave the solution at liquid setting. Store at room temperature.
- Centrifuge the bacteria culture at 6000g, 10 min.
- Aspirate and discard supernatant.
- Resuspend the pellet in LB medium, in 1/20 of the original volume. Break pellet gently
- Add an equal amount of autoclaved 20% glycerol (to final 10% glycerol).
- Mix gently by inverting the tube.
- Label cryovials.
- Use a 1 mL pipettor to aliquote 1 mL of bacteria glycerol suspension inside the cryovials.
- Place the cryovials in a -80 deg freezer.

Appendix I: Molecular Biology Protocols

Contents

I.1. Measure DNA Concentration using the Nanodrop Spectrophotometer

Equipment

- Thermo Scientific Nanodrop Spectrophotometer ND-1000 (Newman Lab, 68-380)

Procedure

- Always use gloved hands when using the device and the computer since there may be ethidium bromide.
- Open the cover of the instrument.
- Turn on the software ND-1000. Choose "nucleotides" in the new window that opens.
- As the software requests, add a 2 uL aliquote on the bottom imaging base, close cover.
- Use a kimwipe to remove the aliquote.
- Add a 2 uL aliquote of the buffer of the DNA solution on the bottom imaging base, close cover.
- Click the "blank" button.
- Use a kimwipe to remove the aliquote.
- Add a 2 uL aliquote of the DNA solution on the bottom imaging base, close cover.
- Click the "measure" button.
- The instrument provides the emission spectrum of the sample, the estimated concentration in ng/uL, as well as the rations of the O.D. at 260 versus 280.
- Use a kimwipe to remove the aliquote.

I.2. Agarose Gel Electrophoresis of DNA Solutions

Supplies

- TAE buffer (Boston Bioproducts # BM-250X)
- Ethidium Bromide solution (Sigma #E1385)
- Electrophoresis-grade agarose (Bio-rad #161-3100)
- DNA Ladder (New England Biolabs # N0468S)
- Gel Loading Buffer 6X (Boston Bioproducts #BM-100F)
- Nitrile gloves.

Equipment

- Water bath
- Horizontal gel electrophoresis apparatus
- Gel casting platform
- Gel slot formers (combs)
- DC power supply
- UV transilluminator (Dedon Lab, NE47 second floor)

Procedure

Preparing the agarose gel

- Add the desired amount of agarose (0.8 to 1.5%) in TAE buffer (50 mL for the yannas lab system). The final thickness of the gel should be between 0.5 and 1 cm.
- Create a proper volume of ethidium bromide working solution by diluting to a final concentration of 0.5 ug/mL using water. Add the ethidium bromide (EtBr) solution to the agarose solution.
- Melt the agarose solution by autoclaving. Let the solution cool in a waterbath set to 55 deg.
- Seal the gel casting platform at its open ends using adhesive tape.
- Pour the melted agarose solution and insert gel comb making sure no bubbles remain.
- Let the gel harder in room temperature. Remove the tape from the open ends and the gel comb.

Load and run the gel

- Place the gel casting platform that contains the hardened gel in the electrophoresis tank. The gel should be placed in such a way that DNA is initially close to the negative pole and will migrate towards the positive pole.
- Add sufficient TAE buffer to the electrophoresis tank so that the gel is just covered. Make sure no bubbles are trapped in the wells.
- Dilute the DNA samples to the final desired volume using the gel loading dye.
- Load the DNA samples and the DNA ladder in the wells using a pipette. Make sure that the DNA samples don't overflow the gels and samples from adjacent wells don't mix.
- Set the electrophoresis voltage to approximately 1 to 10 V/cm of gel length.
- The progress of the electrophoresis can be monitored by the migration of the DNA ladder and the bromophenol blue contained in the gel loading buffer.
- When the bromophenol blue had reached close enough to the positive end of the gel, turn off the power supply.
- Dispose the TAE buffer that contains EtBr as a hazardous waste.

Visualize the bands in the gel

- Place the gel on a large clean Petri dish. Since the gel contains ethidium bromide make sure to handle the gel with gloved (nitrile gloves, not latex!!!) hands.
- Place the Petri dish on a UV transilluminator (Dedon Lab, NE47 second floor).
- Select the proper zoom and focus the camera on the agarose gel surface.
- Dispose gels that contain ethidium bromide as hazardous, in appropriate containers.

I.3. Restriction Digestion of DNA

Supplies

- BamHI Restriction Enzyme, 10000U, 100000U/mL, (New England Bio Labs # R0136T)
- SalI Restriction Enzyme, 2000U, 100000U/mL, (New England Bio Labs # R0138T)
- Pipettes and tips (1 mL, 200 uL)
- Sterile PCR tubes

Equipment

- Heating block (Polymers and Fibers Lab, 3-315).
- Water bath (Polymers and Fibers Lab, 3-315).

Procedure

The following protocol is a 20 uL reaction that contains approximately 1 ug of DNA. For larger reactions scale all buffer volumes.

- Prepare the 20 µl reaction volume inside a sterile PCR tube
  - 1 ug DNA (purified plasmids, linearized vectors, oligos)
  - 2 uL of 10X Buffer (depends on restriction enzyme).
    - Double digestions can be done simultaneously when they require same buffer and enzymes do not interfere. Check New England Bio Labs manual. For example BamHI and SalI can be applied in parallel since they operate at the same buffer and temperature conditions. If this is not possible, run two reactions in series.
  - 2 uL BSA buffer (required by some enzymes) to final BSA concentration 100 ug/mL.
  - 1-5 units of each restriction enzyme per µg DNA.
    - Make sure total restriction enzyme solution is less than 10% of total reaction volume (e.g. don't add more than 2 µl of restriction enzymes to a 20 µl reaction).
  - The remaining should be molecular biology water.
- Incubate for 2h in the appropriate temperature (usually 37°C).
  - Check New England Bio Labs manual.
- In order to verify digestion, take a small aliquote for agarose gel electrophoresis.
- Inactivate the enzymes and stop the reaction.
  - Most enzymes can be deactivated by heating the solution for 15 min at 75°C.
  - Check New England Bio Labs manual for special cases. Another method is repeated freeze-thaw cycles.

I.4. Insert and Ligate a cDNA Construct in a Linearized Vector

Supplies

- ATP 10 mM (New England Biolabs #P0756S)
- T4 DNA Ligase, includes buffer (New England Biolabs #M0202S)
- Sterile PCR tubes

Equipment

- Heating block
- Tropicooler device (Dedon Lab, MIT).

Procedure

The number of open end moles for a mass m of a sd DNA of N bp length is:

$$\#moles = 2m/(649 * N)$$

1 pmol of ends of pGEX vector corresponds to approximately to 1.67 mg of pGEX DNA.

1 pmol of ends of TC-labeled I-domain corresponds to approx. 160 ug of I-domain-TC DNA.

- Inside a PCR tube add 0.1 ug of TC-label I-domain DNA (0.6 fmol of free ends, 0.03 uM) in 4 uL buffer, 1 ug of linearized pGEX vector (0.6 fmol of free ends, 0.03 uM) in 4 uL buffer, 0.2 uL of 100 mM ATP, 2 mL of 10X T4 Ligase buffer, and 20 units of T4 DNA ligase. Add water so that the total volume becomes 20 uL.
- Incubate the solution for 2h at 16°C in a tropi-cooler.
- Terminate reaction by heating at 65 deg for 10 min.
- The resulting solution (20 uL) contains DNA constructs made of 0.03 uM TC-label I-domain DNA and 0.03 uM linearized pGEX vector. The resulting solution can be used right away for bacteria transformation (preferably) or can be stored at -20 °C.

Appendix J: Biochemistry

Contents

J.1. Measure Protein Concentration via 280 nm Absorption

Supplies

- Protein sample of unknown concentration
- 10 ul pipette and tips
- Kimwipes

Equipment

- Nanodrop spectrophotometer (Weiss lab, NE47-235)

Application Note

Used to quantify protein samples whose density is at least 0.2 mg/ml. For more dilute samples, use the BCA method. Need to correct errors induced by DNA impurities using the A260/A280 ratio, however this correction requires $\bar{A}_{280} < 1.5$. Very dense samples need to be diluted to satisfy this limit.

Procedure

- Estimate molar absorption coefficient ε_{280} for the protein of interest
  - Count the number of tryptophan, tyrosine and cysteine residues W, Y and C in the sequence of the protein of interest.
  - Calculate ε_{280} based on the formula [Pace et al. 1995]:
    $\varepsilon_{280} = 5500W + 1490Y + 125C$ $[M^{-1}cm^{-1}]$
- Nanodrop measurement
  - Instrument provides $\bar{A}_{280}$ (280 absorption for d = 1 cm path)
  - Blank: Use 2ul buffer used to reconstitute protein sample3.
  - Measurement: Add 2ul protein sample.
- Estimate the concentration C and mass density ρ:

$$C = \gamma \cdot \frac{A_{280}}{d \cdot \varepsilon_{280}} = \gamma \cdot \frac{1}{\varepsilon_{280}} \cdot \bar{A}_{280} = \gamma \cdot C_0 \cdot \bar{A}_{280} \quad [M]$$

$$\rho = M \cdot C = \gamma \cdot \frac{M}{\varepsilon_{280}} \cdot \bar{A}_{280} = \gamma \cdot \rho_0 \cdot \bar{A}_{280} \quad [mg/ml]$$

$$\gamma = 1.55 - 0.76 \frac{\bar{A}_{260}}{\bar{A}_{280}}$$

where M is the protein's molecular weight [g/mole]. The correction factor γ (Warburg-Christian method) corrects the 280nm absorption by nucleic acid impurities based on the $\bar{A}_{260}/\bar{A}_{280}$ ratio (correction is valid when $\bar{A}_{280} < 1.5$. If $\bar{A}_{280} > 1.5$ repeat measurement with diluted sample)

Protein	Accession	Locus	MW [g/mole]	W	Y	C	ε_{280} $[M^{-1}cm^{-1}]$	ρ_0 [µg/µl]	C_0 [uM]
BSA fraction V	CAA76847		69323	3	21	35	52165	1.329	19.171
GST	U13850	258-956	26986	4	14	4	43360	0.622	23.049
A1	P56199	151-364	23817	1	4	2	11710	2.034	85.401
A2	P17301	154-368	23504	1	9	3	19285	1.219	51.864
A10	O75578	145-361	24023	2	5	3	18825	1.276	53.116
A11	Q9UKX5	142-356	24250	1	9	1	19035	1.274	52.536
TC (WDCCPGCCK)			1014	1	0	4	6000	0.169	166.67

TC-A1			24831	2	4	6	17710	1.402	56.462
TC-A2			24518	2	9	7	25285	0.970	39.527
GST-A1			50803	5	18	6	55070	0.922	18.148
GST-A2			50490	5	23	7	62645	0.806	16.003
GST-TC-A1			51817	6	18	10	61070	0.848	16.365
GST-TC-A2			51504	6	23	11	68645	0.750	14.562
Collagen I, bovine	2×P02453 1×P02465	162-1215 80-1100	280482	0	11	0	16390	17.113	61.013
Fibronectin 7-15	P02751	1173-1630 1723-1991 2100-2190	87979	9	29	2	92960	0.946	10.757
Fibronectin 7-15 EIIIA	P02751	1173-1991 2100-2190	97823	10	32	2	102930	0.950	9.715
Fibronectin V	P02751	1992-2099	12059	0	4	0	5960	2.023	167.785
Fibronectin 7-15 V	P02751	1173-1630 1723-2190	100038	9	33	2	98920	1.011	10.109
Fibronectin 7-15 EIIIA V	P02751	1173-2190	109882	10	36	2	108890	1.009	9.184
GST-Fibronectin 7-15 V			127024	13	47	6	142280	0.893	7.030
GST-Fibronectin 7-15 EIIIA V			136868	14	50	6	152250	0.899	6.568

Table J1-7-3: Parameters for the estimation of the concentration of a protein solution based on A_{280}.

J.2. Measure Protein Concentration via the BCA Assay

Supplies

- Protein sample of unknown concentration
- BCA kit (Pierce23227)
- BSA protein standards 2mg/ml in PBS (Pierce)

Equipment

- Nanodrop spectrophotometer (Weiss lab, NE47-235)

Application Note

Can be used to quantify protein samples whose density is as low as 0.5 µg/ml and as high as 2 mg/ml. More robust than the Bradford assay, however it is also sensitive to several reagents (e.g. TCEP), for more info check the Pierce application note in references.

Procedure

- Prepare a series of protein standards (2, 1, 0.5, 0.25, 0.125, 1/16, 1/32 mg/ml) by diluting the stock 2 mg/ml BSA standard in the corresponding buffer.
  - Protein standards should be in same buffer as proteins to be measured.
- Prepare at least 5 µl working reagent (WR) per sample (standards or unknown) to be measured
  - Add 25 parts of BCA reagent A to 1 part BCA reagent B. Mix until solution becomes homogenous.
- For each sample (standards or unknown), add 5 µl WR to 5 µl sample.
- Incubate samples 30 min at 37°C.
- Measure 562 nm absorption in a nanodrop spectrophotometer
  - Use 3 ul BCA-sample solution per measurement.
  - Measure first known protein standards in order to get a standard curve.

J.3. Measure Protein Concentration via Bradford Protein Assay

Supplies

- Protein sample of unknown concentration
- Quickstart bradford assay dye (Bio-rad 500-0205)
- BSA protein standards 2mg/ml in PBS (Pierce)

Equipment

- Nanodrop spectrophotometer (Weiss lab, NE47-235)

Application Note

Can be used to quantify protein samples whose density is as low as 0.5 µg/ml and as high as 2 mg/ml. Bradford (coomasie) assay is sensitive to detergents (check the Pierce application note in references), however it is fast and can be used to quickly estimate protein concentration.

Procedure

- Prepare a series of protein standards (2, 1, 0.5, 0.25, 0.125, 1/16, 1/32 mg/ml) by diluting the stock 2 mg/ml BSA standard in the corresponding buffer.
  - Protein standards should be diluted in same buffer as proteins to be measured.
  - Buffers tested: PBS, PBS+++ (PBS, 2.5 mM MgCl2, 1 mM TCEP, 0.02% tween 20), 0.05M Acetic acid, glutathione elution buffer (10 mM glutathione in 50 mM Tris HCl pH 8).
- Mix each sample (standards or unknown) with Bradford assay reagent (need at least 3 ul mixture per sample).
  - Use 1:1 sample:bradford reagent mixture for very dilute samples (up to 0.1 mg/ml).
  - Use 1:32 sample:bradford reagent mixture for samples up to 2 mg/ml.
- Mix well and incubate 10 min at room temperature.
- Fast qualitative conclusions are based on comparing the color of the coomasie-sample with the color of the standards
- Quantitative measurements: Measure 595 nm absorption in a nanodrop spectrophotometer
  - Use 3 ul bradford-sample solution per sample.
  - Measure first known protein standards in order to get a standard curve.

J.4. SDS-PAGE Electrophoresis of Protein Samples

Supplies

- Protein solution.
- Ready Gel Tris-HCl gel, 4-15%, 10-well 30 µL (Bio-Rad 161-1104) stored at 4°C.
- Tris-Glycine-SDS Running Buffer 10× (Boston BioProducts BP-150).
- Laemmli's non-reducing SDS Buffer 4× (Boston BioProducts #BP-110NR).
- Precision Plus All Blue standards (Biorad 161-0373).
- Invitrogen Simply Blue stain, 1L (Invitrogen LC6060).
- Acetic Acid glacial (Mallinckrodt).
- Methanol (Mallinckrodt).
- β-mercaptoethanol (Sigma Aldrich 63689).
- 4% NaCl solution (0.68 M).
- 1M NaOH solution.

Equipment

- Mini-PROTEAN Tetra Cell (BioRad 165-8030), (room 3-315).
- Power supply, able to provide up to ~200V and up to ~50 mA (room 3-315).
- Dry block heater, includes 1.5 mL eppendorf tube block (room 3-315).
- Camera apparatus and FluorChem8900 image acquisition software (Dedon Lab).

Procedure

Preparation

- Rinse the Mini-PROTEAN Tetra cell with de-ionized water.
  - The cell can be used to run up to 4 gels at the same time.
- Turn on the dry heating block and set the temperature at 95°C.
- Prepare 1L 1× Tris-Glycine-SDS running buffer by mixing 100 ml of 10× Tris-Glycine-SDS running buffer with 900 ml clean water inside a bottle.
  - The running buffer can be re-used 2-3 times. It is stored at 4°C.
- Prepare V µl 3.7× reducing Laemmli buffer (RLB).
  - Add 92.5V µl 4× Laemmli non-reducing buffer to 7.5V µl β-mercaptoethanol.

Position the Gel

- Take one ready gel, cut the black tape on the bottom using a razor, and remove slowly and carefully the cover of the gel wells by gentle pulling.
- Place the ready gel in the electrophoresis cell so that wells face inner side of the electrophoresis cell. Secure the gel using the two handles.
  - Use the cell that contains two electric plugs on the top, if 1 or 2 gels will be run. Use the second cell (doesn't contain plugs) if more than 2 gels will be run.
  - When running 1 or 3 ready gels, use the provided buffer dams to cover the extra empty spot of the cell.
  - Making sure that red/black plugs are on the red/black marked sides of the tank.
  - The space between two gels or between a gel and a dam forms an "inner well".
- Place the cell-ready gel-dam assembly inside the tank.
- Add 1× running buffer in the inner well until the gel wells are fully covered.
- Add running buffer to the tank up to the signed level (for running 1-2 or 3-4 gels).

Prepare Protein Samples

- Prepare the samples inside clean 0.5 ml eppendorf tubes. Use one tube per sample.

- Each sample has volume v ul (10 - 30 ul, check max volume that can be loaded)
- Each sample should contain 1 – 10 ug protein in the case it contains few proteins or 25-50 ug in case it contains a complex mix of proteins.

- Prepare samples by adding v/3.7 3.7× RLB and 0.73v protein solution.
- Check and correct if necessary the pH of the samples
  - The color of the samples should be dark blue (same as the Laemmli buffer).
  - Use a 10 ml pipette to add tiny drops of 1M NaOH to correct the color.
- Store the samples at 4 deg until use.

Load the Gel

- Heat the electrophoresis protein samples in the dry heater at 95°C for five minutes.
- Let the electrophoresis protein samples cool back to room temperature.
- Micro-centrifuge the electrophoresis protein samples for 10 sec.
- Use a 10 µl pipette to load each protein sample (max 10-30 ul per well) inside a well of the ready gel.
  - Only the fine 10 ul pipette can enter the wells without damaging the gel.
  - If possible, leave empty lanes between samples.
- Use a 10 µl pipette to load 10 µl protein standard to one or two gel wells.

Run the Gel

- Place the lid of the assembly on top on the tank-cells.
- Connect the electrodes to the power supply using the cables.
- Turn on the power supply and select the applied voltage and electrophoresis time.
  - Run the gel at 60V for 3h or 32V for 6h or 24V for 8h.
  - Electrophoresis progress can be monitored using the bromophenol blue dye line.

Retrieve the Polyacrylamide Gel

- After the gel is run, drain the electrophoresis buffer (it can be disposed in the sink) or store it at 4 deg so that it can be re-used (up to 3 times total).
- Unsecure the handles and remove the ready gels from the electrophoresis cell.
  - Each ready gel contains a polyacrylamide gel sandwiched between plastic covers.
- Remove the transparent tape that connects two plastic covers.
- Gently separate the two plastic covers. The polyacrylamide gel will remain attached to one of the two plastic covers.
  - Don't touch the polyacrylamide gel without gloves. Acrylamide is neurotoxic.
- Transfer the polyacrylamide gel inside a small plastic that contains a small amount of water (acts as a lubricant; friction could cause the gel to break during handling).
  - Be careful when transferring the gel to avoid breaking it. Use both hands gently.

Coomasie Staining (Fast Protocol)

- Prepare 35 ml staining solution (10% acetic acid, 10% methanol, 20% commasie)
  - Add 7 ml simply blue coomasie stain to 1.25× stock staining solution (12.5% acetic acid, 12.5% methanol) stored at room temperature.
  - Handle staining solution in the chemical hood as it is toxic and irritating.
- Pour 35 ml diluted staining solution to the gel.
- Incubate at room temperature inside the chemical hood for 8h.
  - Within 2h the staining pattern should appear.
- Dispose the staining solution (according to EHS).
- Wash gel with 20 ml deionized water (dispose with staining solution according to EHS). Add 20 ml deionized water and incubate at least 2h before taking a picture.

Coomasie Staining (Standard Protocol – Not Used Anymore)

- Wash the gel three times with clean water.
- Add 15-20 ml simply blue coomasie stain (barely cover the gel).

- Incubate overnight at room temperature with gentle shaking.
- Dispose stain in the sink and wash the gel twice with deionized water.
- Add 20 ml 4% NaCl and incubate at room temperature for 2h with gentle shaking.
  - This steps aims to reduce non-specific binding of coomasie and enhance contrast.
- Dispose the NaCl solution, wash the gel twice with deionized water.
- Add 20 ml deionized water and incubate overnight at room temperature.
  - This step makes the coomasie staining more bright.

Imaging the Gel

- Clean a flat glass plate (larger than the gel) with 70% ethanol.
- Use both hands to transfer carefully the polyacrylamide gel on the glass plate.
- Open the imaging system (FluorChem8900) door, un-flip the white light, place the glass plate on the white light and close the door.
- Choose filter wheel position 1 (no filter), turn on the transilluminant white light ("transilluminesence white" button).
- Log in the computer (login: guest, password: growl) and run the "fluorchem8900" acquisition software. The login and password for the software are the same.
- Press the "acquire" button to start acquisition. Manually adjust the zoom, focus and iris size of the camera until the sample is in focus and the image has good brightness and contrast.
- Click the "acquire image" button to acquire an image of the gel.
- Use the "white", "black" and "gamma" settings to modify the brightness, contrast and gamma correction of the image and verify that the image is good enough.
- Save the image as a "tif" file bly choosing the menu "file">"save".

J.5. SDS-PAGE with FlAsH Detection of Tetracysteine-Tagged Proteins

Supplies

- Protein sample.
- Laemmli' non-reducing SDS Buffer 4X.
- Ready Gel Tris-HCl gel, 4-15%, 10-well 30 µL (Bio-Rad #161-1104); store at 4 °C.
- Tris-Glycine-SDS Running Buffer 10X (Boston BioProducts #BP-150).
- Precision Plus All Blue standards (Biorad #161-0373).
- TC-FlAsH In-cell tetracysteine tag detection kit (Invitrogen #T34561). The FlAsH dye is stored in 100 µM aliquots at -20 deg.
- TCEP Hydrochloride solution 0.5M pH 7 (Sigma Aldrich #646547).
- Invitrogen Simply Blue stain, 1L (Invitrogen #LC6060).
- 4% NaCl solution (0.68 M).
- 1M NaOH solution.

Equipment

- Mini-PROTEAN Tetra Cell (BioRad #165-8030), (room 3-315).
- Power supply, able to provide up to ~200V and up to ~50 mA (room 3-315).
- Dry block heater, includes 1.5 mL eppendorf tube block (room 3-315).
- Camera apparatus and FluorChem8900 image acquisition software (Dedon Lab).

Procedure

Preparation

- Rinse the Mini-PROTEAN Tetra cell with de-ionized water.
  - The cell can be used to run up to 4 gels at the same time.
- Turn on the dry heating block and set the temperature at 95 °C.
- Prepare 1L 1X Tris-Glycine-SDS running buffer by mixing 100 ml of 10X Tris-Glycine-SDS running buffer with 900 ml clean water inside a bottle.
  - One can prepare the running buffer from Tris, glycine and SDS.
  - The running buffer can be re-used 2-3 times. It is stored at 4 deg.

Position the Gel in the Cell

- Take one ready gel, cut the black tape on the bottom using a razor, and remove slowly and carefully the cover of the gel wells by gentle pulling.
- Place the ready gel in the electrophoresis cell so that wells face inner side of the electrophoresis cell. Secure the gel using the two handles.
  - Use the cell that contains two electric plugs on the top, if 1 or 2 gels will be run. Use the second cell (doesn't contain plugs) if more than 2 gels will be run.
  - When running 1 or 3 ready gels, use the provided buffer dams to cover the extra empty spot of the cell.
  - Making sure that red/black plugs are on the red/black marked sides of the tank.
  - The space between two gels or between a gel and a dam forms an "inner well".
- Place the cell-ready gel-dam assembly inside the tank.
- Add 1X running buffer in the inner well until the gel wells are fully covered.
- Add running buffer to the tank up to the signed level (for running 1-2 or 3-4 gels).

Electrophoresis Protein Samples Preparation

- Prepare the samples inside clean 1.5 ml eppendorf tubes. Use one tube per sample.

- The maximum volume v of each sample is limited by the size of the gel wells (usually between 10 - 30 ul).
- Each sample should contain 4-5 µg TC-tagged protein. Minimum detection level is aproximately 1.5 µg (SNR approximately 2)

- Prepare the samples by mixing protein samples (see above comment for the proper amount of protein to be added), v/4 4× non-reducing Laemmli SDS buffer, TCP (final concentration 5 mM) and deionized water to total volume 19v/20.
- Check and correct if necessary the pH of the samples
  - The color of the samples should be dark blue (same as the Laemmli buffer).
  - Use a 10 ml pipette to add tiny drops of 1M NaOH to correct the color.
- Store the samples at 4 deg until use.

FlAsH Staining

- Heat the electrophoresis protein samples in the dry heater (95 deg) for five minutes.
- Let the electrophoresis protein samples cool back to room temperature.
- Micro-centrifuge the electrophoresis protein samples for 10 sec.
- Add v/20 40 uM FlAsH dye (2 uM final concentration).
- Incubate the protein samples for 30 – 45 min at room temperature.

Load the Gel

- Use a 10 µl pipette to load each electrophoresis protein sample (max 10-30 ul per well) inside a well of the ready gel.
  - Only the fine 10 ul pipette can enter the wells without damaging the gel.
  - If possible, leave empty lanes between samples.
- Use a 10 µl pipette to load 10 µl protein standard to one or two gel wells.

Run the Gel

- Place the lid of the assembly on top on the tank-cells.
- Connect the electrodes to the power supply using the cables.
- Turn on the power supply and select the applied voltage and electrophoresis time.
  - The gels run within approximately 3h/6h when 60V/30V is applied.
  - The protein displacement progress can be monitored using the blue line made of the tiny bromophenol blue dye.

Retrieve the Polyacrylamide Gel

- After the gel is run, drain the electrophoresis buffer (it can be disposed in the sink) or store it at 4 deg so that it can be re-used (up to 3 times total).
- Unsecure the handles and remove the ready gels from the electrophoresis cell.
  - Each ready gel contains a polyacrylamide gel sandwiched between plastic covers.
- Remove the transparent tape that connects two plastic covers.
- Gently separate the two plastic covers. The polyacrylamide gel will remain attached to one of the two plastic covers.
  - Don't touch the polyacrylamide gel without gloves. Acrylamide is neurotoxic.

Fluorescent Imaging of FlAsH-labeled Proteins

- Clean a flat glass plate (larger than the gel) with 70% ethanol.
- Use both hands to transfer carefully the polyacrylamide gel on the glass plate.
  - Always wear nitrile gloves!
  - Be careful when transferring the gel to avoid breaking it. Use both hands gently.
- Open the imaging system (FluorChem8900) door, flip the white light to expose the UV illuminator, place the glass plate on the UV illuminator and close the door.
- Choose filter wheel position 3, turn on the transilluminant white light ("transilluminesence white" button).
- Log in the computer (login: guest, password: growl) and run the "fluorchem8900" acquisition

software. The login and password for the software are the same.

- Press the "acquire" button to start acquisition. Manually adjust the zoom, focus and iris size of the camera until the sample is in focus and the image has good brightness and contrast.
- Choose filter wheel position 1 (SYBR green filter), turn on the UV transilluminant light ("transilluminesence UV" button).
- Press the "acquire" button to start acquisition. Adjust the exposure time in order to acquire an image as sharp as possible.
- Click the "acquire image" button to acquire an image of the gel.
- Use the "white", "black" and "gamma" settings to modify the brightness, contrast and gamma correction of the image and verify that the image is good enough.
- Save the image as a "tif" file bly choosing the menu "file">"save".

Coomasie Staining

- Transfer the polyacrylamide gel inside a small plastic that contains a small amount of water (acts as a lubricant; friction could cause the gel to break during handling).
  - Be careful when transferring the gel to avoid breaking it. Use both hands gently.
- Wash the gel three times with clean water.
- Add a small amount of simply blue coomasie stain (barely cover the gel).
- Incubate the box overnight at room temperature with gentle shaking.

De-Staining

- Dispose the stain in the sink and wash the gel twice with clean water.
- Add 20 ml 4% NaCl solution and incubate at room temperature for 2h with gentle shaking.
  - This steps aims to reduce non-specific binding of coomasie and enhance contrast.
- Dispose the NaCl solution, wash the gel twice with clean water.
- Add 10 ml water (barely cover the gel) and incubate overnight at room temperature.
  - This step makes the coomasie staining more bright.

Imaging the Gel

- Clean a flat glass plate (larger than the gel) with 70% ethanol.
- Use both hands to transfer carefully the polyacrylamide gel on the glass plate.
- Open the imaging system (FluorChem8900) door, un-flip the white light, place the glass plate on the white light and close the door.
- Choose filter wheel position 1 (no filter), turn on the transilluminant white light ("transilluminesence white" button).
- Log in the computer (login: guest, password: growl) and run the "fluorchem8900" acquisition software. The login and password for the software are the same.
- Press the "acquire" button to start acquisition. Manually adjust the zoom, focus and iris size of the camera until the sample is in focus and the image has good brightness and contrast.
- Click the "acquire image" button to acquire an image of the gel.
- Use the "white", "black" and "gamma" settings to modify the brightness, contrast and gamma correction of the image and verify that the image is good enough.
- Save the image as a "tif" file bly choosing the menu "file">"save".

J.6. Purification of GST-Tagged Proteins Expressed in E-Coli

Supplies

- Autoclaved distilled dionized water (αddH2O) (Cellgro #25-055-CV)
- Immobilized glutathione in agarose beads (Pierce #15160), stored at 4oC.
- L-glutathione reduced (Sigma # G4251), stored at 4oC.
- Phosphate-buffered saline (Boston Bioproducts #BM-220S)
- EDTA 0.5M (Boston Bioproducts #BM-150)
- DTT 1M (Sigma Aldrich #43816)
- Sodium azide powder (Sigma Aldrich #438456)
- Halt protease inhibitor single-use cocktail (Pierce #78430)
- Amicon YM3 centrifugal filter device, 3 kDa MWCO (Millipore #UFC800308)
- 5 ml polystyrene round bottom test tube (BD #352054)
- Screw-cap microcentrifuge tubes (VWR #80089-660)
- 50 ml centrifuge bottles for JA17 Beckman centrifuge rotors (max 50000 g)
- 15 ml polypropylene centrifuge tubes (BD #352070)
- Lysozyme power (Sigma #L6876), stored at -80oC
- DNAse I (Sigma #D4527), stored in 5 mg/ml in 0.15M NaCl aliquots at -80oC or -20oC.
- Benzonase (Novagen #70746-3).
- 10× phosphate buffered saline without divalent cations (PBS-) (Invitrogen 70011-044).
- Thrombin protease stock solution, 500 units reconstituted in 1 ml PBS, divided in twenty 50 µl aliquots (G.E. healthcare 27-0846-01).
- 3 ml slide-a-lyzer dialysis cassette, 3.5 kDa MWCO (Thermo scientific 66330)

Equipment

- Ice machine (NE47, autoclave room)
- Cooled tabletop centrifuge (3-315)
- Microcentrifuge placed inside a 4oC cold box (3-315)
- Tip sonificator (Dedon Lab, NE47 second floor, cold room)
- Cooled floor centrifuge (Palker Elkin, Dedon Lab, NE 47 second floor).
- Nutator shaker (3-315).

Procedure

- This protocol describes the batch purification of a GST-tagged protein expressed in a 200 ml e-coli culture. Bacteria have been pelleted in a JA17 50 ml bottle
    - This protocol may take place immediately after bacteria pelleting. Alternatively, pelleted bacteria may have been stored at -80°C.
- Affinity purification of GST-tagged proteins uses glutathione immobilized on agarose beads.
    - Do not vortex the bead slurry, because this can damage the beads. Instead, mix bead suspensions by gentle tipping and inverting.
    - Before transferring bead slurry by pipetting, cut the front of the pipette tip using a razor. This prevents damaging the beads during transport.
- It takes about 20 min for the tabletop centrifuge to stabilize its temperature at 4°C.

<u>Preparation</u>

- For each 200 ml bacteria pellet, prepare the following:
- 1ml 50% glutathione bead slurry (prepare fresh)
    - Shake gently the immobilized glutathione bottle to resuspend the 50% bead slurry (bed volume is half the slurry volume) that contains 0.05% sodium azide.

- Transfer 1 ml bead slurry into an ice-cold 5 ml test tube.
- Add 4 ml ice-cold PBS, mix bead suspension by tipping and inverting the tube.
- Centrifuge the tube 190×g for 2 minutes at 4oC.
- Aspirate supernatant without disturbing bead pellet.
- Repeat the PBS wash three more times.
- Store the 50% glutathione-agarose bead slury at 4oC until use.

- 5.25 ml lysis buffer [PBS-, protease inhibitors, 1 mM DTT, 2.5 mM MgCl2, 1% Triton X100]
  - Recipe for 22 ml: 2.2 ml 10× PBS-, 19.53 αddH2O, 44 µl 1M MgCl2, 22 µl 1M DTT, 200 µl 100× protease inhibitors.
  - Add DTT and protease inhibitors just before use.
- 150 µl 40× lysozyme solution (20 mg/ml) in PBS
  - Recipe for 620 µl: 12.4 mg lysozyme, 62 ul 10× PBS, and 550 ul αddH2O.
- 600 µl 10% triton X-100 in PBS
  - Recipe for 2.5ml: 1.25 ml 20% triton x-100, 250 ul 10× PBS, and 1ml αddH2O.
- 20 ml wash buffer 1 [PBS, 1 mM DTT, 0.1% Triton X100, 2.5 mM MgCl2, 1/10×protease inhibitors]
  - Recipe for 80 ml: 8 ml 10×PBS-, 80 µl 1M DTT, 400 µl 20% TX100, 200 µl 1M MgCl2, 80 µl 100× protease inhibitor cocktail, 71.25 ml addH2O.
  - Add DTT and protease inhibitors just before use.
  - Use ¼ of wash buffer 1 to prepare the "dnaK" buffer
  - Store at 4oC until use.
- 5 ml DnaK removal buffer [wash buffer 1, 2.5 mM ATP]
  - Recipe: Add 6.875 mg ATP per 5 ml wash buffer 1
  - Store at 4oC until use.
- 20 ml wash buffer 2 [PBS]
  - Recipe for 80 ml: 8 ml PBS-, 72 ml addH2O.
  - Store at 4oC until use.
- 3 ml elution buffer [10 mM reduced glutathione, 50 mM Tris pH8.0]
  - Recipe for 12.5 ml: 38.7 mg reduced glutathione, 625 µl 1M TrisHCl pH 8, to 11.875 ml addH2O. Check pH=8.
  - Add reduced glutathione just before use. Store at 4oC until use.

For each 200 ml bacteria pellet, make sure the following aliquots are available:

- 10 µl 5mg/ml DNAseI stock solution stored at -80oC or -20oC.
- 3 µl (75 units) benzonase nuclease stored at -20oC.
- 25 µl 1 U/µl (0.2 NIH U/µl) thrombin stock solution stored at -20oC or preferably -80oC.

Cell Lysis

Resuspend bacteria in lysis buffer (1.5h + 15 min per sample)

During cell lysis keep the 50 ml JA17 tubes that contain bacteria pellets in a bucket containing ice.

- If bacteria were stored in -80°C, incubate the bacteria pellet on ice for 30 min to let bacteria thaw.
- Add 5.25 ml lysis buffer into the JA17 bottle.
- Break pellet and resuspend bacteria in lysis buffer by vortexing and using a serological pipette.
  - Make sure no bacteria chunks remain in the suspension before sonification.
- Add 150ul 40× lysozyme solution, 10 ul 5mg/ml DNAseI solution and 3µl 50U benzonase to each sample.

- Incubate 1h on ice. Lysozyme digests bacteria walls. DNase I and benzonase chop the released genomic DNA and reduce solution viscosity.
- If the extract remains viscous add 5 ul more 5mg/ml DNAseI or 2µl more benzonase, and incubate more time.

Sonication (15 min per sample)
During lysis keep the 50 ml JA17 tubes that contain bacteria pellets in a bucket containing ice.
- Prepare the sonification tip:
  - Before sonification of each sample, wipe sonification tip with 70% ethanol in kimwipes.
- Sonicate the bacteria suspension by applying 4 pulsetrains (each 100% duty cycle, 10 sec long). Let the suspension cool on ice for 60 sec between pulsetrains.
  - Start and stop the sonification using the time knob.
  - Adjust sonification power resonance to maximize sonification power (resonance phenomenon). This takes place around power setting 1.5.
  - When the cells are lysed, the bacteria suspension becomes darker and less turbid.
- Add 600 µl ice-cold 10% triton x-100 in PBS. Mix until solution becomes homogenous.
  - The total volume of each cell extract is approximately 6 ml.
- Incubate 15 minutes on ice.
  - If extract becomes viscous add 10 ul 5mg/ml DNAseI or 20U benzonase, and incubate more.
- Centrifuge sonicated bacteria suspension 14000×g for 60 minutes at 4°C.
  - If viscous solution remains between the solid pellet and the supernatant, transport the supernatant and the viscous solution in a new clean JA17 tube and centrifuge 14000×g for 40 minutes at 4°C.
- Transfer protein extract into an ice-cold 15 ml conical tube placed on ice.
  - Make sure that no pellet or insoluble material is transferred.

Affinity Purification by Glutathione-agarose beads

Catch GST-tagged proteins using glutathione-agarose beads (2h)
- Transfer the 6 ml protein extract into the 15 ml test tube that contains the 1 ml 50% glutathione-agarose bead slurry.
- Incubate 3h on a nutator at 4°C.

Remove unbound fraction (5 min + 5 min per sample)
- Pellet the bead suspension.
  - Bead pelleting is done by centrifuging 190×g for 1 min at 4°C
- Aspirate gently supernatant without disturbing the bead pellet. Store at 4°C for troubleshooting ("unbound" fraction).

Wash beads (15 min + 10 min per sample)

- Add 5 ml ice-cold wash buffer 1 to each bead pellet, resuspend beads by gentle shaking.
- Pellet the bead suspension. Aspirate gently and discard the supernatant without disturbing the bead pellet.
- Repeat the wash using wash buffer 1 twice more.
- Add 5 ml ice-cold DnaK removal buffer to each bead pellet, resuspend beads by gentle shaking.
- Incubate 20 min on a nutator at room temperature.
- Pellet the bead suspension. Aspirate gently and discard the supernatant. Avoid disturbing the bead pellet.
- Add 5 ml ice-cold wash buffer 2 (PBS) to each bead pellet, resuspend beads by gentle shaking.
- Pellet the bead suspension. Aspirate gently and discard the supernatant. Avoid disturbing the bead pellet.
- Repeat the wash using wash buffer 2 three times more.

Elution

- Add 1 ml elution buffer to the bead pellet.
- Transfer bead suspension to a 2 ml eppendorf tube.
- Incubate 1h at 4°C on a nutator shaker.
- Micro-centrifuge protein the samples 190×g for 5 minutes at 4°C.
- Transfer 1 ml supernatant in an ice-cold 2ml eppendorf tube without disturbing the bead pellet. Store at 4°C ("eluted fraction 1").
- Add 1 ml elution buffer to the bead pellet, resuspend beads by tube inversion.
- Micro-centrifuge protein the samples 190×g for 5 minutes at 4°C.
- Transfer 1 ml supernatant in an ice-cold 2ml eppendorf tube without disturbing the bead pellet. Store at 4°C ("eluted fraction 2").
- Repeat elution step once more to get ("eluted fraction 3").

Dialysis

- Measure the amount of purified protein in each fraction using a Bradford protein assay.
- Add the eluted fractions that contain significant protein into a single fraction.
- Transfer the eluted fraction into a 3 ml dialysis cassette according to manufacturer's instructions using a 18 gauge needle.
- Use buoy to hold 1 or 2 dialysis cassettes inside a 1L beaker containing fresh PBS (equilibrated at 4oC) that contains a magnetic stir-bar on top of a magnetic stirrer. Incubate 3h at 4oC while stirring.
- Change the PBS with fresh PBS and incubate 3h more at 4oC while stirring.
- Change the PBS with fresh PBS and incubate overnight more at 4oC while stirring.
- Use a 18-gauge needle to remove the dialyzed protein from the dialysis cassette into a sterile 5 ml syringe. Remove the needle, place a 0.2µm Pall filter, and filter-sterilize the eluted protein extract into ice-cold eppendorf tubes.

Thrombin Cleavage

- For each ml of eluted protein add 10 µl 100 cleavage buffer (200 mM MgCl2, 0.2M TCEP, 2% Tween 20 in PBS) and 25µl thrombin stock solution (0.5 units/µl in PBS; 27-0846-01, G.E. healthcare).
- Incubate 36 hours at 4oC to allow thrombin to cleave the fusion protein.
- Add 1 mM additional TCEP to the cleavage solution 2h before the end of the incubation.
- For each ml of eluted protein, add 500 µl washed glutathione agarose beads.
- Incubating the bead suspension for 3h at 4oC in order to catch the released GST and the uncleaved fusion protein.
- Pelleting the beads by centrifugation (280×g, 3 min, 4oC) and store the supernatant into ice-cold eppendorf tubes.
- Quantify the concentration and purity of the cleaved protein solution via A280 absoption and SDS-PAGE.

Concentration

- If the concentration of the purified cleaved protein solution is not high enough, transfer the solution into a 4 ml Amicon 4 kDa filter centrifugation unit.
- Concentrate the cleaved protein solution by several centrifugation steps of 2200xg for 15 minutes. At the end of each step remove the filter centrifugation units from the centrifuge and use a pipette to mix the retained solution.
    - Careful centrifugation is necessary to avoid protein precipitation in the proximity of the filter
- Repeat the short centrifugation steps until the volume of the cleaved protein solution has reduced enough.
- Remove the cleaved protein solution from the Amicon filter centrifugation unit following the instructions of the manufacturer.
- Use purified proteins in downstream applications.

J.7. Protein Quantification by Immuno-blotting

Jin Guo and Dimitrios Tzeranis

Supplies

- Transfer buffer 10X (BP-100, Boston BioProducts). Store 10x buffer at room temperature. Store 10x buffer at 4oC.
- Tris-buffered Saline with 0.05% Tween-20 (TBST) powder (Sigma 9039-10PAK).
- Instant dry non-fat milk (Carnation, wet lab/room temperature)
- Methanol (wet lab flammable material zoom/room temperature)
- Immobilon PVDF membrane (IPSN07852, Millipore)
- Primary antibody for antigen of interest
- HRP-conjugated secondary antibody against the species of the primary antibody.
- SuperSignal West Femto Chemiluminescent Substrate (Thermo Fisher Scientific #34095, wet lab/ 4 degree freezer), stored at 4oC.
- Bovine Serum Albumin (03116999001, Roche), stored at 4oC.

Equipment

- Mini-PROTEAN Tetra Cell (BioRad 165-8030), (room 3-315).
- Power supply, able to provide up to ~200V and up to ~50 mA (room 3-315).
- FluorChem8900 image acquisition system (Dedon Lab or Grodzinsky Lab).

Procedure

Preparation

- Run the protein samples by SDS-PAGE as described in protocol J.4.
  - After the gel is run do not stain the gel using coomasie blue.
- 1 hour before SDS-PAGE is done i) start soaking the transfer cassette, filter paper, and fiber pads (Mini Trans-Blot Cell) in transfer buffer for 1 hour inside a plastic container in the chemical hood. Remove any air bubbles from the fiber pads, ii) soak the Immobilon membrane in 100% methanol for 15 seconds and let sit in transfer buffer.

Transfer

- When SDS-PAGE run is done, drain the running buffer (it can be disposed in the sink) or store it at 4oC for re-use (use up to 3 times).
- Remove the sticker that connects two plastic covers of the gel. Separate the two plastic covers carefully. The polyacrylamide gel will remain attached to one of the two plastic covers.
- Assemble the blotting sandwich. Place one pre-wetted fiber pad on top of the black side of the open cassette. Add a sheet of filter paper. Carefully add the gel, and remove any air bubbles. Place the membrane on top of the gel. Note that the membrane is cut smaller than the gel, so make sure it covers the appropriate area. Place another sheet of filter paper on top, followed by a second fiber pad. Close the cassette firmly while keeping the contents stable. Use rubber bands to fasten the cassette. Place the blotting sandwich in the transfer module, with the black side of the cassette facing the black plate of the module.
- Place the module in the tank. Fill the tank with transfer buffer so that the whole membrane is submerged. Add a standard stir bar to help maintain even temperature and ion distribution.
- Put on the tank lid and move the Tetra Cell assembly to 4°C cold room or cold box. Place the Tetra Cell assembly on top of a magnetic stirrer. Set the stirrer on a fast speed setting. Plug the transfer cell cables into the power supply.

- Apply a specific amount of voltage or current for a specific time duration. Appropriate conditions depend on the the size of the protein. For proteins whose size is approximately 60 kDa (e.g. SMADs, TGFRs and Tubulin) appropriate seetings are 30V for 12hr or 110V for 1.5hr.
    - Towards the end, prepare 8 ml blocking buffer.
- Disassemble the blotting sandwich in a plastic container in the chemical hood.
- Check if transfer was successful by evaluating if corresponding lanes (of relevant size) of the protein standard have been transferred into the membrane.

Blocking

- Carefully remove the membrane, and use flat-end forceps to transfer it in a small container with enough blocking buffer (either 5% nonfat milk or 3% BSA in TBST) so that membrane is immersed completely.
    - Rinse the transfer cell, fiber pads, and cassettes well with deionized water.
- Incubate the membrane with rocking for 1.5 h at room temperature or overnight at 4 deg.

Immuno-blotting

- Prepare antibody diluent: 1 % BSA, 0.05% Tween 20 in TBS or 5% Milk, 0.05% TBST. Store at 4oC.
- Prepare 150 ml wash buffer (0.05% Tween 20 in TBS) per membrane.
- Prepare primary antibody solution by diluting the stoc antibody in antibody diluent. Dilution depends on the protein and the protocol.
- When membrane blocking is done, rinse the membrane 4 times with 10 mL wash buffer.
- Add 2 ml primary antibody working solution (dilution ration, see Table 1) to each membrane in a small plastic container.
- Incubate the membrane with shaking overnight at 4°C.
- Towards the end of the primary antibody incubation, prepare 2 ml secondary antibody solution per membrane, by diluting stock antibody solutions with antibody diluent.
- Aspirate the primary antibody solution, and wash each membrane four times with 10 ml wash buffer for 10 minutes in a plastic container
    - When immunoblotting takes place at 4oC, the primary antibody solutions can be stored at 4°C for future use (within a few days).
- Add 2 ml secondary antibody solution per membrane and incubate 1 hour at room temperature or overnight at 4 deg.
- Aspirate the secondary antibody solution.
- Wash each membrane 4 times with 10 ml wash buffer for 10 minutes.

Chemiluminescence-based imaging

- Before staining and imaging, make sure that the imager (FluorChem 8900) is available. Log into the FluorChem 8900 software and use an old blot to focus the CCD.
- Stain and image each membrane by itself so that results are consistent:
- In a dark room, prepare 500 ul chemiluminescence solution by adding 250 ul luminal and 250 ul oxidizing solution
- Use flat-end forceps to place the membrane between two plastic sheets. Place the sandwich over a kimwipe, and use the kimwipe to aspirate the extra wash buffer
- Pipette 500 ul chemiluminescence solution on the membrane, use a pipette tip to smear the solution homogeneous.
- Remove air bubbles, incubate membrane for 2 min in the dark room, and then transfer the system to a FluorChem8900 system.
- Place the membrane on the FluorChem8900 imager, use the white board inside the machine.
- Image the membrane with full open iris, filter set 1, and appropriate exposure time

(depends on the signal level so that the signal is good and it does not saturate).

- Make sure the membrane is always on focus when take imaging.
- Process images using FluorChem's software or ImageJ.

Antigen	Clone	Vendor	Catalog #	Storage	Description	Size [kDa]	WB dilutions
ALK1 (ACVRL1)		Abgent	AP7807a	4°C	Rabbit pAb H,M 0.25 mg/ml	56	Prim: 1:250 Sec: 1:1000
ALK5 (TGFBR1)	V-22	Santa Cruz	Sc-398	4°C	Rabbit pAb IgG H,M 0.2 mg/ml	53	Prim: 1:500 Sec: 1:4000
SMAD1	D59D7	Cell Signaling	6944	-20°C	Rabbit mAb H,M Incubate membrane with diluted antibody in 5% w/v nonfat dry milk	60	Prim: 1:1000 Sec: 1:2000
SMAD2	D43B4	Cell signaling	5339S	-20°C	Rabbit mAb H,M,R 110ug/ml	55-60	Prim: 1:1000 Sec: 1:4000
SMAD3	C67H9	Cell signaling	9523S	-20°C	Rabbit mAb H,M,R 75ug/ml	52	Prim: 1:500 Sec: 1:2000
SMAD5	Q-20	Santa Cruz	sc-26418	4°C	Goat pAb IgG H,M,R 0.2mg/ml	52	Prim: 1:200 Sec: 1:10000
P-Smad2 (S465/467)	138D4	Cell signaling	3108	-20°C	Rabbit mAb H,M,R	60	Prim: 1:1E3 Sec: 1:1E3
P-Smad1/5 (S463/S465)	41D10	Cell signaling	9516	-20°C	Rabbit mAb H,M,R	60	Prim: 1:1E3 Sec: 1:1E3
Tubulin		Abcam	ab7291	-20°C	Mouse anti human	50	Prim: 1:1E4 Sec: 1:4E4
αSMA		Sigma	A2547	-20°C	Mouse mAb Aliquoted 1:10		Prim: 1:500 Sec: 1:1E5
Secondary anti goat		Santa Cruz	sc-2020	4°C			
Secondary anti rabbit		Cell signaling	7074	-20°C			
Secondary anti mouse		Pierce	31430	-20°C			

Table J.7-4: Protocol parameters for immune-blotting protein quantification.

J.8. Circular Dichroism Spectroscopy Standard Operating Procedure

Provided by Debby Pheasant
(Biophysical Instrumentation Facility, MIT)

- Protein solution should be at A280 < 1.0 in an appropriate buffer. Check the chart in Notes from Aviv to see which buffer components have absorbance which could interfere. For lengthy scans, solutions should be degassed or sparged with argon (30 min. with stirring). The cells* hold 300ul of solution. Run a buffer blank as well as your sample and subtract the blank from your run.
- To prepare the instrument, turn on the house nitrogen (5 valves) and check the flow rate by looking at the height of the silver beads in the air valve windows. Insufficient flow may indicate a leak. Allow the nitrogen to flush through the instrument for 15 minutes. It is important to purge the lines since oxygen/ozone will etch the lenses and mirrors in the CD.
- After 15 minutes of nitrogen flushing, turn on the "lamp power" switch. (left black switch on the power box, down low) Wait for the "Lamp Ready" light to come on, wait 1 minute then push the red button: "Push to Start". If the lamp lights, you will see the green "Lamp On" light come on. If the lamp doesn't light, try again. Record the number of times it took to start the lamp in the log book under "# starts". Let the lamp warm up for at least 45 minutes.
- After the lamp is warmed up, turn on the "CPU & instrument" switch. (right black switch on power box) Then turn on the computer with the button on the front of the HP tower. Log on as "Aviv user".
- Before opening the Aviv software, make a folder for yourself in the DATA folder on the C: drive.
- Open the Aviv V 320b software. With this software you must always hit "enter" on the keyboard then click "return" or "okay" on the screen to save a value. The Data Collection window will open and initialize. To set your data path, open Displays then Data Browser. Set your data path in the Default Dataset Path window. (ex: C:\DATA\Debby) Leave the cursor in that window, hit enter, then click the Return button. Now go to Configure Experiment. Give a description and name to your experiment then choose the type it will be: either Wavelength scan or Temperature scan. (You can do both at the same time but you need a macro).
- Go to Experiment Configuration and under Wavelength Configuration, enter starting and ending wavelengths (260-195nm is a typical range) and change sampling parameters if you wish. The WL Step Scan is the traditional scan. (The WL Schedule is to quickly get values at wavelengths which you specify. It will, however, draw straight lines between your points.) For Temperature Experiment Configuration, enter starting and ending temperatures for a scan (range is -10° to 110° C). Enter your preferred temperature steps and averaging time. Temperature Step is the traditional scan and Temperature Schedule is like WL Schedule.
- With everything set, go to Save Data Options and choose CD signal and dynode. Always check the second box in the Experiment Data section. Now choose Experiment Configuration and Set Experiment Counter. Set Counter to 1. Close the Configure Experiment window by choosing Exit/Save Configuration. Go to Control Panels and set Temperature Control Parameters. (For a wavelength scan, set the equilibration time to zero).
- To start the experiment, hit the Run Experiment button. Keep your eye on the dynode level as the scan moves to lower wavelengths. If it gets to 400V above your baseline, stop the experiment because something is absorbing light and you won't get good CD data.

- When your experiment is over, choose to have your data go to the Data Browser and Hard Disk. Fill in the log book before turning off lamp. Read the lamp hours off the black dial. Read max. dynode and wavelength range off the screen. To determine the air dynode at 200nm, remove the cuvette, change the wavelength under Monochromator to 200nm, then click Return. When the wavelength has changed, read dynode level in the CD-PMT box and record that number in the log book. This helps us assess the lamp's lifetime.
- If you did a temperature scan, go back to the Control Panel and remove the equilibration time. Close the software before turning off the lamp so the instrument can return to its home wavelength of 543.5 nm. Now turn off the lamp with the black switch. Let the nitrogen flush the instrument for 5 minutes then turn off the nitrogen, closing valves in the reverse order from turning them on. You can save your data to a CD, flash drive or email it to yourself. Then turn off the computer at the screen. After it has shut down completely, turn off the black "CPU & instrument" switch.

Appendix K: Fluorescence Labeling

Contents

K.1. Preparation of 4% Paraformaldehyde in PBS

Supplies

- Sterile Phosphate Buffered Saline (PBS, 20012-043, Invitrogen, Inc., Carlsbad, CA).
- Paraformaldehyde (P6148, Sigma Aldrich), stored at 4°C. TOXIC!

Equipment

- Magnetic stirrer
- Chemical Hood

Procedure

- Weight and transfer 800 mg paraformaldehyde in a 50 ml beaker.
- Perform the remaining steps in a chemical hood.
- Add 20 ml PBS, cover the conical beaker with parafilm.
- Inside the chemical hood, stir and heat (55 to 65°C) until the solution becomes clear
    - make sure solution does not boil! Otherwise PFA is useless
- Let the solution cool to room temperature.
- Store at 4°C until use.

K.2. Fluorescence Staining of Nerve Samples

M. Buydash and Dimitrios Tzeranis

Supplies

- Wheat Germ Agglutinin conjugated to alexa fluor 488 (W11261, invitrogen).
- Nerve samples: 6 µm thick sections of rat sciatic nerves, embedded in OCT, stored at -20°C.
- Hydrophobic pen, stored at 4°C
- Tris Buffered Saline with 0.05% tween20 (TBST)
- Tris Buffered Saline (TBS)
- DAKO protein block, stored at 4°C
- DAKO antibody diluent, stored at 4°C
- Triton X100 (TX100)
- S100 primary antibody (ab76729, Abcam)
- α-SMA primary antibody (A2547, Sigma Aldrich).
- NF200 primary antibody (ab72996, Abcam).
- Mouse IgG1 control antibody (sc3877, Santa Cruz).
- rabbit IgG control antibody (sc2027, Santa Cruz).
- Donkey anti-rabbit alexa488 secondary antibody (Jackson immunolabs)
- Donkey anti-rabbit alexa350 secondary antibody (Invitrogen)
- Donkey anti-mouse alexa488 secondary antibody (Jackson immunolabs)
- Donkey anti-mouse alexa350 secondary antibody (Invitrogen)
- Donkey anti-chicken TRITC secondary antibody (Jackson immunolabs)
- DAPI (D1306, Invitrogen)
- Rhodamine phalloidin (R415, Invitrogen)
- Phalloidin alexa fluor 532 (A22282, Invitrogen)
- Mounting medium (Fluoro-Gel)
- #1.5 coverslips

Equipment

- Immunofluorescence microscope (Van Vliet Lab)
- Spectral multi-photon microscope (So lab).

Procedure

- Thaw samples by incubating at room temperature for 30 min.
- Place samples inside a microscope glass holder filled with TBST and shake gently two times for 10 min.
- Prepare 80 µl blocking buffer per section.
  - DAKO protein block, 0.3% Triton X100: 0.24 µl TX100 80 µl DAKO protein block.
- Use a folded kimwipe to remove TBST around and between nerve sections.
- Add TBST if necessary to prevent drying.
- Use a hydrophobic pen to isolate each tissue section (on the dried part of the glass).
- Aspirate remaining TBST from the sections and add 75 µl blocking buffer per section.
- Incubate for 60 min at room temperature.
- During the last part of the blocking incubation prepare 60 µl primary antibody or control antibody solution per section.
  - Dilute the antibodies in DAKO antibody diluent.
  - The control antibody should be from the same host and have the same antibody isoform as the primary antibody.

- The control antibody solution should contain the same antibody density (µg/ml) as the primary antibody solution.
- Micro-centrifuge primary antibody solutions at 14000rpm for 40 seconds to pellet antibody aggregates.
- Add 50 µl primary/control antibody solution per section.
  - Use one tip for each group of sections that contain the same primary/control antibody.
- Incubate the sections with the primary/control antibody solutions overnight at 4oC.
- Aspirate the primary/control antibody solutions.
- Wash the sections three times for 10 minutes each using 80-100 µl TBS.
  - Apply the following steps in the dark, using a red lamp to prevent photo-bleaching.
- During the last two washes, prepare 60 µl secondary antibody solution per section.
  - Dilute the secondary antibody in DAKO antibody diluent.
  - Micro-centrifuge the secondary antibody solutions at 14000rpm for 40 seconds to pellet antibody chunks.
- After the last wash, add 50 µl secondary antibody solution per section.
- Incubate the samples for 2 hours at room temperature in the dark.
- Toward the end of the secondary antibody incubation, prepare 60 µl counter-staining solution per section.
- Aspirate the secondary antibody solution from each section.
- Wash the sections two times for 10 minutes each using 80-100 µl TBS.
- Add 50 µl counterstain solution per section.
- Incubate for 30 min at room temperature in the dark.
- Aspirate the counterstain solution and dispose according to EHS regulations.
- Wash the sections three times with 80 µl TBS for 10 min each. Dispose of washes according to EHS regulations.
- During the washes, prepare coverslips.
  - Wipe a #1.5 coverslip with 70% ethanol and dry thoroughly.
  - Note the location of the sections with respect to the coverslip (when the coverslip does not cover the white part of the microscope glass).
  - Place a drop of mounting medium inside an eppendorf tube. Transfer 4 µl mounting medium on the coverslip at locations corresponding to the sections.
  - ! ! Avoid bubbles.
- After the last wash, slowly place the coverslip on top of the samples without trapping bubbles between the section and the coverslip.
- Seal the sample and the coverslip using nail hardener.
- Image the samples no sooner than 15 min to let the hardener solidify.
- Image samples using an epi-fluorescence microscope (van Vliet lab) or a multi-photon microscope (So lab).

K.3. Immunofluorescence staining of cell-seeded collagen scaffolds

M. Buydash and Dimitrios Tzeranis

Supplies

- CMTMR (C2927, Invitrogen), aliquoted in 10 mM samples (in DMSO) stored at -20°C.
- DMEM (11965-092, Gibco)
- FBS (16000-077, Gibco) stored at -20°C.
- Penicillin/Streptomycin (5140, Gibco), stored at -20°C.
- Primary and secondary antibodies (see Table K3-1)
- BSA (3116999001 Roche), stored at 4°C.
- Triton X100 (93427 Fluka)
- PBS powder (P3813-10PAK, Sigma Aldrich)
- Paraformaldehyde (P6148-500G, Sigma Aldrich), stored at 4°C.
- Methanol.
- Hoechst33342 (H3570), stored at -20°C.
- SYTO82 (NE47-235), stored at 4°C.

Equipment

- Immunofluorescence microscope (Van Vliet Lab)
- Spectral multi-photon microscope (So lab).

Procedure

This protocol described the immunofluorescent staining of cells seeded inside collagen-GAG scaffolds (CGS). Usually 40000 fibroblasts are seeded in 5mm-diameter porous collagen scaffolds, are treated according to some experimental design, and then are fixed and stained.

Stain cells with a cell-tracker dye

- Use sterile forceps to transfer each cell-seeded scaffold inside a well of a 96-well plate.
- Prepare the cell-tracker staining solution (25 uM CMTMR (emission peak 565nm) or 25 uM CMFDA (emission peak 517nm) or 25 mM CMRA (emission peak 576nm) in DMEM) by diluting 10 mM CMTMR/CMFDA/CMRA 1:400 in serum-free DMEM.
- Add 200ul 25uM cell-tracker staining solution into scaffold in a well.
- Incubate the plate in a sterile hydrated cell incubator (37oC, 5% CO2) for45min.
- Aspirate and dispose the cell-tracker staining solution. Add 200ul complete culture medium (DMEM supplemented with 10% FBS and 1%P/S) per well.
- Incubate cell-seeded scaffolds in a sterile incubator (37oC, 5% CO2) for45min.
- Aspirate and dispose the complete culture medium. Add 200ul PBS per well.
- Wash twice the scaffold by gently shaking the plate for 10min. Use the big orbital shaker (shaking speed should be between 4 and 5).
    - !! Avoid aggressive shaking or vortexing because it causes cell detachment.

Fixation

- Two possible fixation methods:
    - Methanol protocol: Add 200 ul ice-cold methanol (stored at -20oC) per well. Shake gently for 7 min at room temperature.
    - Paraformaldehyde protocol: Add 200 ul 4% paraformaldehyde (stored at 4oC) per well. Shake gently for 15 min at room temperature.
- Paraformaldehyde (PFA) fixation is the preferred protocol for preparation as it preserves cells better.
- Aspirate and dispose the fixation medium (methanol/PFA) by EHS specifications.

- Add 200 ul PBS per well, shake gently for 10 min at room temperature.
- Aspirate and dispose PBS.
- Repeat the 10-min PBS wash two more times.

Immunofluorescence Staining

- Add 200 ul permeability buffer (0.25% TritonX100 in PBS, prepared by adding TritonX100 to PBS, stored at 4 oC refrigerator) per well, shake gently for 13 min at room temperature.
- Aspirate and dispose permeability buffer.
- Add 200 ul permeability buffer per well, gently shaking for 5 min.
- Aspirate and dispose the permeability buffer.
- Wash samples with 200 uL PBS for 10 min, shake gently for 3 times at room temperature.
- Add 200 uL blocking buffer (5% BSA in 0.25% PBST). Incubate at room temperature for 1.5 hour while shaking gently.
- During sample blocking, prepare the primary antibody solutions. Dilute primary antibody stock solutions in antibody diluent (1% BSA in PBST), see Table K3-1.
- Aspirate blocking solution. Add 150~200ul primary antibody solution per sample.
- Incubate at 4°C (NE47, cold room) overnight while shaking gently.
- The following day, aspirate the primary antibody solution.
- Do three washes with 200ul PBS (10 min, room temperature, shake gently).
- During the three PBS washes prepare the secondary antibody solutions. Dilute secondary antibody stock solutions 1:200 in antibody diluent (1% BSA in PBST), see Table K3-1.
- After the last wash, aspirate PBS and add 150~200ul secondary antibody solution per sample. Incubate samples at RT for 2h, shake gently and protect samples from light.
- Do three washes with 200ul PBS (10 min, room temperature, shake gently).

Antigen	Clone	Antigen Size	Vendor	Catalog #	Storage	Info	Prim. ab dilution	Sec. ab dilution
Smad1	D59D7	60 kDa	Cell signaling	6944	-20°C	Rabbit IgG, mAb	1:200	1:200
SMAD2	D59D7	55-60 kDa	Cell signaling	5339S	-20°C	Rabbit IgG, mAb 110ug/ml	1:50	1:200
SMAD3	C67H9	52 kDa	Cell signaling	9523S	-20°C	Rabbit IgG, mAb 75ug/ml	1:50	1:200
SMAD5	Q-20	52kDa	Santa Cruz biotech	sc-26418	4°C	Goat IgG, pAb 200 µg/ml	1:100	1:200
P-Smad1/5 (S463/S465)	41D10	52kDa	Cell signaling	9516	-20°C	Rabbit IgG, mAb	1:50	1:200
Tubulin	DM1A	50 kDa	Abcam	ab7291	-20°C Aliquoted	Mouse IgG, mAb 1 mg/ml	1:500	1:200
αSMA	1A4	42 kDa	Sigma aldrich	A2547	-20°C Aliquoted 1:10 dilution	Mouse IgG2a, mAb	1:400	1:200

Table K3-5: Information for immune-labeling of cell-seeded scaffolds.

Counterstaining

- Prepare counterstaining solution:
  - Nucleic acid stain: Use either 1:8E5 Hoechst33342 diluted in PBS, or 1:3E4 SYTO82 diluted in PBS. Use a fresh aliquot.
  - Actin stain: dilute 1:100 phaloidin-TRITC.
- Add 200ul counterstaining solution per well.
- Incubate samples 30 min for 37oC, shake gently.
- Wash samples three times with 200ul PBS, 10 min, at room temperature.

Sample Preparation and Storage

- Prepare the scaffold imaging housing: Use a Press-to-Seal™ silicone isolator, tear off the transparent membrane (red side) of the isolator first, press a #1.5 coverslip onto the isolator, flip the coverslip-isolator and tear off the white membrane of the isolator.
- Transport 2~6 scaffolds on the coverslip. Press a glass slide onto the coverslip-isolator and generate the "sandwich" coverslip-isolator0glass slide housing.
- Store the samples in a white cryo-box (NE47-216, 4oC refrigerator). Place a tube of water inside the box to keep the box humidied.
- Image the scaffolds the same or the following day.

Imaging

- Image samples using an epi-fluorescence microscope (van Vliet lab) or a multi-photon microscope (So lab).

Appendix L: Standard Operating Procedure for the So Lab "Original" Spectral Multi-Photon Microscope

The various systems of the "original" multi-photon microscope are described in Appendix A. The microscope is able to acquire 2D or 3D images:

- 2D: acquire the image of a plane, whose field of view depends on the objective magnification and the motion range of the scanning mirrors (see Table L1-1). In order to image an area that is larger than the field of view, it is necessary to use the prior xy stage to translate the sample and acquire multiple images.
- 3D: acquire a "z-stack", which is a sequence of parallel planes that are separated by some constant axial spacing Δz. The acquired volume is limited in the transverse plane by the field of view, and in the axial plane by the travel range of the piezoelectric actuator (100 μm).

	scan mirror motion range					
Objective magnification	32×	16×	8×	4×	2×	1×
40×	110×110μm	55×55μm	28×28μm	14×14μm	7×7μm	3.5×3.5μm
20×	220×220μm	110×110μm	55×55μm	28×28μm	14×14μm	7×7μm
10×	440×440μm	220×220μm	110×110μm	55×55μm	28×28μm	14×14μm

Table L1-6: Field of view of a single image as a function of the objective magnification, and the scanning mirror tilt motion range.

L1. The "Confocal" Microscope Control GUI

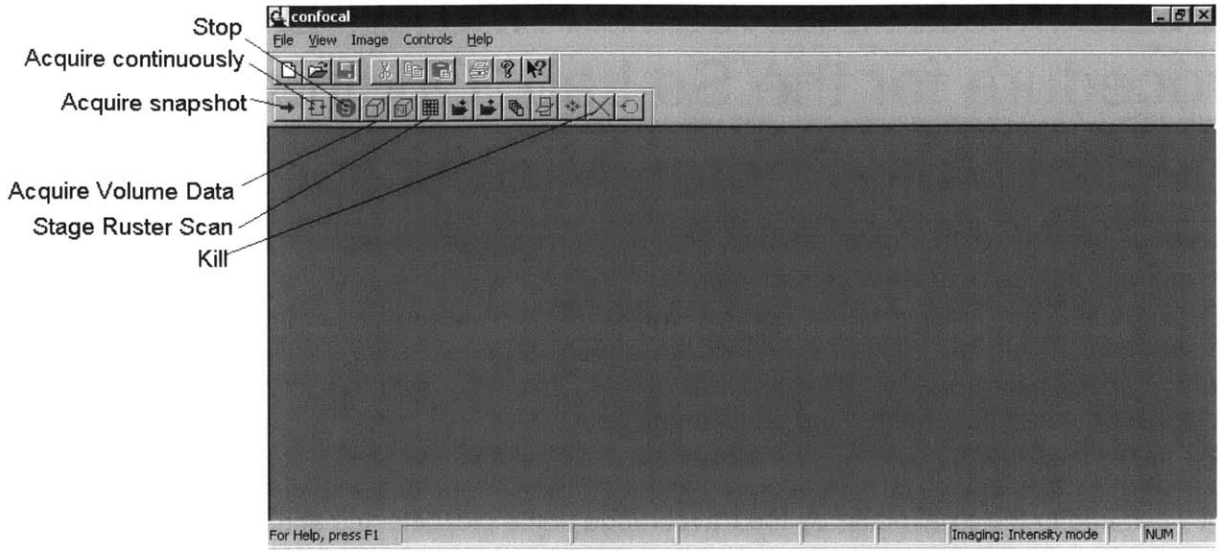
"Confocal" is the GUI application that controls the microscope and acquires data. The confocal application can be loaded using a shortcut located on the computer desktop that controls the "original" microscope.

There are two versions of the confocal application.

- 16-channel version (called "confocal16") acquires data from the 16-channel MA-PMT R5900P-00-L16.
- 1-channel version (called "confocal4") acquires intensity imaging from any single-channel PMT (R7700P, H7421-40, H7421-50).

The C++ code of each version, together with the compiled application, is located inside a folder called "confocal16" and "confocal4" respectively. The folder of the appropriate application needs to be renamed "confocal" and placed in the "code" folder. A shortcut to this folder is located on the desktop. Obviously only one of the two confocal versions can be located in this directory. The version not in use, is usually stored on the desktop. Care is needed to avoid confusing the two folders that contain the two versions of the confocal application.

Main screen of confocal application



- Acquire snapshot: acquire a single image and display it on the screen
- Acquire continuously: acquire images of a plane repeatedly and display the latest one. Continuous acquisition stops by clicking the stop button, and is done when the "Data acquisition" message in the bottom of the screen disappears.
- Stop: stop continuous image acquisition.
- Stage Raster Scan: setup the x-y stage and z-piezo for 3D image acquisition.
- Acquire Volume Data: acquire a 3D image. Click this after clicking Stage Raster Scan.
- Kill: Stop immediately any image acquisition. Use only in urgent conditions, especially when the "FAIL" message appears in the bottom of the screen. Avoid using this button if possible!!

Hardware Settings Menu

Image acquisition parameters are chosen in a set of tabs that appears by clicking on the Controls menu, and then on Hardware Settings. The key fields are described below:

- Acquisition Tab:
  - Origin X/Y: defines the initial position of the XY scanning mirror during the scanning of a plane. Units are analogous to angular position of the scanning mirrors.
  - Origin Z: In 2D imaging provides the constant position of the z-piezo. In 3D imaging, provides the z position of the first plane of the z-stack. Expressed in units of 10 nm (e.g. 10 means 100 nm).
  - Steps X/Y: DO NOT CHANGE this field (the application will crash).
  - Steps Z: In 3D imaging provides the number of steps along the Z axis of the z-stack.
  - Resolution XY: Controls the scanning mirror tilt motion range (Table 1). Its value can be 32, 16, 8, 4, 2 or 1. The peak-to-peak motion of the mirror equals 8000 angle units (the units used in the Origin X/Y field).

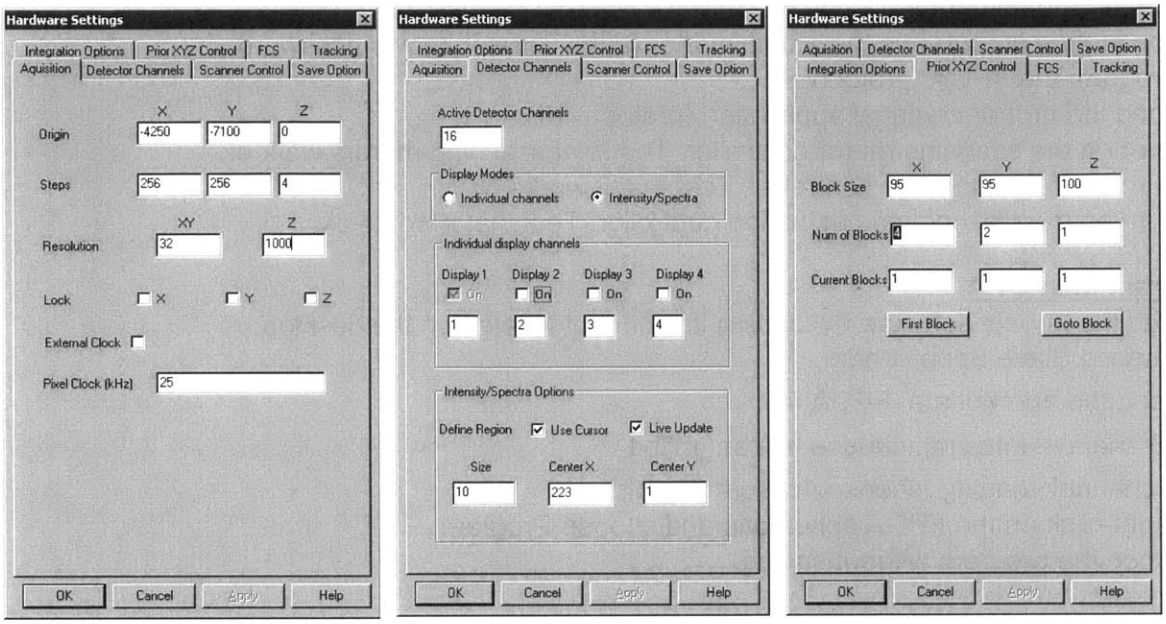
    Motion range [angle units] = 250*Res

    Controls the field of view of each image (also depends on objective magnification), which is approximately (see Table 1)

    Field of view (um) = 275/2*Res/MAG
  - Resolution Z: Used in 3D imaging. Defines distance between neighboring planes of the z-stack. Expressed in units of 10 nm (e.g. 10 means 100 nm).

- Lock X, Y, Z: Makes the position of the scanning mirrors and the z-piezo to be constant and equal to the position defined by the Origin X/Y and Origin Z fields respectively. Used in Fluorescence Correlation Spectroscopy (FCS).
- Pixel Clock: Defines the pixel sampling frequency. It can be chosen in the [0.5, 25] kHz range. Equals the inverse of the sampling time per pixel in msec (25kHz corresponds to 40 μsec per pixel). Larger values correspond to faster sampling, faster acquisition, but less signal (photons per pixel) and SNR.

- Detector Channels Tab:
  - Make sure to select the "Intensity/Spectra" field when in 16-channel detection.



- Prior XYZ Control Tab:

- Block Size field X/Y: Defines the spacing [in microns] between neighboring images (2D imaging) or z-stacks (3D imaging).
- Num of Blocks field X/Y : Defines the numbers of steps along the X/Y axis.
  - The three fields of the Z column are of no use in the particular stage used.

L2. Safety Precautions

- WEAR SAFETY GLASSES
- Fill log-in paper when you start using the microscope (date, time, wavelength, beam power, pump current, name)
- BLOCK THE LASER OUTPUT EVERY TIME THE BEAM PATH IS MODIFIED.
- MAKE SURE THE LASER BEAM PATH IS KNOWN.
- FOLLOW INSTRUCTIONS OF MIT LASER SAFETY CLASS.
- KEEP ROOM LIGHTS OFF WHEN A PMT IS ON!

L3. Prepare the System for Imaging

Usually all the instrument components are turned on and the instrument is ready-to-use. In this case the user just needs to do a fine alignment before acquiring experimental data. If it is necessary to restart the computer, then the user needs to restart, program and check various system components. In order to switch between 16-channel and 1-channel imaging, then the user needs to reprogram the FPGA and choose the appropriate confocal application.

Turn on system components

- Check that the LP power supply is on and provides voltage to electronics.
  - The 16-channel photon counting card operates at 13.4V and consumes approximately 1.1A.
  - The voltage regulator board operates at 13.7V.
- Verify that any high voltage power supply for the PMTs do not provide voltage to PMTs.
- Turn off the scanning mirror controller, prior stage controller and piezoelectric actuation controller.
- Restart the computer.
- Log in the computer as the administrator (administrator knows password).
- Program the FPGA (protocol 9).
- Load the proper confocal application (protocol 10).
- Turn on the scanning mirror controller. Test that scanning mirrors work ok.
- Turn on the prior stage controller. Test that it works ok.
- Turn on the piezoelectric actuation controller. Test that it works ok.

Program the FGPA

- Run the Impact software by loading its shortcut located on the desktop.
- Choose Slave Serial Mode.
  - Load the appropriate FPGA file:
  - 16-channel imaging: choose cscan_ml.bit
  - 1-channel imaging: choose 4chscan2sp.bit.
- Right-click on the FPGA schematic and choose Program.
- Close the program without saving.

Run the Proper Confocal Application

- There are two "Confocal" applications: the 16-channel and the 1-channel. The source code and the compiled executable of each application is located inside a folder.
- The folder that contains the appropriate confocal application needs to be renamed "conf" and be copied in the "code" directory. There is a shortcut to this directory on the PC desktop.
- Make sure not to delete the confocal applications as you switch between 1-channel and 16-channel imaging.
- Run the confocal application by running the "confocal" shortcut located on the desktop.
  - The confocal application runs only in 8-bit windows color mode.
  - Maximize the confocal application window (to avoid a potential C++ bug).
  - In 16-channel imaging: Select the Intensity/Spectra mode on the Detector Channels tab of the Hardware Settings (Controls menu).

Turn on and check the x-y stage and the z-piezo actuator

- Turn on the prior stage controller.
- Acquire a 2×4×1 x-y-z 3D image with (95,95,0 μm) spacing (see Protocol XX)
  - 1 mm prior motion corresponds approximately to a full rotation of the prior stage knobs.
  - Check that the knob returns to its initial position.

- The stage sometimes does not perform the XY raster motion as asked. In this case, do a power cycle (turn off then turn on the prior stage controller) and repeat 2×4×1 acquisition.
- Turn on the piezoelectric actuator controller.
- Acquire a 1×1×4 x-y-z 3D image at (95,95,10 µm) spacing (see Protocol XX).
  - This command acquires a z-stack of 4 planes that are spaced 10 µm apart.
  - When the piezoelectric actuator operates normally, a "tick" noise can be heard every time the piezo translates the objective.
  - If the piezo does not perform task correctly, it is usually a bad cable connection (the cable that controls the piezo at the point where it enters the piezo).
- Run a 1×2×4 XYZ run at (95,95,10 µm) spacing (see Protocol XX).
  - This step verifies that piezo electric and the x-y stage operate correctly simultaneously.
- If the instruments do not perform the task correctly, do a power cycle on the prior stage controller and repeat this 3D image acquisition.

Fine tune microscope alignment

The microscope is already aligned by the administrator. However it is possible to check and fine-tune this alignment.

L4. Imaging

Sample preparation

- Use coverslip #1.5 for best performance.
- Very thin samples are sandwiched between a microscope glass and a coverslip.
- Thick samples can be positioned between the microscope glass and a coverslip, separated by a silicone spacer.

Prepare the objective

- Clean objective using a lens cleaning paper and spectroscopic-grade methanol (ask administrator to show you how if you do not know).
- Add a drop of immersion medium (water or oil) on the objective lens.
  - Make sure no bubbles are trapped in the immersion medium.
  - ASK ADMINISTRATOR BEFORE USING OIL IMMERSION!

Position the sample on the microscope

- Use the z-positioning knob to move the objective a bit away from the sample.
- Place and secure the microscope glass on its proper spring-secured position.
- Use the z-positioning knob to move the objective closer to the sample until the immersion medium makes contact with the coverslip.
- CHECK THAT IMMERSION MEDIUM IS IN CONTACT WITH THE COVERSLIP AND NOT THE MICROSCOPE GLASS. Imaging cannot take place through microscope glass.
- Do not contaminate the immersion water with fluorescent dye solutions.
- Make sure the coverslip is clean! Clean with 70% ethanol if necessary.
- The sample to be imaged must make contact with (or be very close to) the coverslip, which must make contact with the immersion medium.

Focus on the sample

- Verify that the excitation laser shutter is closed.

- Verify that the laser beam is not blocked in the output of the ti-sapphire laser.
- Before imaging it is necessary to manually focus on the sample.
- BE VERY CAREFUL DURING FOCUSING. AVOID HAVING THE COVERSLIP TOUCH THE OBJECTIVE BECAUSE THIS CAN DAMAGE THE OBJECTIVE
- Set the microscope sliding filter-set housing at its left position (select dichroic D2) and guide the emission ligh into the eyepiece (M11 in path, M14 out of path).
- Check the emission path: turning off PMT power supplies, shine some light into the objective, verify that light reaches the eyepiece.
- MAKE SURE PMTs are OFF when shining light into an objective.
- Use the z-positioning knob to move SLOWLY the objective closer to the sample until the sample is on focus.
- Focusing is sample-dependent:
- Focusing on solid piece samples (e.g. a piece of tissue, collagen fibers) is easy. When the object is in focus it will appear sharp on the eyepiece.
- Focusing on liquid solutions can be done by exciting the solution with a low power laser beam (2 mW) and simultaneously aquire images continuously (Acquire continuously button). When the laser focuses inside the solution, a sharp fluorescent line will appear in the eyepiece.
- Make sure the filter F2 is on place and wear IR-blocking goggles.
- Focusing on cells on a petri dish is harder and requires some training.
- Focusing on very thin samples (e.g. bacteria on a coverslip) is even harder.

Image acquisition

- After focusing on the sample, images can be acquired.
- Place a "imaging on, do not turn lights off" sign on the lab door. This will prevent other lab members from turning on the room's lights when the PMTs are on.
- Place the power meter head after the polarizer A1 and set its offset to zero.
- Open the shutter (at the moment the shutter is located right after A1) and measure the laser beam power.
- Select the proper laser beam power by adjusting the roll position of the half-wave plate H1.
- Close the shutter and remove the power meter head out of the beam path.
- Make sure the microscope light is off.
- Turn off the microscope light before changing the emission path. This will prevent light from shining on the PMTs, which causes PMT degradation.
- Change the emission light path.
  - For de-scanned configuration imaging (16-channel, FLIM, 1-channel) Set the microscope sliding filter-set housing at its center position (select mirror M5).
  - For scanned configuration imaging (1-channel, FCS). Set the microscope sliding filter-set housing at its left position (select dichroic D2). Get the mirror M11 out of beam path (beam goes into bottom PMT).
- Select imaging conditions in the Acquisition Tab of the Hardware Settings in the Tools menu.
- See the "Imaging Overview" and the "The "Confocal" Microscope Control GUI" sections above for description.
- Origin X/Y: The value of the fields OriginX and OriginY should depend on i) the values OriginX0 and OriginY0 that are chosen by the administrator during full microscope

alignment, and ii) the field of view (parameter ResX/Y) as OriginX= OriginX0 – 125*ResXY and OriginY= OriginY0 – 125*ResXY

- MAKE SURE ALL LIGHTS I THE ROOM ARE OFF.
- Turn on the high voltage power supply of the PMT of interest.
- Open the shutter.
- Acquire the image of interest:
  - In order to acquire a single snap of a plane click the Acquire snapshot button.
    - The image can be then saved, through the save button.
    - When an image is acquired, a message appears in the bottom of the "confocal" GUI window.
  - In order to acquire images continuously, click the Acquire continuously button. When it is necessary to stop the acquisition, click the Stop button.
  - In order to use the piezoelectric actuator and acquire a z-stack, select the z-stack start (Origin Z), the number of steps (Steps Z) and the z step size (Resolution Z) in the acquisition tab of the Acquisition Tab of the Hardware Settings command menu. Then click the Stage Raster Scan button and then click the Acquire Volume Data button.
    - The image can be then saved, through the save button.
  - In order to use the stage and the piezoelectric actuator for raster-scanning, select the z-stack start (Origin Z), the number of steps (Steps Z) and the z step size (Resolution Z) in the acquisition tab of the Acquisition Tab of the Hardware Settings command menu. Then select the number of X and Y steps (Number of Blocks X/Y) and the size of X and Y steps (BlockSize X/Y) in the Prior XYZ Control tab of the Hardware Settings command menu. Then click the Stage Raster Scan button and then click the Acquire Volume Data button.
    - Raster-scanning data can only be saved as soon as they are acquired. Before clicking the Stage Raster Scan and Acquire Volume Data buttons, go to the Save Option tab of the Hardware Settings command menu, add the directory path where data will be saved, the prefix of the saved files and click to enable the two save options.
- Be very patient with the confocal GUI. It is always better to wait until image acquisition is done. If the program stalls, one can try to use the Kill button on the GUI. After using the Kill button it is better to check all the parameters in the Hardware Settings command menu and to test that the stage works ok.
- In the case where it is necessary to force quit the confocal GUI, it is advisable to restart and follow the protocols described in part II.

Turn off microscope (everyday use)

- Block the output of the Ti-Sapph laser using the laser blocker.
- Ensure that all PMT high-voltage power supplies are turned off.
- Remove the sample from the sample holder on the stage.
- Place a kimwipe in contact with the water immersion medium. The kimwipe will aspirate most of the immersion medium.
  - Do NOT use the kimwipe to clean the objective; the kimwipe can scratch the objective.
- Clean the objective using lens cleaning paper and spectroscopic-grade methanol (ask administrator to show you how).
- Turn off room lights.
- Remove the "imaging on, do not turn lights off" from the lab door.
- If you are the last user for the day and someone plans to use the laser the following day, reduce the laser power to 8W. If none plans to use the Ti-Sapph the same or the following day, turn off the laser

Turn off Ti-Sapphire laser

- Press the P1 button to set green laser power level to 0.2W
- When the green laser output reaches 0.2W, then press the power button and the laser turns off completely.

Appendix M: Image Processing Pipeline Implementation in MATLAB

Overview

The image processing pipeline described in Chapter 2 is implemented in MATLAB. Due to the large number of data files that need to be processed, the large number of image processing functions involved, and the requirement for non-experts (in optics and image processing) to be able to use the pipeline, a graphical user interface (GUI) was developed that calls various functions that implement different parts of the pipeline.

The software consists of two kinds of functions (Figure M.1): "Processing functions" implement the various parts of the pipeline. "GUI" functions are part of the GUI. The way each function calls the others is shown in Figure M.1.

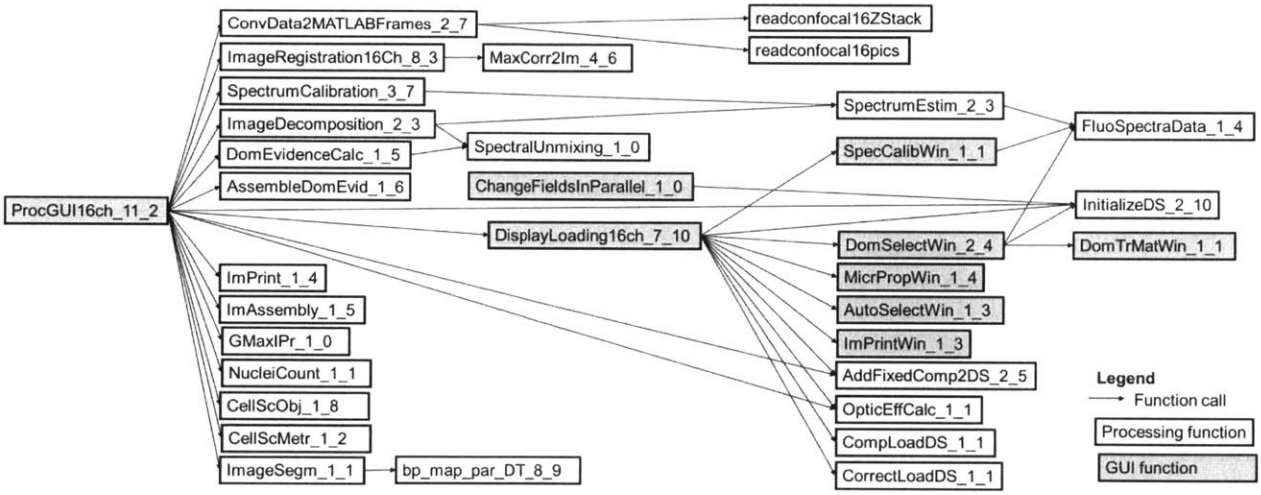


Figure M.7.3: Function tree of the MATLAB software that implements the image processing pipeline.

A brief description of the key functions shown in Figure M.1 is provided in Table M.1. Each function requires an input (either raw data or the data outcome of another function) and provides an output (which is saved in the computer hard disk). This generates a hierarchy between the functions (e.g. some functions need to run first so that a particular function can run and process their outcome), which is shown in Figure M.2.

The two key GUI functions are "ProcGUI16ch" and "DisplayLoading16ch". For each experiment, the user is required to generate a "parameter" file that contains imaging parameters and image processing parameters. This is implemented using function "DisplayLoading16ch", which can start a new parameter file from scratch or generate a new parameter file by editing older parameter files. "ProcGUI16ch" is the major GUI file that controls the execution of the various functions that implement the processing pipeline. This function calls the "DisplayLoading16ch" in order to select particular experimental data, enables batch mode (process multiple experiments in a queue), and enables parallel processing (utilization of multiple CPU cores).

Function	Description	Input	Output
Processing Functions – General Purpose			
ConvData2MATLABFra	Convert raw data to MATLAB	User raw data	RawFrame (1)

mes	files		MaxIntPlane(2)
ImageRegistration16Ch	Image registration	RawFrame (1)	RegImagDat (3) ImageExt (4)
SpectrumCalibration	MA-PMT spectral position calibration	User raw data	pLO,pDL,pK (5)
ImageDecomposition	Spectral unmixing (include all sources of all classes)	RawFrame (1) pLO,pDL,pK (5)	CompConMap (6)
ImAssembly	Assemble unmixed frames	CompConMap (6) RegImagDat (3)	AsblConMap (7)
GMaxIPr	Generate assembly of unmixed images at plane of max intensity	MaxIntPlane(2) RegImagDat (3) AsblConMap (7)	MaxIData, MaxIntImage (8)
ImPrint	Save assembled unmixed frames as images	AsblConMap (7) MaxIntImage (8)	-
DomEvidenceCalc	Calculate class evidence of each frame	RawFrame (1) pLO,pDL,pK (5)	CompConMap, EvidenceDom (9)
AssembleDomEvid	Assemble class evidence calculations of all frames	CompConMap, EvidenceDom (9) RegImagDat (3)	DomProbMapPROC, AsblCompConMap (10)
ImageSegm	Image segmentation: first local (Bayesian classifier), then global by MRF	DomProbMapPROC, AsblCompConMap (10) ImageExt (4)	ImSegm (11)
Processing Functions – Cell-Scaffold Processing			
CellScObj	Identify cell objects in segmented images	AsblConMap (7) ImSegm (11)	CellsOBJ (12)
CellScMetr	Quantify features for each cell	DomProbMapPROC (10) CellsOBJ (12)	CellMetrics, Metr (13)
GUI Functions			
ProcGUI16ch	- Call DisplayLoading16ch - Pick and run particular parts of the image processing pipeline - Display results - Turn parallel processing on/off - Enable batch mode.	-	-
DisplayLoading16ch	Load, edit, save parameter files.	-	-

Table M-7: Description of the major functions of the MATLAB software implementation of the image processing codes described in chapters 2 and 3.

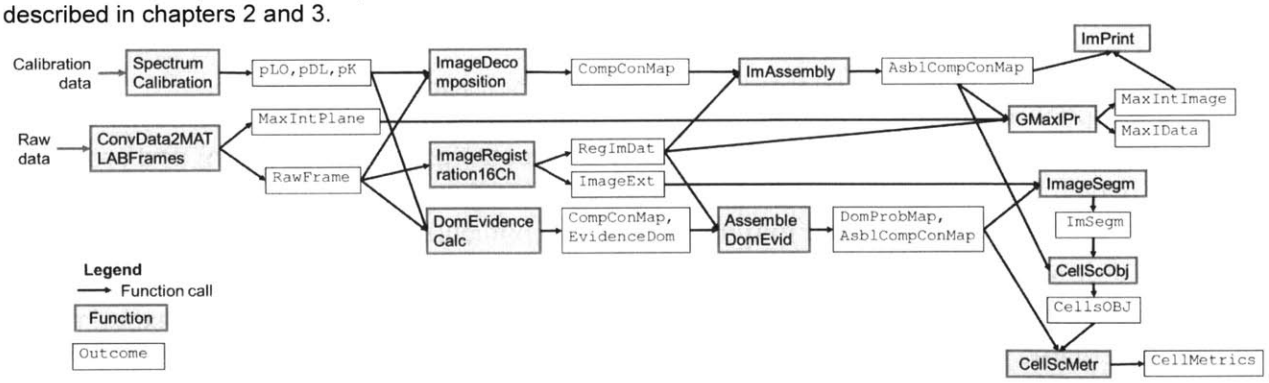


Figure M.4: Function hierarchy. Depiction of the function outcomes required to run downstream functions.

The MATLAB source code of the key processing functions is provided in the following sections: The complete MATLAB GUI software is available online at web.mit.edu/tzeranis/Public/Code and in the web site of the So lab (web.mit.edu/solab).

ConvData2MATLABFrames

```
function ZstepSaved = ConvData2MATLABFrames_3_1(Dat)
% m-file that converts raw microscope data (generated by the confocal aps)
% into matlab M-files
% Dimitris Tzeranis February 2009 - JUne 2013
```

```
 fprintf('=== Convert Raw Data to Matlab Frames ===\n')
 % create folder if necessary
if isempty(dir(Dat.DatCnv.Dir))
    mkdir(Dat.DatCnv.Dir);
end
%% PROCESS MMM DATA (Jaewon's Code)
if strcmpi(Dat.Img.Instr,'MMM')
    FileName = [Dat.Nom.MainDir,Dat.Nom.DASH,Dat.Nom.fname,Dat.Nom.Suffix];
    Raw4Frame = ReadMMMFrame_1_0(FileName);
    % save images in disk
    for ctF = 1:4
        RawFrame = Raw4Frame(:,:,:,ctF);
        SaveFileName = [Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
            Padding(ctF-1,Dat.Nom.PadNum),'_',Padding(1,Dat.Nom.PadNum)];
        save(SaveFileName,'RawFrame')
    end
    ZstepSaved = 1;
    fprintf('\t... done\n\n')
    return
end
%% PROCESS ORIGINAL MICROSCOPE DATA
DatDatCnv = struct('DatCnv',Dat.DatCnv,'Nom',Dat.Nom,'Img',Dat.Img);
if (Dat.Img.Xsteps==1) && (Dat.Img.Ysteps==1)
    FileName = [Dat.Nom.MainDir,Dat.Nom.DASH,Dat.Nom.fname,Dat.Nom.Suffix];
    if ~isempty(dir(FileName))
        if Dat.Img.Zsteps==1 % single image
            % read
            RawFrame = uint16(readconfocal16pics(FileName));
            % save
            SaveFileName = [Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
                        Padding(0,Dat.Nom.PadNum),'_',Padding(1,Dat.Nom.PadNum)];
            save(SaveFileName,'RawFrame')
            ZstepSaved = 1;
        else % Z-stack
            ReadFindMaxISaveZstack(DatDatCnv,FileName,0);
            %
            ZstepSaved = Dat.DatCnv.ZRange(1):Dat.DatCnv.ZRange(2);
        end
    else
        % the file does not exist
        disp(['the file ',FileName,' was not found'])
        ZstepSaved = [];
        fprintf('... done\n\n')
        return
    end
else
    % identify which frames to process
    XWinSz = Dat.DatCnv.XRange(2) - Dat.DatCnv.XRange(1) + 1;
    YWinSz = Dat.DatCnv.YRange(2) - Dat.DatCnv.YRange(1) + 1;
    FileNum = zeros(1,YWinSz*XWinSz);
    ctFN = 0;
    for ctR = Dat.DatCnv.YRange(1):Dat.DatCnv.YRange(2)
        for ctC = Dat.DatCnv.XRange(1):Dat.DatCnv.XRange(2)
            ctF = ctC + (ctR-1)*Dat.Img.Xsteps;
            ctFN = ctFN +1;
            FileNum(ctFN) = ctF-1;
        end
    end
    CheckFile = false(1,length(FileNum));
    if Dat.Proc.ParProc
        disp('distributed computing enabled')
        parfor ctFF = 1:length(FileNum)
            ctF = FileNum(ctFF);
            if Dat.Nom.Under
                FileName = [Dat.Nom.MainDir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
                    Padding(ctF+Dat.Nom.FirstFileNum,Dat.Nom.PadNum),Dat.Nom.Suffix];
            else
                FileName = [Dat.Nom.MainDir,Dat.Nom.DASH,Dat.Nom.fname,...
                    Padding(ctF+Dat.Nom.FirstFileNum,Dat.Nom.PadNum),Dat.Nom.Suffix];
            end
            if ~isempty(dir(FileName))
```

```
                CheckFile(ctFF) = true;
                if Dat.DatCnv.ZBinSw && ((Dat.DatCnv.ZRange(2)-Dat.DatCnv.ZRange(1))>0)
                    % binning
                    ReadBinSaveZstack(DatDatCnv,FileName,ctF);
                else
                    % no binning
                    ReadFindMaxISaveZstack(DatDatCnv,FileName,ctF);
                end
            else
                % the file does not exist
                disp(['the file ',FileName,' was not found'])
            end
        end

    else
        disp('distributed computing not enabled')
        for ctFF = 1:length(FileNum)
            ctF = FileNum(ctFF);
            % filename to read
            if Dat.Nom.Under
                FileName = [Dat.Nom.MainDir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
                    Padding(ctF+Dat.Nom.FirstFileNum,Dat.Nom.PadNum),Dat.Nom.Suffix];
            else
                FileName = [Dat.Nom.MainDir,Dat.Nom.DASH,Dat.Nom.fname,...
                    Padding(ctF+Dat.Nom.FirstFileNum,Dat.Nom.PadNum),Dat.Nom.Suffix];
            end
             % read file if it exists
            if ~isempty(dir(FileName))
                CheckFile(ctFF) = true;
                % save
                if Dat.DatCnv.ZBinSw && ((Dat.DatCnv.ZRange(2)-Dat.DatCnv.ZRange(1))>0)
                    % binning
                    ReadBinSaveZstack(DatDatCnv,FileName,ctF);
                else
                    % no binning
                    ReadFindMaxISaveZstack(DatDatCnv,FileName,ctF);
                end
            else
                % the file does not exist
                disp(['the file ',FileName,' was not found'])
            end
        end
   end
    if sum(CheckFile)==length(FileNum)
        ZstepSaved = Dat.DatCnv.ZRange(1):Dat.DatCnv.ZRange(2);
    else
        ZstepSaved = [];
    end
end
fprintf('\t\t..... done\n\n')
return
%% ReadFindMaxISaveZstack
function ReadFindMaxISaveZstack(Dat,FileName,ctF)
% read and save data as MATLAB files
ZstackRead = readconfocal16ZStack(FileName,Dat.Img.Zsteps);
ZPlanes = Dat.DatCnv.ZRange(1):Dat.DatCnv.ZRange(2);
PC = zeros(256,256,length(ZPlanes),'uint16');
for ctZ = 1:length(ZPlanes)
    RawFrame = ZstackRead(ZPlanes(ctZ)).RawData;
    PC(:,:,ctZ) = uint16(sum(RawFrame,3));
    RawFrame = uint16(RawFrame);
    %
    SaveFileName = [Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
        Padding(ctF,Dat.Nom.PadNum),'_',Padding(ZPlanes(ctZ),Dat.Nom.PadNum)];
    save(SaveFileName,'RawFrame')
end
% find and save plane of maximum intensity
[maxPC,MaxIntPlane] = max(PC,[],3);
clear maxPC
MaxIntPlane = uint8(MaxIntPlane);
SaveFileName = [Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,...
```

```
    '_MaxIntPlane_',Padding(ctF,Dat.Nom.PadNum)];
save(SaveFileName,'MaxIntPlane')
return
%% ReadBinSaveZstack
function ReadBinSaveZstack(Dat,FileName,ctF)
ZstackRead = readconfocal16ZStack(FileName,Dat.Img.Zsteps);
RawFrame = ZstackRead(Dat.DatCnv.ZRange(1)).RawData;
for ctB = (Dat.DatCnv.ZRange(1)+1):Dat.DatCnv.ZRange(2);
    RawFrame = RawFrame + ZstackRead(ctB).RawData;
end
RawFrame = uint16(RawFrame);
% save
SaveFileName = [Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
    Padding(ctF,Dat.Nom.PadNum),'_',Padding(1,Dat.Nom.PadNum)];
save(SaveFileName,'RawFrame')
return
```

readconfocal16pics

```
function fdata = readconfocal16pics(fname)
% m-file to read int files generated by the "confocal" application.
% reads files that contain 16 channels
% Ceated by Hyuk San. Modified by Dimitris Tzeranis, So Lab, March 2006
% SYNTAX: fdata = readconfocal16pics('fname');
fid = fopen(fname,'r'); % open file
fdata = zeros(256,256,16); % initialization
for ct1 = 1:16;fdata(:,:,ct1) = fread(fid,[256 256],'uint16'); end % read
fclose(fid); % closing file
```

readconfocal16ZStack

```
% m-file to read int files generated by the "confocal" application of the
% original microscope when it runs in z-scanning mode. Each image contains
% 16 channles and its size is 2 MB (2*1024^2 bytes).
% Based on the file "readconfocal16pic" that reads a single 16-channel
fid = fopen(fname,'r'); % open file
for ct2 = 1:NumberFrames; fdataS(ct2).RawData = zeros(256,256,16,'uint16'); end
for ct1 = 1:16
    for ct2 = 1 : NumberFrames
        fdataS(ct2).RawData(:,:,ct1) = (fread(fid,[256 256],'uint16'))';
    end
end
fclose(fid); % close file
```

ImageRegistration16Ch

```
function ZstepSaved = ImageRegistration16Ch_8_5(Dat)
% function ImageRegistration16ChV2 that generates an assembly image by
% putting together many small images. The algorithm works as follows:
%  1) for each pair of neighbouring images calculate how the cross
%  correlation CC of the overlaping parts of the images varies as a function
%  of the shift between the two images.
%  2) pick as nominal shift rO = [dx,dy] the one that maximizes cross-
%  correlation.
%  3) In the neighborhood of rO approximate cc = cc(r()) - 1/2[dx,dy]*K*[dx dy]
%  The "stiffness" matrix K denotes how sensitive is the result to the
%  choice of O
%  4) Combine the K, rO calculated from the elementary image pairs into a
%  larger system KK*RO = bb(rO)
%  5) Solve for RO. This is the optimum global position of the images.
%  6) From RO calculate the optimum shif between images
%  7) assemble image based on R0
% Dimitrios Tzeranis, Feb 2009 - June 2013
fprintf('=== Image registration ===\n')
[unused1,unused2,unused3] = mkdir(Dat.ImReg.Dir);
DEBUGMODE = false; % if true save 'NaNMatrix','degenC','degenR','RegData'
%% === case stepX = stepY = 1
if (Dat.Img.Xsteps==1) && (Dat.Img.Ysteps==1)
    for ctZ = Dat.ImReg.ZRange(1):Dat.ImReg.ZRange(2)
        % === data input
        % read RawFrame
        load([Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',
```

```
 Padding(0,Dat.Nom.PadNum),'_',Padding(ctZ,Dat.Nom.PadNum)]);
        AsblFrame =  RawFrame; clear RawFrame
        if  Dat.ImReg.RemLSw
            AsblFrame = AsblFrame(Dat.ImReg.RemLPixNum + 1:end,:,:);
        end
        ImageExt = true(Dat.Img.NumRow,Dat.Img.NumCol);
        SaveFileDir = [Dat.ImReg.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',
Dat.ImReg.SvName,'_ImageExt'];
        save(SaveFileDir,'ImageExt')
        if Dat.ImReg.SwSvImg
            photons = sum(AsblFrame,3);
            photons = imadjust(double(photons)/max(max(photons)));
            SaveFileDir = [Dat.ImReg.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
                Dat.ImReg.SvName,'_',Padding(ctZ,Dat.Nom.PadNum),'.png'];
            imwrite(photons,SaveFileDir,'png')
        end
    end
    ZstepSaved = Dat.ImReg.ZRange(1):Dat.ImReg.ZRange(2);
    fprintf('\t\t..... done\n\n')
    return
end
%% === case Step X>1 or Step Y>1
if Dat.Proc.ParProc
    fprintf('\tdistributed computing enabled\n')
else
    fprintf('\tdistributed computing not enabled\n')
end
% initialize output
RegImagDat = struct('dispO',[],'posiO',[],'KK',[],'bb',[],'insertO',[],'rangeO',[]);
% size of the assembly window in images
XWinSz = Dat.ImReg.XRange(2) - Dat.ImReg.XRange(1) + 1;
YWinSz = Dat.ImReg.YRange(2) - Dat.ImReg.YRange(1) + 1;
% identify the frame numbers to be processed, store in FileNum vector
FileNum = zeros(1,YWinSz*XWinSz);
ctFN = 0;
for ctR = Dat.ImReg.YRange(1):Dat.ImReg.YRange(2)
    for ctC = Dat.ImReg.XRange(1):Dat.ImReg.XRange(2)
        ctF = ctC + (ctR-1)*Dat.Img.Xsteps;
        ctFN = ctFN +1;        FileNum(ctFN) = ctF;
    end
end
% === calculate optimum dispacement & stiffness for pairs of neighbor images
% prepare Data matrix for the function ReadMChImage
DataIMR = cell(5,1);
DataIMR{1} = [Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_'];
DataIMR{2} = Dat.Nom.PadNum; DataIMR{3} = Dat.ImReg.RemLSw;
DataIMR{4} = Dat.ImReg.RemLPixNum; DataIMR{5} = Dat.Img.Instr;
 % calculate Dat.ImReg.SrchxRL, Dat.ImReg.SrchyTB
Dat.ImReg.SrchxRL = (Dat.Img.NumCol -
round(Dat.Img.XstepSz/(Dat.Img.PxSz*(Dat.Img.Res/32)*(40/Dat.Img.Magn)))) +
[-20:20]; % search space x, left-right case
Dat.ImReg.SrchxRL(Dat.ImReg.SrchxRL<4) = []; Dat.ImReg.SrchxRL(Dat.ImReg.SrchxRL>250) = [];
Dat.ImReg.SrchyTB = (Dat.Img.NumRow -
round(Dat.Img.YstepSz/(Dat.Img.PxSz*(Dat.Img.Res/32)*(40/Dat.Img.Magn)))) +
[-20:20]; % search space y, top-bottom case
Dat.ImReg.SrchyTB(Dat.ImReg.SrchyTB<4) = [];
Dat.ImReg.SrchyTB(Dat.ImReg.SrchyTB>250) = [];
% prepare Data matrix for the function MaxCorr2Im_V4_6
DataCR = cell(4,1);
DataCR{1} = Dat.ImReg.SrchxRL;DataCR{2} = Dat.ImReg.SrchyRL;
DataCR{3} = Dat.ImReg.SrchxTB;DataCR{4} = Dat.ImReg.SrchyTB;
DataCR{5} = Dat.Img.XstepSz;DataCR{6} = Dat.Img.YstepSz;
DataCR{7} = Dat.Img.PxSz;DataCR{8} = Dat.Img.Magn;
DataCR{9} = Dat.Img.Res;
 % initialize local KK and bb
RegData = struct('KKL',[],'KKT',[],'bbL',[],'bbT',[]);
for ctF = 1:XWinSz*YWinSz
    RegData(ctF).KKL = zeros(2);    RegData(ctF).KKT = zeros(2);
    RegData(ctF).bbL = zeros(2,1);    RegData(ctF).bbT = zeros(2,1);
end
% calculate local KK and bbs
```

```
fprintf('\tCompare neighbouring images for Row: ')
if Dat.Proc.ParProc
    ctF = 1;      ctZ = Dat.ImReg.CalcZRange(1);     % define snap size
    Im16F = sum(ReadMChImage(FileNum(ctF),ctZ,DataIMR),3);
    ImSzR = size(Im16F,1);     ImSzC = size(Im16F,2);
    CalcZRange = Dat.ImReg.CalcZRange;     % Z range
    % initialize NaNMatrix
    NaNMatrixRL = zeros(YWinSz*XWinSz,1);
    NaNMatrixTB = zeros(YWinSz*XWinSz,1);
     % calculate the K matrix and the b vector based on all stacks
    parfor ctF = 1:XWinSz*YWinSz
        ctR = floor((ctF-1)/XWinSz) + 1;
        ctC = ctF - (ctR-1)*XWinSz;
        % correlate neighb images
        if (ctR == 1) && (ctC == 1)
        elseif (ctR == 1) && (ctC > 1)
            % corr2 L-R
            for ctZ = CalcZRange(1):CalcZRange(2)
                ctL = ctF - 1;
                ImL = sum(ReadMChImage(FileNum(ctL),ctZ,DataIMR),3);
                ImF = sum(ReadMChImage(FileNum(ctF),ctZ,DataIMR),3);
                [rO,KrO,NaNMatrixRL(ctF)] = MaxCorr2Im_4_6(ImL,ImF,1,DataCR);
                RegData(ctF).KKL = RegData(ctF).KKL + KrO;
                RegData(ctF).bbL = RegData(ctF).bbL + KrO*rO;
            end
         elseif (ctR > 1) && (ctC == 1)
            fprintf(' %s',num2str(ctR))
            % corr2 T-B
            for ctZ = CalcZRange(1):CalcZRange(2)
                ctT = ctF - XWinSz;
                ImT = sum(ReadMChImage(FileNum(ctT),ctZ,DataIMR),3);
                ImF = sum(ReadMChImage(FileNum(ctF),ctZ,DataIMR),3);
                [rO,KrO,NaNMatrixTB(ctF)] = MaxCorr2Im_4_6(ImT,ImF,2,DataCR);
                RegData(ctF).KKT = RegData(ctF).KKT + KrO;
                RegData(ctF).bbT = RegData(ctF).bbT + KrO*rO;
            end
        else
            for ctZ = CalcZRange(1):CalcZRange(2)
                % corr2 L-R
                ctL = ctF - 1;
                ImL = sum(ReadMChImage(FileNum(ctL),ctZ,DataIMR),3);
                ImF = sum(ReadMChImage(FileNum(ctF),ctZ,DataIMR),3);
                [rO,KrO,NaNMatrixRL(ctF)] = MaxCorr2Im_4_6(ImL,ImF,1,DataCR);
                RegData(ctF).KKL = RegData(ctF).KKL + KrO;
                RegData(ctF).bbL = RegData(ctF).bbL + KrO*rO;
                % corr2 T-B
                ctT = ctF - XWinSz;
                ImT = sum(ReadMChImage(FileNum(ctT),ctZ,DataIMR),3);
                [rO,KrO,NaNMatrixTB(ctF)] = MaxCorr2Im_4_6(ImT,ImF,2,DataCR);
                RegData(ctF).KKT = RegData(ctF).KKT + KrO;
                RegData(ctF).bbT = RegData(ctF).bbT + KrO*rO;
            end
        end %location of frame
    end % all frames
    NaNMatrix = zeros(YWinSz,XWinSz,2);
    NaNMatrix(:,:,1) = reshape(NaNMatrixRL,[YWinSz,XWinSz]);
    NaNMatrix(:,:,2) = reshape(NaNMatrixTB,[YWinSz,XWinSz]);
    clear NaNMatrixTB NaNMatrixRL
else %SERIAL IMPLEMENTATION - NO PARALLEL PROCESSING
    % initialize NaNMatrix matrix
    NaNMatrix = zeros(YWinSz,XWinSz,2); % first plane for L, second for T
    % calculate the K matrix and the b vector based on all stacks1  2
    for ctF = 1:XWinSz*YWinSz
        ctR = floor((ctF-1)/XWinSz) + 1;
        ctC = ctF - (ctR-1)*XWinSz;
        % correlate neighb images
        if (ctR == 1) && (ctC == 1)
            ctZ = Dat.ImReg.CalcZRange(1);
            Im16F = sum(ReadMChImage(FileNum(ctF),ctZ,DataIMR),3);
            ImSzR = size(Im16F,1);
            ImSzC = size(Im16F,2);
```

```
            fprintf(' %s',num2str(ctR))
        elseif (ctR == 1) && (ctC > 1)
            % corr2 L-R
            for ctZ = Dat.ImReg.CalcZRange(1):Dat.ImReg.CalcZRange(2)
                ctL = ctF - 1;
                ImL = sum(ReadMChImage(FileNum(ctL),ctZ,DataIMR),3);
                ImF = sum(ReadMChImage(FileNum(ctF),ctZ,DataIMR),3);
                [rO,KrO,NaNMatrix(ctR,ctC,1)] = MaxCorr2Im_4_6(ImL,ImF,1,DataCR);
                RegData(ctF).KKL = RegData(ctF).KKL + KrO;
                RegData(ctF).bbL = RegData(ctF).bbL + KrO*rO;
            end
         elseif (ctR > 1) && (ctC == 1)
            fprintf(' %s',num2str(ctR))
            % corr2 T-B
            for ctZ = Dat.ImReg.CalcZRange(1):Dat.ImReg.CalcZRange(2)
                ctT = ctF - XWinSz;
                ImT = sum(ReadMChImage(FileNum(ctT),ctZ,DataIMR),3);
                ImF = sum(ReadMChImage(FileNum(ctF),ctZ,DataIMR),3);
                [rO,KrO,NaNMatrix(ctR,ctC,2)] = MaxCorr2Im_4_6(ImT,ImF,2,DataCR);
                RegData(ctF).KKT = RegData(ctF).KKT + KrO;
                RegData(ctF).bbT = RegData(ctF).bbT + KrO*rO;
            end
        else
            for ctZ = Dat.ImReg.CalcZRange(1):Dat.ImReg.CalcZRange(2)
                % corr2 L-R
                ctL = ctF - 1;
                ImL = sum(ReadMChImage(FileNum(ctL),ctZ,DataIMR),3);
                ImF = sum(ReadMChImage(FileNum(ctF),ctZ,DataIMR),3);
                [rO,KrO,NaNMatrix(ctR,ctC,1)] = MaxCorr2Im_4_6(ImL,ImF,1,DataCR);
                RegData(ctF).KKL = RegData(ctF).KKL + KrO;
                RegData(ctF).bbL = RegData(ctF).bbL + KrO*rO;
                % corr2 T-B
                ctT = ctF - XWinSz;
                ImT = sum(ReadMChImage(FileNum(ctT),ctZ,DataIMR),3);
                [rO,KrO,NaNMatrix(ctR,ctC,2)] = MaxCorr2Im_4_6(ImT,ImF,2,DataCR);
                RegData(ctF).KKT = RegData(ctF).KKT + KrO;
                RegData(ctF).bbT = RegData(ctF).bbT + KrO*rO;
            end
        end %location of frame
    end % all frames
end % parallel or not
fprintf('\n')
%% === identify degenerate columns/rows
degenC = find(sum(NaNMatrix(:,:,1),1)==size(NaNMatrix,1)); %degenerate columns
degenR = find(sum(NaNMatrix(:,:,2),2)==size(NaNMatrix,2)); %degenerate rows
if ~isempty(degenC) || ~isempty(degenR)
    if ~isempty(degenC);
        fprintf(['\tcolumns ',num2str(degenC(:)'),' are degenerate wrt their left columns\n'])
    end
    if ~isempty(degenR);
        fprintf(['\trows ',num2str(degenR(:)'),' are degenerate wrt their above rows\n'])
    end
end
%% ===== generate K,b matrices
RegImagDat.KK = zeros(2*XWinSz*YWinSz);
RegImagDat.bb = zeros(2*XWinSz*YWinSz,1);
fprintf('\tGenerate K,b matrices for Row: ')
for ctF = 1:XWinSz*YWinSz
    ctR = floor((ctF-1)/XWinSz) + 1;
    ctC = ctF - (ctR-1)*XWinSz;
    if (ctR == 1) && (ctC == 1)
        fprintf(' %s',num2str(ctR))
    elseif (ctR == 1) && (ctC > 1)
        ctL = ctF - 1;
        Kaddress = [2*ctF-1,2*ctF,2*ctL-1,2*ctL];
        RegImagDat.KK(Kaddress,Kaddress) = RegImagDat.KK(Kaddress,Kaddress) + [1 0 -1 0 ;0 1 0 -
1]'*(RegData(ctF).KKL)*[1 0 -1 0 ;0 1 0 -1];
        RegImagDat.bb(Kaddress) = RegImagDat.bb(Kaddress) + ...
            [RegData(ctF).bbL;-RegData(ctF).bbL];
    elseif (ctR > 1) && (ctC == 1)
        fprintf(' %s',num2str(ctR))
```

```
        ctT = ctF - XWinSz;
        Kaddress = [2*ctF-1,2*ctF,2*ctT-1,2*ctT];
        RegImagDat.KK(Kaddress,Kaddress) = RegImagDat.KK(Kaddress,Kaddress) + [1 0 -1 0 ;0 1 0 -
1]'*(RegData(ctF).KKT)*[1 0 -1 0 ;0 1 0 -1];
        RegImagDat.bb(Kaddress) = RegImagDat.bb(Kaddress) + ...
            [RegData(ctF).bbT;-RegData(ctF).bbT];
    else
        ctL = ctF - 1;
        Kaddress = [2*ctF-1,2*ctF,2*ctL-1,2*ctL];
        RegImagDat.KK(Kaddress,Kaddress) = RegImagDat.KK(Kaddress,Kaddress) + [1 0 -1 0 ;0 1 0 -
1]'*(RegData(ctF).KKL)*[1 0 -1 0 ;0 1 0 -1];
        RegImagDat.bb(Kaddress) = RegImagDat.bb(Kaddress) + ...
            [RegData(ctF).bbL;-RegData(ctF).bbL];
        %
        ctT = ctF - XWinSz;
        Kaddress = [2*ctF-1,2*ctF,2*ctT-1,2*ctT];
        RegImagDat.KK(Kaddress,Kaddress) = RegImagDat.KK(Kaddress,Kaddress) + [1 0 -1 0 ;0 1 0 -
1]'*(RegData(ctF).KKT)*[1 0 -1 0 ;0 1 0 -1];
        RegImagDat.bb(Kaddress) = RegImagDat.bb(Kaddress) + ...
            [RegData(ctF).bbT;-RegData(ctF).bbT];
    end
end
fprintf(' \n ')
% saving temporarily registration results (DEBUG MODE ONLY)
if DEBUGMODE
    SaveFileDir = [Dat.ImReg.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_ImageRegTemp'];
    save(SaveFileDir,'RegImagDat','NaNMatrix','degenC','degenR','RegData')
end
%% === identify the slides of the image that cannot be registered
addressF = zeros(2,1);
SnapsDGch = false(1,XWinSz*YWinSz);
for ctF = 1:XWinSz*YWinSz
    addressF = [2*ctF-1;2*ctF];
    if (sum(sum(abs(RegImagDat.KK(addressF,:))))+sum(sum(abs(RegImagDat.KK(:,addressF)))))==0
        SnapsDGch(ctF) = true;
    end
end
SnapsDG = find(SnapsDGch); % the snaps that canoot be registered
%% === solve for the optimum global positions of images
fprintf('\t Calculate optimum displacements | ')
RegImagDat.posiO = zeros(2*XWinSz*YWinSz,1);
if isempty(SnapsDG)
    % solve the linear system. by definition, the position of frame 1 is [0,0]
    RegImagDat.posiO = [0;0;RegImagDat.KK(3:end,3:end)\RegImagDat.bb(3:end)];
else
    Snaps2calc = 1:XWinSz*YWinSz;    Snaps2calc([1;SnapsDG(:)]) = [];
    addr2calc = [2*Snaps2calc(:) - 1;2*Snaps2calc(:)];
    RegImagDat.posiO(addr2calc) = RegImagDat.KK(addr2calc,addr2calc)\RegImagDat.bb(addr2calc);
    % for each degenerate pixel, try to find a neighbor whose I know the
    % optimum position
    PxSz = (Dat.Img.Res)/(Dat.Img.Magn)*(40/32)*Dat.Img.PxSz; %pixel size [um/pixels]
    FOV  = 256*PxSz;    FOVrow  = Dat.Img.NumRow*PxSz;
    FOVcol  = Dat.Img.NumCol*PxSz;    SnapsDGres = sort(SnapsDG);
    while ~isempty(SnapsDGres)
        DG2solved = [];
        for ctDG = 1:length(SnapsDGres)
            ctF = SnapsDG(ctDG);       ctR = floor((ctF-1)/XWinSz) + 1;
            ctC = ctF - (ctR-1)*XWinSz;
            addF = [2*ctF-1,2*ctF];
            if (ctC-1)>0 && isempty(find(SnapsDG==(ctF-1))) % try right
                ctG = ctF - 1;            addG = [2*ctG-1,2*ctG];
                RegImagDat.posiO(addF) = RegImagDat.posiO(addG) + ...
                [(FOVcol-Dat.Img.XstepSz)/PxSz;0];
            elseif ((ctC+1)<=XWinSz) && isempty(find(SnapsDG==(ctF+1))) % try left
                ctG = ctF + 1;      addG = [2*ctG-1,2*ctG];
                RegImagDat.posiO(addF) = RegImagDat.posiO(addG) - ...
                [(FOVcol-Dat.Img.XstepSz)/PxSz;0];
            elseif ((ctR-1)>0) && isempty(find(SnapsDG==(ctF-XWinSz))) % try top
                ctG = ctF - XWinSz;
                addG = [2*ctG-1,2*ctG];
                RegImagDat.posiO(addF) = RegImagDat.posiO(addG) + ...
```

```
            [0;(FOVrow-Dat.Img.YstepSz)/PxSz];
        elseif ((ctR+1)<=YWinSz) && isempty(find(SnapsDG==(ctF+XWinSz))) % try bottom
            ctG = ctF + XWinSz;          addG = [2*ctG-1,2*ctG];
            RegImagDat.posiO(addF) = RegImagDat.posiO(addG) - ...
            [0;(FOVrow-Dat.Img.YstepSz)/PxSz];
        else
            ctG = 0;
        end
        if ctG;
            DG2solved = [DG2solved;ctDG];
        end
    end
    SnapsDGres(DG2solved) = [];
  end
end
RegImagDat.posiO = round(RegImagDat.posiO);
%% === calculate optimum displacements between images
RegImagDat.dispO = zeros(YWinSz*XWinSz,1);
for ctF = 1:XWinSz*YWinSz
    ctR = floor((ctF-1)/XWinSz) + 1;
    ctC = ctF - (ctR-1)*XWinSz;
    addF = [2*ctF-1,2*ctF];
    if (ctR == 1) && (ctC == 1)
        RegImagDat.dispO(1:2) = 0;
    elseif (ctR > 1) && (ctC == 1)
        ctT = ctF - XWinSz;          addT = [2*ctT-1,2*ctT];
        RegImagDat.dispO(addF) = RegImagDat.posiO(addF) - RegImagDat.posiO(addT);
    else
        ctL = ctF - 1;          addL = [2*ctL-1,2*ctL];
        RegImagDat.dispO(addF) = RegImagDat.posiO(addF) - RegImagDat.posiO(addL);
    end
end
%% === calculate insert points
RegImagDat.insertO= zeros(YWinSz*XWinSz,1);RegImagDat.rangeO = zeros(YWinSz*XWinSz,1);
for ctF = 1:XWinSz*YWinSz
    ctR = floor((ctF-1)/XWinSz) + 1;    ctC = ctF - (ctR-1)*XWinSz;
    addF = [2*ctF-1,2*ctF];
    RegImagDat.insertO(addF) = [-ImSzC*(ctC-1);ImSzR*(ctR-1)] + ...
        [RegImagDat.posiO(addF(1));-RegImagDat.posiO(addF(2))] + [ImSzC*XWinSz - ...
        ((ImSzC))+1;1];
    RegImagDat.rangeO(addF) = RegImagDat.insertO(addF) + [ImSzC-1;ImSzR-1];
end
% saving registration results
SaveFileDir = [Dat.ImReg.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_ImageRegOverview'];
save(SaveFileDir,'RegImagDat')
%% === image assembly
% find the range of values of the insert and range matrices elemtents
maxR = max(RegImagDat.rangeO(1:2:end));
minR = min(RegImagDat.insertO(1:2:end));maxC = max(RegImagDat.rangeO(2:2:end));
minC = min(RegImagDat.insertO(2:2:end));
% remove columns/rows that are zero
if minR~=1
    TotalPixX = maxR - minR + 1;
    RegImagDat.rangeO(1:2:end)  = RegImagDat.rangeO(1:2:end) - minR + 1;
    RegImagDat.insertO(1:2:end) = RegImagDat.insertO(1:2:end) - minR + 1;
else
    TotalPixX = maxR;
end
TotalPixY = maxC;
if minC<=0
    TotalPixY = TotalPixY - minC + 1;
    RegImagDat.rangeO(2:2:end) = RegImagDat.rangeO(2:2:end) - minC + 1;
    RegImagDat.insertO(2:2:end)= RegImagDat.insertO(2:2:end) - minC + 1;
end
fprintf(' Assemble & Save Plane: ')
% assemblying frames
for ctZ = Dat.ImReg.ZRange(1):Dat.ImReg.ZRange(2)
    fprintf('%s ',num2str(ctZ))
    AsblFrame = zeros(TotalPixY,TotalPixX,Dat.Img.ChNum,'uint16');
    if ctZ == Dat.ImReg.ZRange(1)
        ImageExt = false(TotalPixY,TotalPixX);
```

```
    end
    for ctF = 1:XWinSz*YWinSz
        ctR = floor((ctF-1)/XWinSz) + 1;
        ctC = ctF - (ctR-1)*XWinSz;
        addF = [2*ctF-1,2*ctF];
        ImF = ReadMChImage(FileNum(ctF),ctZ,DataIMR);
        dO = RegImagDat.dispO(addF);
        % process images to combine data from neighbouring images
        if (ctR == 1) && (ctC == 1)
        elseif (ctR == 1) && (ctC > 1)
            % === L-R
            ctL = ctF - 1;
            ImL = ReadMChImage(FileNum(ctL),ctZ,DataIMR);
            [cs1,cs2] = CalculateWindowCS(ImL,ImF,1,dO(1),dO(2));
            MaskLR = (linspace(0,1,cs1(4)-cs1(3)+1)'*ones(1,cs1(2)-cs1(1)+1))';
            for ctCH = 1:Dat.Img.ChNum
                ImF(cs2(1):cs2(2),cs2(3):cs2(4),ctCH) = ...
                    round((1-MaskLR).*double(ImF(cs2(1):cs2(2),cs2(3):cs2(4),ctCH)) +
MaskLR.*double(ImL(cs1(1):cs1(2),cs1(3):cs1(4),ctCH)));
            end
        elseif (ctR > 1) && (ctC == 1)
            % === T-B
            ctT = ctF - XWinSz;
            ImT = ReadMChImage(FileNum(ctT),ctZ,DataIMR);
            [cs1,cs2] = CalculateWindowCS(ImT,ImF,2,dO(1),dO(2));
            MaskTB = (ones(cs1(4)-cs1(3)+1,1)*linspace(0,1,cs1(2)-cs1(1)+1))';
            for ctCH = 1:Dat.Img.ChNum
                ImF(cs2(1):cs2(2),cs2(3):cs2(4),ctCH) = ...
                    round((MaskTB).*double(ImF(cs2(1):cs2(2),cs2(3):cs2(4),ctCH)) + (1-
MaskTB).*double(ImT(cs1(1):cs1(2),cs1(3):cs1(4),ctCH)));
            end
        else
            % === L-R
            ctL = ctF - 1;
            ImL = ReadMChImage(FileNum(ctL),ctZ,DataIMR);
            [cs1,cs2] = CalculateWindowCS(ImL,ImF,1,dO(1),dO(2));
            MaskLR = (linspace(0,1,cs1(4)-cs1(3)+1)'*ones(1,cs1(2)-cs1(1)+1))';
            for ctCH = 1:Dat.Img.ChNum
                ImF(cs2(1):cs2(2),cs2(3):cs2(4),ctCH) = ...
                    round((1-MaskLR).*double(ImF(cs2(1):cs2(2),cs2(3):cs2(4),ctCH)) +
MaskLR.*double(ImL(cs1(1):cs1(2),cs1(3):cs1(4),ctCH)));
            end
            % === T-B
            ctT = ctF - XWinSz;
            addT = [2*ctT-1,2*ctT];
            dO =  RegImagDat.posiO(addF) - RegImagDat.posiO(addT);
            ImT = ReadMChImage(FileNum(ctT),ctZ,DataIMR);
            [cs1,cs2] = CalculateWindowCS(ImT,ImF,2,dO(1),dO(2));
            MaskTB = (ones(cs1(4)-cs1(3)+1,1)*linspace(0,1,cs1(2)-cs1(1)+1))';
            for ctCH = 1:Dat.Img.ChNum
ImF(cs2(1):cs2(2),cs2(3):cs2(4),ctCH) =
round((MaskTB).*double(ImF(cs2(1):cs2(2),cs2(3):cs2(4),ctCH)) +                    (1-
MaskTB).*double(ImT(cs1(1):cs1(2),cs1(3):cs1(4),ctCH)));
            end
        end
        AsblFrame(RegImagDat.insertO(addF(2)):RegImagDat.rangeO(addF(2)),
 RegImagDat.insertO(addF(1)):RegImagDat.rangeO(addF(1)),:) = ImF;
        if ctZ == Dat.ImReg.ZRange(1)
        ImageExt(RegImagDat.insertO(addF(2)):RegImagDat.rangeO(addF(2)),...
                 RegImagDat.insertO(addF(1)):RegImagDat.rangeO(addF(1))) = true;
        end
    end
    % save ImageExt
    SaveFileDir = [Dat.ImReg.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
        Dat.ImReg.SvName,'_ImageExt'];
    save(SaveFileDir,'ImageExt')
    % save images
    if Dat.ImReg.SwSvImg
        photons = sum(AsblFrame,3);
        photons = imadjust(double(photons)/max(max(photons)));
        % scale
```

```
        ImLength = length(photons);
        ScaleF = ImLength/800;
        if ScaleF>1; photons = imresize(photons,1/ScaleF);  end
        % save to HD
        SaveFileDir = [Dat.ImReg.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
            Dat.ImReg.SvName,'_',Padding(ctZ,Dat.Nom.PadNum),'.png'];
        imwrite(photons,SaveFileDir,'png')
    end

end
ZstepSaved = Dat.ImReg.ZRange(1):Dat.ImReg.ZRange(2); % save version
fprintf('\t\t..... done\n\n')
return
 function [cs1,cs2] = CalculateWindowCS(Im1,Im2,Dim,dx,dy)
% coordinates of the windows in the base image (B) and the image (F) that
% is positioned with respect to B @ displacement [dx,dy]' along dimension
% Dim. Im1:  base image (R or T). Im2:  image that needs to be positioned (L or B)
% Dimitris Tzeranis, February 2009
if Dim == 1 % R-L
    if dy>=0
        cs2 = [dy+1,size(Im2,1),size(Im2,2)-dx+1,size(Im2,2)];
        cs1 = [1,size(Im1,1)-dy,1,dx];
    else
        cs2 = [1,size(Im2,1)+dy,size(Im2,2)-dx+1,size(Im2,2)];
        cs1 = [1-dy,size(Im1,1),1,dx];
    end
else % T-B
    if dx>=0
        cs2 = [1,dy,1,size(Im2,2)-dx];
        cs1 = [size(Im1,1)-dy+1,size(Im1,1),dx+1,size(Im1,2)];
    else
        cs2 = [1,dy,1-dx,size(Im2,2)];
        cs1 = [size(Im1,1)-dy+1,size(Im1,1),1,size(Im1,2)+dx];
    end
end
cs1 = round(cs1);cs2 = round(cs2);
return
 function ImMF = ReadMChImage(ctF,ctZ,Data)
load([Data{1},Padding(ctF-1,Data{2}),'_',Padding(ctZ,Data{2})]);
if  Data{3}
    if strcmpi(Data{5},'MMM')
        ImMF = RawFrame(8:end,:,:);
    else
        % original microscope: remove left pixels
        ImMF = RawFrame(:,Data{4} + 1:end,:);
    end
else
    ImMF = RawFrame;
end
return
```

MaxCorr2Im

```
function [rO,KrO,NaNFlag] = MaxCorr2Im_4_6(Im1,Im2,Dim,Data)
% m-file that registers two images based on intensity data only
% - gets raw imaging data (no thresholding)
% - looks for a common optimum threshold that maximizes correlation and
% then approximates correlation in the area as a quadratic function of
% image overlap
% Im1:  base image (R or T)
% Im2:  image that needs to be positioned (L or B)
% Dim:  1 for L-R, 2 for T-B
% dxm:  range of shift along x
% dym:  range of shift along y
% rO:   optimum displacement in pixels [xo,yo]
% KrO:  stiffness matrix for the result,
%       (1,1): fxx , (1,2): fxy ,  (2,2): fyy
% Dimitrios Tzeranis, February - July 2011
NaNFlag = 0;
dxmRL = Data{1};% Dat.Asbl.SrchSpc.dxmRL
dymRL = Data{2};% Dat.Asbl.SrchSpc.dymRL
```

```
dxmTB = Data{3};% Dat.Asbl.SrchSpc.dxmTB
dymTB = Data{4};% Dat.Asbl.SrchSpc.dymTB
% ============= Part 1: find position of maximum correlation
if Dim == 1
    dxm = dxmRL;    dym = dymRL;
else
    dxm = dxmTB;    dym = dymTB;
end
 CorrCoeff = zeros(length(dxm),length(dym));
 ctF = 0;
for ctx = 1:length(dxm)
    dx = dxm(ctx);
    for cty = 1:length(dym)
        dy = dym(cty);        ctF = ctF + 1;
        [Wind1,Wind2] = CalculateWindow(Im1,Im2,Dim,dx,dy);
        [szWR,szWC] = size(Wind1);
        if Dim == 1 % R-L
            szWRm = round(linspace(1,szWR+1,9));
            for ctTH = 1:8
                temp = Wind1(szWRm(ctTH):szWRm(ctTH+1)-1,:);
                Wind1(szWRm(ctTH):szWRm(ctTH+1)-1,:) = (temp>=prctile(temp(:),50));
                temp = Wind2(szWRm(ctTH):szWRm(ctTH+1)-1,:);
                Wind2(szWRm(ctTH):szWRm(ctTH+1)-1,:) = (temp>=prctile(temp(:),50));
            end
        else % T-B
            szWRm = round(linspace(1,szWC+1,9));
            for ctTH = 1:8
                temp = Wind1(:,szWRm(ctTH):szWRm(ctTH+1)-1);
                Wind1(:,szWRm(ctTH):szWRm(ctTH+1)-1) = (temp>=prctile(temp(:),50));
                temp = Wind2(:,szWRm(ctTH):szWRm(ctTH+1)-1);
                Wind2(:,szWRm(ctTH):szWRm(ctTH+1)-1) = (temp>=prctile(temp(:),50));
            end
        end
        CorrCoeff(ctx,cty) = corr2(Wind2,Wind1);
    end
end
 maxCorr = max(max(CorrCoeff));
if isnan(maxCorr)
    if Dim == 1 % R-L
         rO = [(256-round(Data{5}/(Data{7}*(Data{9}/32*40/Data{8}))));0];
    else
         rO = [0;(256-round(Data{6}/(Data{7}*(Data{9}/32*40/Data{8}))))];
    end
    KrO = 1e-3*eye(2);
    NaNFlag = 1;
    return
end
[ctx,cty] = find(CorrCoeff == maxCorr);
rO = [dxm(ctx);dym(cty)];
NumrO = size(rO,2);
if NumrO>1
    rO = rO(:,round(NumrO/2));
end
 % ============= Part 2: calculatew stiffnes around max error
% calculate the "stiffness" matrix by approximating the CorrCoeff as a
% quadratic function z = b - A*(r-rO), where b = max(max(CorrCoeff)) and rO
% is the point of the optimum
 dxm = rO(1) + [-4:4];
dym = rO(2) + [-4:4];
 if Dim == 1
    dxm(dxm<1) = [];
else
    dym(dym<1) = [];
end
 CorrCoeff = zeros(length(dxm),length(dym));
b = zeros(length(dxm)*length(dym),1);
A = ones(length(dxm)*length(dym),3);
 ctF = 0;
for ctx = 1:length(dxm)
    dx = dxm(ctx);
    for cty = 1:length(dym)
```

```
        dy = dym(cty);
        ctF = ctF + 1;
        [Wind1,Wind2] = CalculateWindow(Im1,Im2,Dim,dx,dy);
        [szWR,szWC] = size(Wind1);
        if Dim == 1 % R-L
            szWRm = round(linspace(1,szWR+1,9));
            for ctTH = 1:8
                temp = Wind1(szWRm(ctTH):szWRm(ctTH+1)-1,:);
                Wind1(szWRm(ctTH):szWRm(ctTH+1)-1,:) = (temp>=prctile(temp(:),50));
                temp = Wind2(szWRm(ctTH):szWRm(ctTH+1)-1,:);
                Wind2(szWRm(ctTH):szWRm(ctTH+1)-1,:) = (temp>=prctile(temp(:),50));
            end
        else % T-B
            szWRm = round(linspace(1,szWC+1,9));
            for ctTH = 1:8
                temp = Wind1(:,szWRm(ctTH):szWRm(ctTH+1)-1);
                Wind1(:,szWRm(ctTH):szWRm(ctTH+1)-1) = (temp>=prctile(temp(:),50));
                temp = Wind2(:,szWRm(ctTH):szWRm(ctTH+1)-1);
                Wind2(:,szWRm(ctTH):szWRm(ctTH+1)-1) = (temp>=prctile(temp(:),50));
            end
        end
        CorrCoeff(ctx,cty) = corr2(Wind2,Wind1);
        b(ctF) = CorrCoeff(ctx,cty);        A(ctF,1) = dx;        A(ctF,3) = dy;
    end
end
 b = reshape(CorrCoeff,length(dxm)*length(dym),1) - max(max(CorrCoeff));
A(:,2) = -2*(A(:,1)-dxm(ctx)).*(A(:,3)-dym(cty));
A(:,1) = -(A(:,1)-dxm(ctx)).^2; A(:,3) = -(A(:,3)-dym(cty)).^2; x = A\b;
% if there is NaN, then stiffness is very small
if sum(isnan(x))
    NaNFlag = 1;    x = [1e-3 0 1e-3];
end
% form stiffness matrix
KrO = [x(1),x(2);x(2),x(3)];
 return
 function [Wind1,Wind2] = CalculateWindow(Im1,Im2,Dim,dx,dy)
 SkipPix = 3;% This needs to be smaller than dx,dy!!!!
 if Dim == 1 % R-L
    if dy>=0
        Wind2 = Im2(dy+1:end,end-dx+1+SkipPix:end,:);
        Wind1 = Im1(1:end-dy,1+SkipPix:dx,:);
    else
        Wind2 = Im2(1:end+dy,end-dx+1+SkipPix:end,:);
        Wind1 = Im1(1-dy:end,1+SkipPix:dx,:);
    end
else % T-B
    if dx>=0
        Wind2 = Im2(1+SkipPix:dy,1:end-dx,:);
        Wind1 = Im1(end-dy+1:end-SkipPix,dx+1:end,:);
    else
        Wind2 = Im2(1+SkipPix:dy,1-dx:end,:);
        Wind1 = Im1(end-dy+1:end-SkipPix,1:end+dx,:);
    end
end
 return
```

SpectrumCalibration

```
function Output = SpectrumCalibration_3_8(Dat)
% m file that calibrates the spectrum
% Dimitris Tzeranis Feb 2009 - June 2013
[unused,unused2,unused3] = mkdir(Dat.SpcClb.Dir);
%% ========= SPECTRUM CALIBRATION
fprintf('=== Spectrum Calibration ===\n')
 % Choose the variables Dat.SpcClb.L0o (beginning of the spectrum range
%   of first channel) and Dat.SpcClb.DeltaLo (spectrum width of each
%   channel) in order to minimize the error between the measured and
%   estimated spectra of several calibration dyes
 % === initialize global variables
NumDyes2Use = Dat.SpcClb.NuComp;Tag = Dat.SpcClb.SpcIG;
 % === calculate the system optical efficiency
```

```
% Dat.Img.OptEff = OpticalEfficiencyCalculation_V1_0(Dat.Img,false);
 % === define variables for optimization
Data = struct('NumDyes2Use',[],'Tag',[],'ChUsed',[],'SHGch',[],'OptEff',[],...
    'SpecResp',[],'PMTDiscRelGains',[]);
Data.NumDyes2Use = Dat.SpcClb.NuComp;Data.Tag = Dat.SpcClb.SpcIG;
Data.ChUsed = Dat.Img.ChUsed;Data.SHGch = Dat.SpcClb.SHGch;
Data.OptEff = Dat.Img.OptEff;Data.SpecResp = Dat.Img.SpecResp;
Data.PMTDiscRelGains = Dat.Img.PMTDiscRelGains;
 %% === spectrum calibration loop
% read the calibration dye images
CalDyeImages = cell(NumDyes2Use,1);
for ctD = 1:NumDyes2Use
    CalDyeImages{ctD} =
readconfocal16pics([Dat.Nom.MainDir,Dat.Nom.DASH,Dat.SpcClb.SpcFname{ctD}]);
end
 % divide each 256x256 image into SpectrEstimYSpacing parts along Y axis
YSpacLim = 256/Dat.SpcClb.DivNum;CalDyeSpec = zeros(NumDyes2Use,16);
 xL   = zeros(1,Dat.SpcClb.DivNum);yL0o = zeros(1,Dat.SpcClb.DivNum);
yDLo = zeros(1,Dat.SpcClb.DivNum);yKo = zeros(1,Dat.SpcClb.DivNum);
 fprintf('\trows:')
 for ctDiv = 1:Dat.SpcClb.DivNum
    % which rows to read
    R2ReeadRng = round([(ctDiv-1),ctDiv]*YSpacLim + [1 0]);
    fprintf('%sto%s|',num2str(R2ReeadRng(1)),num2str(R2ReeadRng(2)))
        % read files and calculate spectrum
    for ctD = 1:NumDyes2Use
        tmp = CalDyeImages{ctD};
        CalDyeSpec(ctD,:) = LStackSpectrum(tmp(:,R2ReeadRng(1):R2ReeadRng(2),:));
    end
    Data.CalDyeSpec = CalDyeSpec;
        % === initial estimation of  the three parameters
    x0 = zeros(1,3);
    if ctDiv ==1
        if ~isempty(Dat.SpcClb.SHGch) && ~isempty(Dat.Img.LasWvl)
            x0(1) = Dat.Img.LasWvl/2 - (13.5)/2 - (Dat.SpcClb.SHGch-1)*13.5;
        else
            x0(1) = mean(Dat.SpcClb.L0Range);
        end
        x0(2) = 13.5;        x0(3) = 0;
    else
        x0 = xx;
    end
     % === run optimization
    OptiLim = [Dat.SpcClb.L0Range(1),Dat.SpcClb.L0Range(2)
        Dat.SpcClb.DeltaLRange(1),Dat.SpcClb.DeltaLRange(2)
        -1,1];
    if Dat.SpcClb.DispOptim
        xx = fminunc(@(x)SpectrumError(x,Data),x0,optimset('LargeScale','off','Display','off'));
    else
        xx =
fmincon(@(x)SpectrumError(x,Data),x0,[],[],[],[],OptiLim(:,1),OptiLim(:,2),[],optimset('TolFun',1
e-6,'Algorithm','active-set','Display','off'));
    end
        xL(ctDiv) = mean(R2ReeadRng);    yL0o(ctDiv) = xx(1);
    yDLo(ctDiv) = xx(2);    yKo(ctDiv)  = xx(3);
end
fprintf('\n')
 % approximate optimum values of L0, DL, K as function or row
fprintf('\tlocal calculations\n')
pL0 = polyfit(xL,yL0o,1); pDL = polyfit(xL,yDLo,1); pK = polyfit(xL,yKo,1);
 % save calibration results
SaveFileDir = [Dat.SpcClb.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',Dat.SpcClb.SvName];
save(SaveFileDir,'pL0','pDL','pK')
 if Dat.SpcClb.SaveSw
    % ========== plot how L0, DL, and K vary over rows
    figure,plot(xL,yL0o,'x',1:256,polyval(pL0,1:256,'r--'))
    title(['L0 = ',num2str(pL0(1)),'*R + ',num2str(pL0(2))])
    A = getframe(gcf);
    SaveFileDir = [Dat.SpcClb.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_polynomialL0.bmp'];
    imwrite(A.cdata,SaveFileDir,'bmp')
    close(gcf)
```

```
    figure,plot(xL,yDLo,'x',1:256,polyval(pDL,1:256,'r--'))
    title(['DL = ',num2str(pDL(1)),'*R + ',num2str(pDL(2))])
    A = getframe(gcf);
    SaveFileDir = [Dat.SpcClb.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_polynomialDL.bmp'];
    imwrite(A.cdata,SaveFileDir,'bmp')
    close(gcf)
    figure,plot(xL,yKo,'x',1:256,polyval(pK,1:256,'r--'))
    title(['K = ',num2str(pK(1)),'*R + ',num2str(pK(2))])
    A = getframe(gcf);
    SaveFileDir = [Dat.SpcClb.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_polynomialK.bmp'];
    imwrite(A.cdata,SaveFileDir,'bmp')
    close(gcf)
    % ========== plot aproximation at top, middle and bottom row
    Rows = [1 128 256]; % where to compare (which rows)
        SpcEstData    = cell(7,1);
    SpcEstData{1} = pL0;    SpcEstData{2} = pDL;    SpcEstData{3} = Dat.SpcClb.SHGch;
    SpcEstData{4} = pK;    SpcEstData{5} = Dat.Img.OptEff;
    SpcEstData{6} = Dat.Img.SpecResp;    SpcEstData{7} = Dat.Img.PMTDiscRelGains;
        ColorsP = 'bgrcmk';    figure,hold on
    for ctR = 1:length(Rows)
        for ctD = 1:NumDyes2Use
            SpecE = SpectrumEstim_2_3(Tag{ctD},SpcEstData,true,Rows(ctR));
            SpecE = SpecE(Dat.Img.ChUsed);            SpecE = SpecE/sum(SpecE);
             % measured spectrum
            tmp = CalDyeImages{ctD};
            SpecM = LStackSpectrum(tmp(:,Rows(ctR),:));
            SpecM = SpecM(Dat.Img.ChUsed);
            SpecM = SpecM/sum(SpecM);
plot(Dat.Img.ChUsed,SpecM,ColorsP(ctR),Dat.Img.ChUsed,SpecE,[ColorsP(ctR),'--'])
        end
    end
    title('top row (blue), middle row (green), bottom row (red)')
    A = getframe(gcf);
    SaveFileDir = [Dat.SpcClb.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_estimerror.bmp'];
    imwrite(A.cdata,SaveFileDir,'bmp')    close(gcf)
end
Output = 1; % save version
fprintf('\t\t..... done\n\n')
return
 function Error = SpectrumError(x,Data)
% Function that calculates the error between the estimated and the measured
% emission spectra of several fluorescent dyes. The estimated spectra are
% calculated from the function IntegrateSpectra based on one parameter:
% Dat.SpcClb.L0o (beginning of the spectrum range of first channel)
% L0 = x(1);
% DeltaL = x(2);
% K = x(3);
Errori = zeros(1,Data.NumDyes2Use);
for ctD = 1:(Data.NumDyes2Use)
    % estimated spectrum
    SpcEstData    = cell(7,1);
    SpcEstData{1} = x(1);    SpcEstData{2} = x(2);
    SpcEstData{3} = Data.SHGch;    SpcEstData{4} = x(3);
    SpcEstData{5} = Data.OptEff;    SpcEstData{6} = Data.SpecResp;
    SpcEstData{7} = Data.PMTDiscRelGains;
     SpecE = SpectrumEstim_2_3(Data.Tag{ctD},SpcEstData,0);
    SpecE = SpecE(Data.ChUsed);    SpecE = SpecE/sum(SpecE);
        % measured spectrum
    SpecM = Data.CalDyeSpec(ctD,Data.ChUsed);       SpecM = SpecM/sum(SpecM);
        % error metric calculation
    Errori(ctD) = sum((SpecM(:) - SpecE(:)).^2);
end
 Error = sum(Errori);
return
```

ImageDecomposition

```
function ZPlanes = ImageDecomposition_2_4(Dat)
% Maximum likelihood estimation of the rate vector lambda based on a
% measurement y and a spectrum matrix S.
% Calculation consists of 2 parts
```

```
%  i) NNLS calculation --> LAMBDA_NNLS
% ii) ML optimization, use LAMBDA_NNLS as initial guess --> LAMBDA_ML
% Dimitris Tzeranis June 2009 - May 2013
fprintf('=== Spectral Unmixing (all components) ===\n')
%% ====== initialization
 % load spectrum estimation data
FileDirSpcClb = [Dat.SpcClb.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',Dat.SpcClb.SvName];
load(FileDirSpcClb) % load pL0 pDL pK
 % generate data for SpectrumEstim_2_3
% SpectrumEstim_2_3(SpcIG,SpcEstData,1,Row)
SpcEstData    = cell(4,1);SpcEstData{1} = pL0;
SpcEstData{2} = pDL;SpcEstData{3} = Dat.SpcClb.SHGch; SpcEstData{4} = pK;
SpcEstData{5} = Dat.Img.OptEff;SpcEstData{6} = Dat.Img.SpecResp;
SpcEstData{7} = Dat.Img.PMTDiscRelGains;
 % create folder if necessary
[unused1,unused2,unused3] = mkdir(Dat.ImDcmp.Dir);
 % === identify which frames & planes to process
% which frames to process
XWinSz = Dat.ImDcmp.XRange(2) - Dat.ImDcmp.XRange(1) + 1;
YWinSz = Dat.ImDcmp.YRange(2) - Dat.ImDcmp.YRange(1) + 1;
FileNum = zeros(1,YWinSz*XWinSz);
ctFN = 0;
for ctR = Dat.ImDcmp.YRange(1):Dat.ImDcmp.YRange(2)
    for ctC = Dat.ImDcmp.XRange(1):Dat.ImDcmp.XRange(2)
        ctF = ctC + (ctR-1)*Dat.Img.Xsteps;
        ctFN = ctFN +1;        FileNum(ctFN) = ctF-1;
    end
end
 % which planes to process
ZPlanes = 1;
if ~Dat.DatCnv.ZBinSw;    ZPlanes = Dat.ImDcmp.ZRange(1):Dat.ImDcmp.ZRange(2);end
 %% ====== main loop
FileNameConstR = [Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_'];
FileNameConstS = [Dat.Nom.MainDir,Dat.Nom.DASH];
FileNameConstW = [Dat.ImDcmp.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',Dat.ImDcmp.SvName,'_'];
PadNum = Dat.Nom.PadNum;
ImDcmpWinSw = Dat.ImDcmp.WinSw;ImDcmpCompRead{1} = Dat.ImDcmp.CompRead;
ImDcmpCompFName{1} = Dat.ImDcmp.CompFName;DomNum = 1;
ImDcmpDomName = 'all';
ImDcmpDomComp{1} = Dat.ImDcmp.CompName;ImDcmpDomCompNum = Dat.ImDcmp.CompNum;
SpcClbSHGch = Dat.SpcClb.SHGch;ImDcmpAlgor = Dat.ImDcmp.Algor;
ImDcmpSaveSw = Dat.ImDcmp.SaveSw;NumCh  = Dat.Img.ChNum;
ChUsed = Dat.Img.ChUsed;NumChUsed = length(ChUsed);
 RowNum = Dat.Img.NumRow;ColNum = Dat.Img.NumCol;
 ParProc = Dat.Proc.ParProc; clear Dat
 if ParProc %distributed computing enabled
    fprintf('\tdistributed computing enabled \n')
else
    fprintf('\tdistributed computing not enabled \n')
end
 %% ================== calculate normalized spectra matrix (row-dependent)
NumChUsed = length(ChUsed); SpectraRows = cell(DomNum,1);
UssRow = cell(DomNum,1);PseudoInvSSRow = cell(DomNum,1);
 if ParProc
     parfor ctD = 1:DomNum
        DomComp = ImDcmpDomComp{ctD};        DomCompNum = ImDcmpDomCompNum(ctD);
        DomCompRead = ImDcmpCompRead{ctD};        DomCompFName = ImDcmpCompFName{ctD};
         SpectraRows{ctD} = zeros(NumChUsed,DomCompNum,RowNum);
        UssRow{ctD} = zeros(NumChUsed,DomCompNum,RowNum);
        PseudoInvSSRow{ctD} = zeros(DomCompNum,NumChUsed,RowNum);
        for ctR = 1:RowNum % iterate over rows
            SS = zeros(16,DomCompNum);
            for ctE = 1:DomCompNum
                ElemID = DomComp{ctE};
                if DomCompRead(ctE)
                    FileName = [FileNameConstS,DomCompFName{ctE}];
                    TempSpec = LStackSpectrum(readconfocal16pics(FileName));
                    if strcmpi(ElemID,'collagen')
                        TempSpec(1:SpcClbSHGch) = 0;
                    end
                    SS(:,ctE) = TempSpec/sum(TempSpec);
```

```
                else
                    SS(:,ctE) = SpectrumEstim_2_3(ElemID,SpcEstData,1,ctR);
                end
            end
            % keep only the channels that contain data
            SS = SS(ChUsed,:);
            for ctE = 1:DomCompNum
                SS(:,ctE) = SS(:,ctE)/sum(SS(:,ctE));
            end
            SpectraRows{ctD}(:,:,ctR) = SS;
            % SVD decomposition
            [Uss,Sss,Vss]  = svd(SS);
            UssRow{ctD}(:,:,ctR)   = Uss(:,1:DomCompNum);
            % pseudoinverse
            PseudoInvSSRow{ctD}(:,:,ctR) =
Vss*diag(1./diag(Sss(1:DomCompNum,1:DomCompNum)))*(Uss(:,1:DomCompNum))';
        end
    end
else
    for ctD = 1:DomNum
        DomComp = ImDcmpDomComp{ctD};         DomCompNum = ImDcmpDomCompNum(ctD);
        DomCompRead = ImDcmpCompRead{ctD};
        DomCompFName = ImDcmpCompFName{ctD};
         SpectraRows{ctD} = zeros(NumChUsed,DomCompNum,RowNum);
        UssRow{ctD} = zeros(NumChUsed,DomCompNum,RowNum);
        PseudoInvSSRow{ctD} = zeros(DomCompNum,NumChUsed,RowNum);
         for ctR = 1:RowNum % iterate over rows
            SS = zeros(16,DomCompNum);
            for ctE = 1:DomCompNum
                ElemID = DomComp{ctE};
                if DomCompRead(ctE)
                    FileName = [FileNameConstS,DomCompFName{ctE}];
                    TempSpec = LStackSpectrum(readconfocal16pics(FileName));
                    if strcmpi(ElemID,'collagen')
                        TempSpec(1:SpcClbSHGch) = 0;
                    end
                    SS(:,ctE) = TempSpec/sum(TempSpec);
                else
                    SS(:,ctE) = SpectrumEstim_2_3(ElemID,SpcEstData,1,ctR);
                end
            end
            % keep only the channels that contain data
            SS = SS(ChUsed,:);
            for ctE = 1:DomCompNum
                SS(:,ctE) = SS(:,ctE)/sum(SS(:,ctE));
            end
            SpectraRows{ctD}(:,:,ctR) = SS;

            [Uss,Sss,Vss]  = svd(SS);  UssRow{ctD}(:,:,ctR)   = Uss(:,1:DomCompNum);
            PseudoInvSSRow{ctD}(:,:,ctR) = ...
      Vss*diag(1./diag(Sss(1:DomCompNum,1:DomCompNum)))*(Uss(:,1:DomCompNum))';
        end
    end
end
 %% ================== calculations for 1-photon counts
SinglePhotonDec = cell(DomNum);
for ctD = 1:DomNum
    DomCompNum = ImDcmpDomCompNum(ctD);
    SinglePhotonDec{ctD} = zeros(DomCompNum,NumChUsed,'single');
     SS = SpectraRows{ctD}(:,:,128);
    yy = eye(NumChUsed);
    % === linear model, common noise solution
    solbatch = fcnnls(SS,yy);
    do = SS*solbatch;
    % === linear model, individual noise solution
    % make sure that whn y[i]>0 then d[i]~=0
    do(do==0 & yy>0) = 0.1;     solbatch = zeros(DomCompNum,NumChUsed);
    for ctR = 1:NumChUsed
        % use only those channels for which do>0
        % THIS IS REQUIRED OTHERWISE lsqnonneg does not work well
        IndxNZ = find(do(:,ctR)>0);
```

```
        % NNLS
        solbatch(:,ctR) =
lsqnonneg(diag(1./sqrt(do(IndxNZ,ctR)))*SS(IndxNZ,:),diag(1./sqrt(do(IndxNZ,ctR)))*yy(IndxNZ,ctR)
);
    end
    SinglePhotonDec{ctD} = single(solbatch);
    end

% ==================== MAIN LOOP
if ParProc %distributed computing enabled
    parfor ctFF = 1:length(FileNum) % iterate for all z-stacks
        ctF = FileNum(ctFF);
        if mod(ctF,10) == 0;            fprintf('.');        end
         for ctZZ = 1:length(ZPlanes);
            ctZ = ZPlanes(ctZZ);
            FileName = [FileNameConstR,Padding(ctF,PadNum),'_',Padding(ctZ,PadNum)];
            RawFrame = LoadMatFile(FileName);
            RawFrame = RawFrame(:,:,ChUsed);
            if ImDcmpWinSw
                FrameMean = zeros(RowNum,ColNum,NumChUsed);
                for ctCH = 1:NumChUsed
FrameMean(:,:,ctCH) = imfilter(double(RawFrame(:,:,ctCH)),fspecial('average',3),'replicate');
                end
            end
            % === image decomposition per domain
            CompConMap = cell(DomNum,1);
            % loop over all domains
            for ctD = 1:DomNum
                DomCompNum = ImDcmpDomCompNum(ctD);
                ParamUnmix = cell(2,1);           ParamUnmix{1} = ImDcmpAlgor;
                ParamUnmix{2} = DomCompNum;
                CompConMap{ctD} = zeros(RowNum,ColNum,DomCompNum,'single');
                % iterate over rows
                for ctR = 1:RowNum
                    SS = SpectraRows{ctD}(:,:,ctR);   R = UssRow{ctD}(:,:,ctR);
                    PseudoInvSS = PseudoInvSSRow{ctD}(:,:,ctR);
                    if ImDcmpWinSw % use low-pass filtering
                        yy = squeeze(FrameMean(ctR,:,:))';
                    else
                        yy = squeeze(double(RawFrame(ctR,:,:)))';
                    end
                    yy = yy(ChUsed,:);                    yySUM = sum(yy);
                    yy0indx = find(yySUM==0);
                    yylindx = find(yySUM==1 & sum(yy==1)==1);
                    yyNZindx= setdiff(1:ColNum,[yy0indx,yylindx]);
                    % ===== ===== unmix pixels whose there are not photons
                    CompConMap{ctD}(ctR,yy0indx,:) = single(0);
                    % ===== ===== unmix pixels where just one channel contain 1 photom
                    for ctC = 1:length(yylindx)
                        indx1 = find(yy(:,yylindx(ctC))==1);
CompConMap{ctD}(ctR,yylindx(ctC),:) = SinglePhotonDec{ctD}(:,indx1);
                    end
                    % ===== ===== unmixing remaining pixels
                    yya = yy(:,yyNZindx);
CompConMap{ctD}(ctR,yyNZindx,:) =
SpectralUnmixing_1_0(yya,SS,PseudoInvSS,R,ParamUnmix);
                end % row
             end % domain
            if ImDcmpSaveSw
SaveFileDir = [FileNameConstW,Padding(ctF,PadNum),'_',Padding(ctZ,PadNum)];
               SaveData(CompConMap,SaveFileDir)
            end
        end % plane
    end % z-stack
else % distributed computing not enabled
    for ctFF = 1:length(FileNum) % iterate for all z-stacks
        ctF = FileNum(ctFF);
        if mod(ctF,10) == 0;            fprintf('.');        end
        for ctZZ = 1:length(ZPlanes);
            ctZ = ZPlanes(ctZZ);
             FileName = [FileNameConstR,Padding(ctF,PadNum),'_',Padding(ctZ,PadNum)];
```

```
            RawFrame = LoadMatFile(FileName);    RawFrame = RawFrame(:,:,ChUsed);
            if ImDcmpWinSw
                FrameMean = zeros(RowNum,ColNum,NumChUsed);
                for ctCH = 1:NumChUsed
FrameMean(:,:,ctCH) = imfilter(double(RawFrame(:,:,ctCH)),fspecial('average',3),'replicate');
                end
            end
               % === image decomposition per domain
            CompConMap = cell(DomNum,1);
            for ctD = 1:DomNum
                DomCompNum = ImDcmpDomCompNum(ctD);    ParamUnmix = cell(2,1);
                ParamUnmix{1} = ImDcmpAlgor;        ParamUnmix{2} = DomCompNum;
                CompConMap{ctD} = zeros(RowNum,ColNum,DomCompNum,'single');
                for ctR = 1:RowNum
                    SS = SpectraRows{ctD}(:,:,ctR);
                    R = UssRow{ctD}(:,:,ctR);
                    PseudoInvSS = PseudoInvSSRow{ctD}(:,:,ctR);
                    if ImDcmpWinSw % use low-pass filtering
                        yy = squeeze(FrameMean(ctR,:,:))';
                    else
                        yy = squeeze(double(RawFrame(ctR,:,:)))';
                    end
                    yy = yy(ChUsed,:);
                    yySUM = sum(yy);
                    yy0indx = find(yySUM==0);
                    yy1indx = find(yySUM==1 & sum(yy==1)==1);
                    yyNZindx= setdiff(1:ColNum,[yy0indx,yy1indx]);
                    % ===== ===== unmix pixels whose there are not photons
                    CompConMap{ctD}(ctR,yy0indx,:) = single(0);
                    % ===== ===== unmix pixels where just one channel contain 1 photom
                    for ctC = 1:length(yy1indx)
                        indx1 = find(yy(:,yy1indx(ctC))==1);
CompConMap{ctD}(ctR,yy1indx(ctC),:) = SinglePhotonDec{ctD}(:,indx1);
                    end
                    % ===== ===== unmixing remaining pixels
                    yya = yy(:,yyNZindx);
CompConMap{ctD}(ctR,yyNZindx,:) = SpectralUnmixing_1_0(yya,SS,PseudoInvSS,R,ParamUnmix);
                end % row
            end % domain
            % save results
            if ImDcmpSaveSw
SaveFileDir = [FileNameConstW,Padding(ctF,PadNum),'_',Padding(ctZ,PadNum)];
               save(SaveFileDir,'CompConMap')
            end
        end % plane
    end % z-stack
end
 fprintf('\t... done\n\n')
 return
function Err = funML(x,Param)
Err = -ones(Param{3},1);x = x(:);
for cti = 1:Param{3}
    for ctCH = 1:length(Param{1})
        Err(cti) = Err(cti) + Param{1}(ctCH)*Param{2}(ctCH,cti)/(Param{2}(ctCH,:)*x);
    end
end
 function RawFrame = LoadMatFile(FileName)
load(FileName)
return
 function y = SaveData(CompConMap,SaveFileDir)
save(SaveFileDir,'CompConMap')
return

%% ============================ fast NNLS ============================
function [K, Pset] = fcnnls(C, A)
% NNLS using normal equations and the fast combinatorial strategy
% I/O: [K, Pset] = fcnnls(C, A);
% K = fcnnls(C, A);
% C is the nObs Ã, lVar coefficient matrix
% A is the nObs Ã, pRHS matrix of observations
% K is the lVar Ã, pRHS solution matrix
```

```
% Pset is the lVar Ã, pRHS passive set logical array
% M. H. Van Benthem and M. R. Keenan
% Sandia National Laboratories
% Pset: set of passive sets, one for each column
% Fset: set of column indices for solutions that have not yet converged
% Hset: set of column indices for currently infeasible solutions
% Jset: working set of column indices for currently optimal solutions
% slight modifications (see %D% sign) by Dimitris Tzeranis, Feb 2012
error(nargchk(2,2,nargin))
 [nObs, lVar] = size(C);
if size(A,1)~= nObs, error('C and A have imcompatible sizes'), end
pRHS = size(A,2);W = zeros(lVar, pRHS);iter = 0; maxiter = 3*lVar;
 % Precompute parts of pseudoinverse
CtC = C'*C; CtA = C'*A;
 % Obtain the initial feasible solution and corresponding passive set
K = cssls(CtC, CtA);Pset = K > 0;K(~Pset) = 0;D = K;Fset = find(~all(Pset));
 % Active set algorithm for NNLS main loop
while ~isempty(Fset)
    % Solve for the passive variables (uses subroutine below)
    K(:,Fset) = cssls(CtC, CtA(:,Fset), Pset(:,Fset));
    % Find any infeasible solutions
    Hset = Fset(find(any(K(:,Fset) < 0)));
    % Make infeasible solutions feasible (standard NNLS inner loop)
    if ~isempty(Hset)
        nHset = length(Hset);        alpha = zeros(lVar, nHset);
        while ~isempty(Hset) & (iter < maxiter)
            iter = iter + 1;      alpha(:,1:nHset) = Inf;
            % Find indices of negative variables in passive set
            [i, j] = find(Pset(:,Hset) & (K(:,Hset) < 0));
            hIdx = sub2ind([lVar nHset], i, j);            temp = Hset(j)';%D%
            negIdx = sub2ind(size(K), i(:), temp(:)); %D%
            alpha(hIdx) = D(negIdx)./(D(negIdx) - K(negIdx));
            [alphaMin,minIdx] = min(alpha(:,1:nHset));
            alpha(:,1:nHset) = repmat(alphaMin, lVar, 1);
            D(:,Hset) = D(:,Hset)-alpha(:,1:nHset).*(D(:,Hset)-K(:,Hset));
            idx2zero = sub2ind(size(D), minIdx, Hset);
            D(idx2zero) = 0;            Pset(idx2zero) = 0;
            K(:, Hset) = cssls(CtC, CtA(:,Hset), Pset(:,Hset));
            Hset = find(any(K < 0)); nHset = length(Hset);
        end
    end
     % Make sure the solution has converged
    if iter == maxiter, error('Maximum number iterations exceeded'), end
    % Check solutions for optimality
W(:,Fset) = CtA(:,Fset)-CtC*K(:,Fset);Jset = find(all(~Pset(:,Fset).*W(:,Fset) <= 0));
    Fset = setdiff(Fset, Fset(Jset));
    % For non-optimal solutions, add the appropriate variable to Pset
    if ~isempty(Fset)
        [mx, mxidx] = max(~Pset(:,Fset).*W(:,Fset));
        Pset(sub2ind([lVar pRHS], mxidx, Fset)) = 1;    D(:,Fset) = K(:,Fset);
    end
end
return

function [K] = cssls(CtC, CtA, Pset)
% Solve the set of equations CtA = CtC*K for the variables in set Pset
% using the fast combinatorial approach
 K = zeros(size(CtA));
if (nargin == 2) || isempty(Pset) || all(Pset(:))
    K = CtC\CtA;
else
    [lVar pRHS] = size(Pset);
    codedPset = 2.^(lVar-1:-1:0)*Pset;    [sortedPset, sortedEset] = sort(codedPset);
    breaks = diff(sortedPset);    breakIdx = [0 find(breaks) pRHS];
    for k = 1:length(breakIdx)-1
        cols2solve = sortedEset(breakIdx(k)+1:breakIdx(k+1));
        vars = Pset(:,sortedEset(breakIdx(k)+1));
        K(vars,cols2solve) = CtC(vars,vars)\CtA(vars,cols2solve);
    end
end
 return
```

ImAssembly

```
function ZPlanes = ImAssembly_1_5(Dat)
% assemble decomposed images based on the optimum image registration
% Dimitrios Tzeranis, Nov 2010 - July 2011
fprintf('=== Assemble decomposed images ===\n')
 %% === initialization
[unused1,unused2,unused3] = mkdir(Dat.ImAsbl.Dir);
 ZPlanes = Dat.ImAsbl.ZRange(1):Dat.ImAsbl.ZRange(2); DomNum = 1;% print "all"
 %% === assemble images
fprintf('\tPlane:')
for ctZ = ZPlanes
    fprintf(' %s',num2str(ctZ))
    AsblCompConMap = cell(1,Dat.ImDcmp.DomNum);
    if (Dat.Img.Xsteps==1) && (Dat.Img.Ysteps==1)    % case of a z-stack
        load([Dat.ImDcmp.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
              Dat.ImDcmp.SvName,'_',Padding(0,Dat.Nom.PadNum),'_',...
              Padding(ctZ,Dat.Nom.PadNum)]);   % read "CompConMap"
        for ctD = 1:1%iterate over domains
            if  Dat.ImReg.RemLSw
AsblCompConMap{ctD} = CompConMap{ctD}(Dat.ImReg.RemLPixNum + 1:end,:,:);
            end
        end
    else % general case where I need to assemble the component maps
        Dat.ImAsbl.XRange = Dat.ImReg.XRange;
        Dat.ImAsbl.YRange = Dat.ImReg.YRange;
        XWinSz = Dat.ImAsbl.XRange(2) - Dat.ImAsbl.XRange(1) + 1;
        YWinSz = Dat.ImAsbl.YRange(2) - Dat.ImAsbl.YRange(1) + 1;
        FileNum = zeros(1,YWinSz*XWinSz);         ctFN = 0;
        for ctR = Dat.ImAsbl.YRange(1):Dat.ImAsbl.YRange(2)
            for ctC = Dat.ImAsbl.XRange(1):Dat.ImAsbl.XRange(2)
                ctF = ctC + (ctR-1)*Dat.Img.Xsteps;
                ctFN = ctFN +1;     FileNum(ctFN) = ctF;
            end
        end
        load([Dat.ImReg.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_ImageRegOverview']);
        posiO = RegImagDat.posiO;        dispO = RegImagDat.dispO;
        insertO = RegImagDat.insertO;        rangeO = RegImagDat.rangeO;
        clear RegImagDat
        maxR = max(rangeO(1:2:end));        minR = min(insertO(1:2:end));
        maxC = max(rangeO(2:2:end));        minC = min(insertO(2:2:end));
        if minR~=1
            TotalPixX = maxR - minR + 1;
            rangeO(1:2:end) = rangeO(1:2:end) - minR + 1;
            insertO(1:2:end) = insertO(1:2:end) - minR + 1;
        else
            TotalPixX = maxR;
        end
        TotalPixY = maxC;
        if minC<=0
            TotalPixY = TotalPixY - minC + 1;
            rangeO(2:2:end) = rangeO(2:2:end) - minC + 1;
            insertO(2:2:end) = insertO(2:2:end) - minC + 1;
        end
        for ctD = 1:1   % this is for printing only preliminary
            AsblCompConMap{ctD} = zeros(TotalPixY,TotalPixX,Dat.ImDcmp.CompNum);
            for ctF = 1:XWinSz*YWinSz % for all snaps
                ctR = floor((ctF-1)/XWinSz) + 1;
                ctC = ctF - (ctR-1)*XWinSz;
                addF = [2*ctF-1,2*ctF];
                dO = dispO(addF);
load([Dat.ImDcmp.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',
 Dat.ImDcmp.SvName,'_',Padding(FileNum(ctF)-1,Dat.Nom.PadNum),'_',
(ctZ,Dat.Nom.PadNum)]);                    DecImF = CompConMap{ctD};
  if  Dat.ImReg.RemLSw; DecImF = DecImF(:,Dat.ImReg.RemLPixNum + 1:end,:);   end
                dimF = size(DecImF);
                if (ctR == 1) && (ctC == 1)
                    %
                elseif (ctR == 1) && (ctC > 1)
                    % === L-R
```

```
                    ctL = ctF - 1;
                    load([Dat.ImDcmp.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
                        Dat.ImDcmp.SvName,'_',Padding(FileNum(ctL)-1,Dat.Nom.PadNum),'_',
Padding(ctZ,Dat.Nom.PadNum)]);
                    DecImL = CompConMap{ctD};
if  Dat.ImReg.RemLSw;DecImL = DecImL(:,Dat.ImReg.RemLPixNum + 1:end,:);end
                    dimL = size(DecImL);
                    [cs1,cs2] = CalculateWindowCS2(dimL,dimF,1,dO);
          MaskLR = (linspace(0,1,cs1(4)-cs1(3)+1)'*ones(1,cs1(2)-cs1(1)+1))';
                    for ctE = 1:Dat.ImDcmp.DomCompNum(ctD)
    DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE) = round((1-
MaskLR).*double(DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE)) +
MaskLR.*double(DecImL(cs1(1):cs1(2),cs1(3):cs1(4),ctE)));
                    end
                elseif (ctR > 1) && (ctC == 1)
                    % === T-B
                    ctT = ctF - XWinSz;

load([Dat.ImDcmp.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',Dat.ImDcmp.SvName,'_',Padding(FileNum(ctT)-
1,Dat.Nom.PadNum),'_', Padding(ctZ,Dat.Nom.PadNum)]);
                    DecImT = CompConMap{ctD};
if  Dat.ImReg.RemLSw;DecImT = DecImT(:,Dat.ImReg.RemLPixNum + 1:end,:);end
                    dimT = size(DecImT);
                    [cs1,cs2] = CalculateWindowCS2(dimT,dimF,2,dO);
                    MaskTB = (ones(cs1(4)-cs1(3)+1,1)*linspace(0,1,cs1(2)-cs1(1)+1))';
                    for ctE = 1:Dat.ImDcmp.DomCompNum(ctD)
  DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE) =
round((MaskTB).*double(DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE)) + (1-
MaskTB).*double(DecImT(cs1(1):cs1(2),cs1(3):cs1(4),ctE)));
                    end
                else
                    % === L-R
                    ctL = ctF - 1;
load([Dat.ImDcmp.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',
.ImDcmp.SvName,'_',Padding(FileNum(ctL)-1,Dat.Nom.PadNum),'_',
 Padding(ctZ,Dat.Nom.PadNum)]);
                    DecImL = CompConMap{ctD};
if  Dat.ImReg.RemLSw;DecImL = DecImL(:,Dat.ImReg.RemLPixNum + 1:end,:);end
                    dimL = size(DecImL);
                    [cs1,cs2] = CalculateWindowCS2(dimL,dimF,1,dO);
MaskLR = (linspace(0,1,cs1(4)-cs1(3)+1)'*ones(1,cs1(2)-cs1(1)+1))';
                    for ctE = 1:Dat.ImDcmp.DomCompNum(ctD)
DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE) = round((1-
MaskLR).*double(DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE)) +
.*double(DecImL(cs1(1):cs1(2),cs1(3):cs1(4),ctE)));
                    end
                    % === T-B
                    ctT = ctF - XWinSz;          addT = [2*ctT-1,2*ctT];
                    dO =  posiO(addF) - posiO(addT);
load([Dat.ImDcmp.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',
 Dat.ImDcmp.SvName,'_',Padding(FileNum(ctT)-1,Dat.Nom.PadNum),'_',Padding(ctZ,Dat.Nom.PadNum)]);
                    DecImT = CompConMap{ctD};
   if  Dat.ImReg.RemLSw;DecImT = DecImT(:,Dat.ImReg.RemLPixNum + 1:end,:);end
                    dimT = size(DecImT);
                    [cs1,cs2] = CalculateWindowCS2(dimT,dimF,2,dO);
                    MaskTB = (ones(cs1(4)-cs1(3)+1,1)*linspace(0,1,cs1(2)-cs1(1)+1))';
                    for ctE = 1:Dat.ImDcmp.DomCompNum(ctD)
DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE) =
round((MaskTB).*double(DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE)) + (1-
MaskTB).*double(DecImT(cs1(1):cs1(2),cs1(3):cs1(4),ctE)));
                    end
                end
AsblCompConMap{ctD}(insertO(addF(2)):rangeO(addF(2)),insertO(addF(1)):rangeO(addF(1)),:) =
DecImF;
            end
            AsblCompConMap{ctD} = single(AsblCompConMap{ctD});

        end % domain
    end % z-stack versus assembly
        % save data
    if Dat.ImAsbl.SaveSw
```

```
        SaveFileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
            Dat.ImAsbl.SvName,'_',Padding(ctZ,Dat.Nom.PadNum)];
        save(SaveFileDir,'AsblCompConMap')
    end
end % z plane
fprintf('\n') fprintf('\t\t..... done\n\n')
return
 function [cs1,cs2] = CalculateWindowCS2(d1,d2,Dim,d)
% coordinates of the windows in the base image (B) and the image (F) that
% is positioned with respect to B @ displacement d=[dx,dy]' along dimension
% Dim
% Im1:  base image (R or T)
%   d1 = size(Im1);
% Im2:  image that needs to be positioned (L or B)
%   d2 = size(Im2);
% Dimitris Tzeranis, February 2009, November 2010
 if Dim == 1 % R-L
    if d(2)>=0
        cs2 = [d(2)+1,d2(1),d2(2)-d(1)+1,d2(2)];
        cs1 = [1,d1(1)-d(2),1,d(1)];
    else
        cs2 = [1,d2(1)+d(2),d2(2)-d(1)+1,d2(2)];
        cs1 = [1-d(2),d1(1),1,d(1)];
    end
else % T-B
    if d(1)>=0
        cs2 = [1,d(2),1,d2(2)-d(1)];
        cs1 = [d1(1)-d(2)+1,d1(1),d(1)+1,d1(2)];
    else
        cs2 = [1,d(2),1-d(1),d2(2)];
        cs1 = [d1(1)-d(2)+1,d1(1),1,d1(2)+d(1)];
    end
end
 return
```

GMaxIPr

```
function ZPlanes = GMaxIPr_1_1(Dat)
% assemble the max{intensity} image projection
% Dimitrios Tzeranis, Yannas Lab, So Lab, June 2011 - June 2013
fprintf('\tCalculating the maximum Intensity Projection')
%% === initialization
 [unused1,unused2,unused3] = mkdir(Dat.GMaxIPr.Dir);
 ZPlanes = Dat.GMaxIPr.ZRange(1):Dat.GMaxIPr.ZRange(2);
 if isempty(Dat.Img.Instr);    Dat.Img.Instr = 'original'; end
 %% === generate the image that contains the maximum planes
 if (Dat.Img.Xsteps==1) && (Dat.Img.Ysteps==1)    % case of a z-stack
    if Dat.Img.Zsteps==1
        MaxIData = ones(256,256,'uint8');
    else
        load([Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_MaxIntPlane_',...
            Padding(0,Dat.Nom.PadNum)]); % loads MaxIntPlane
        MaxIData = MaxIntPlane;
    end
    % coordinate matrix
    MaxICrds = cat(3,zeros(256,'uint8'),uint8(repmat([1:256]',[1 256])),uint8(repmat([1:256],[256
1])));
else
    % size of the assembled frames
    Dat.ImAsbl.XRange = Dat.ImReg.XRange;
    Dat.ImAsbl.YRange = Dat.ImReg.YRange;
    XWinSz = Dat.ImAsbl.XRange(2) - Dat.ImAsbl.XRange(1) + 1;
    YWinSz = Dat.ImAsbl.YRange(2) - Dat.ImAsbl.YRange(1) + 1;
     % identify the frame numbers to be processed, store in FileNum vector
    FileNum = zeros(1,YWinSz*XWinSz);    ctFN = 0;
    for ctR = Dat.ImAsbl.YRange(1):Dat.ImAsbl.YRange(2)
        for ctC = Dat.ImAsbl.XRange(1):Dat.ImAsbl.XRange(2)
            ctF = ctC + (ctR-1)*Dat.Img.Xsteps;
            ctFN = ctFN +1;            FileNum(ctFN) = ctF;
        end
    end
```

```
    % find the range of values of the insert and range matrices elements
    load([Dat.ImReg.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_ImageRegOverview']);
    posiO = RegImagDat.posiO;    dispO = RegImagDat.dispO;
    insertO = RegImagDat.insertO;     rangeO = RegImagDat.rangeO;
    clear RegImagDat
     maxR = max(rangeO(1:2:end));    minR = min(insertO(1:2:end));
    maxC = max(rangeO(2:2:end));    minC = min(insertO(2:2:end));
    if minR~=1
        TotalPixX = maxR - minR + 1;
        rangeO(1:2:end) = rangeO(1:2:end) - minR + 1;
        insertO(1:2:end) = insertO(1:2:end) - minR + 1;
    else
        TotalPixX = maxR;
    end
    TotalPixY = maxC;
    if minC<=0
        TotalPixY = TotalPixY - minC + 1;
        rangeO(2:2:end) = rangeO(2:2:end) - minC + 1;
        insertO(2:2:end) = insertO(2:2:end) - minC + 1;
    end
    MaxIData = zeros(TotalPixY,TotalPixX);
    MaxICrds = zeros(TotalPixY,TotalPixX,3,'uint8');
MaxICrdsTMP = cat(3,uint8(repmat([1:256]',[1 256])),uint8(repmat([1:256],[256 1])));
    for ctF = 1:XWinSz*YWinSz % for all snaps
        ctR = floor((ctF-1)/XWinSz) + 1;        ctC = ctF - (ctR-1)*XWinSz;
        addF = [2*ctF-1,2*ctF];        dO = dispO(addF);
        % read DecImF
        load([Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_MaxIntPlane_',...
            Padding(FileNum(ctF)-1,Dat.Nom.PadNum)]); % loads MaxIntPlane
        DecImF = MaxIntPlane;
        CrdsF = cat(3,ones(256,'uint8')*uint8(FileNum(ctF)-1),MaxICrdsTMP);
        if  Dat.ImReg.RemLSw
            if strcmpi(Dat.Img.Instr,'MMM')
                DecImF = DecImF(8:end,:,:);   CrdsF = CrdsF(8:end,:,:);
            else
                % original microscope: remove left pixels
                DecImF = DecImF(:,Dat.ImReg.RemLPixNum + 1:end,:);
                CrdsF = CrdsF(:,Dat.ImReg.RemLPixNum + 1:end,:);
            end
        end
        dimF = size(DecImF);
        if (ctR == 1) && (ctC == 1)
        elseif (ctR == 1) && (ctC > 1)
            % === L-R
            ctL = ctF - 1;
            load([Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_MaxIntPlane_',...
                Padding(FileNum(ctL)-1,Dat.Nom.PadNum)]); % loads MaxIntPlane
            DecImL = MaxIntPlane;
            % remove left pixels
            if  Dat.ImReg.RemLSw
                if strcmpi(Dat.Img.Instr,'MMM')
                    DecImL = DecImL(8:end,:,:);
                else
                    % original microscope: remove left pixels
                    DecImL = DecImL(:,Dat.ImReg.RemLPixNum + 1:end,:);
                end
            end
            dimL = size(DecImL);
            [cs1,cs2] = CalculateWindowCS2(dimL,dimF,1,dO);
            temp = zeros(cs2(2)-cs2(1)+1,cs2(4)-cs2(3)+1,2);
            temp(:,:,1) = DecImF(cs2(1):cs2(2),cs2(3):cs2(4));
            temp(:,:,2) = DecImL(cs1(1):cs1(2),cs1(3):cs1(4));
            DecImF(cs2(1):cs2(2),cs2(3):cs2(4)) = round(mean(temp,3));
        elseif (ctR > 1) && (ctC == 1)
            % === T-B
            ctT = ctF - XWinSz;
            load([Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_MaxIntPlane_',...
                Padding(FileNum(ctT)-1,Dat.Nom.PadNum)]); % loads MaxIntPlane
            DecImT = MaxIntPlane;
            if  Dat.ImReg.RemLSw
                if strcmpi(Dat.Img.Instr,'MMM')
```

```
                DecImT = DecImT(8:end,:,:);
            else
                % original microscope: remove left pixels
                DecImT = DecImT(:,Dat.ImReg.RemLPixNum + 1:end,:);
            end
        end
        dimT = size(DecImT);
        [cs1,cs2] = CalculateWindowCS2(dimT,dimF,2,dO);
        temp = zeros(cs2(2)-cs2(1)+1,cs2(4)-cs2(3)+1,2);
        temp(:,:,1) = DecImF(cs2(1):cs2(2),cs2(3):cs2(4));
        temp(:,:,2) = DecImT(cs1(1):cs1(2),cs1(3):cs1(4));
        DecImF(cs2(1):cs2(2),cs2(3):cs2(4)) = round(mean(temp,3));
    else
        % === L-R
        ctL = ctF - 1;
        load([Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_MaxIntPlane_',...
            Padding(FileNum(ctL)-1,Dat.Nom.PadNum)]); % loads MaxIntPlane
        DecImL = MaxIntPlane;
        if  Dat.ImReg.RemLSw
            if strcmpi(Dat.Img.Instr,'MMM')
                DecImL = DecImL(8:end,:,:);
            else
                % original microscope: remove left pixels
                DecImL = DecImL(:,Dat.ImReg.RemLPixNum + 1:end,:);
            end
        end
        dimL = size(DecImL);
        [cs1,cs2] = CalculateWindowCS2(dimL,dimF,1,dO);
        temp = zeros(cs2(2)-cs2(1)+1,cs2(4)-cs2(3)+1,2);
        temp(:,:,1) = DecImF(cs2(1):cs2(2),cs2(3):cs2(4));
        temp(:,:,2) = DecImL(cs1(1):cs1(2),cs1(3):cs1(4));
        DecImF(cs2(1):cs2(2),cs2(3):cs2(4)) = round(mean(temp,3));
          % === T-B
        ctT = ctF - XWinSz;
        addT = [2*ctT-1,2*ctT];
        dO =  posiO(addF) - posiO(addT);
        load([Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_MaxIntPlane_',...
            Padding(FileNum(ctT)-1,Dat.Nom.PadNum)]); % loads MaxIntPlane
        DecImT = MaxIntPlane;
        if  Dat.ImReg.RemLSw
            if strcmpi(Dat.Img.Instr,'MMM')
                DecImT = DecImT(8:end,:,:);
            else
                % original microscope: remove left pixels
                DecImT = DecImT(:,Dat.ImReg.RemLPixNum + 1:end,:);
            end
        end
        dimT = size(DecImT);
        [cs1,cs2] = CalculateWindowCS2(dimT,dimF,2,dO);
        temp = zeros(cs2(2)-cs2(1)+1,cs2(4)-cs2(3)+1,2);
        temp(:,:,1) = DecImF(cs2(1):cs2(2),cs2(3):cs2(4));
        temp(:,:,2) = DecImT(cs1(1):cs1(2),cs1(3):cs1(4));
        DecImF(cs2(1):cs2(2),cs2(3):cs2(4)) = round(mean(temp,3));
    end
            MaxIData(insertO(addF(2)):rangeO(addF(2)),insertO(addF(1)):rangeO(addF(1)),:) =
DecImF;
MaxICrds(insertO(addF(2)):rangeO(addF(2)),insertO(addF(1)):rangeO(addF(1)),:) = CrdsF;
    end
end
MaxIData = round(medfilt2(double(MaxIData),[Dat.GMaxIPr.MedFSz Dat.GMaxIPr.MedFSz],'symmetric'));
RowNum = size(MaxIData,1);
ColNum = size(MaxIData,2);
MaxIntImage = cell(1,Dat.ImDcmp.DomNum);
for ctD = 1:(Dat.ImDcmp.DomNum)
    MaxIntImage{ctD} = zeros(RowNum,ColNum,Dat.ImDcmp.DomCompNum(ctD));
end
 % generate the maximum projection
for ctZ = ZPlanes
    PxList = find(MaxIData==ctZ);
    if ~isempty(PxList)
        ColList = floor((PxList-1)/RowNum) + 1;
```

```
        RowList = PxList - (ColList-1)*RowNum;
        load([Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
            Dat.ImAsbl.SvName,'_',Padding(ctZ,Dat.Nom.PadNum)]);
        for ctD = 1:(Dat.ImDcmp.DomNum)
            for ctF = 1:length(PxList)
MaxIntImage{ctD}(RowList(ctF),ColList(ctF),:) = AsblCompConMap{1}(RowList(ctF),ColList(ctF),ctD);
            end
        end
    end
end
 % save data
if Dat.GMaxIPr.SaveSw
    save([Dat.GMaxIPr.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',Dat.GMaxIPr.SvName],...
        'MaxIData','MaxIntImage','MaxICrds')
end
 fprintf('\n\t\t..... done\n\n')
return
 function [cs1,cs2] = CalculateWindowCS2(d1,d2,Dim,d)
% coordinates of the windows in the base image (B) and the image (F) that
% is positioned with respect to B @ displacement d=[dx,dy]' along dimension
% Dim
% Im1:  base image (R or T)
%   d1 = size(Im1);
% Im2:  image that needs to be positioned (L or B)
%   d2 = size(Im2);
% Dimitris Tzeranis, February 2009, November 2010
 if Dim == 1 % R-L
    if d(2)>=0
        cs2 = [d(2)+1,d2(1),d2(2)-d(1)+1,d2(2)];
        cs1 = [1,d1(1)-d(2),1,d(1)];
    else
        cs2 = [1,d2(1)+d(2),d2(2)-d(1)+1,d2(2)];
        cs1 = [1-d(2),d1(1),1,d(1)];
    end
else % T-B
    if d(1)>=0
        cs2 = [1,d(2),1,d2(2)-d(1)];
        cs1 = [d1(1)-d(2)+1,d1(1),d(1)+1,d1(2)];
    else
        cs2 = [1,d(2),1-d(1),d2(2)];
        cs1 = [d1(1)-d(2)+1,d1(1),1,d1(2)+d(1)];
    end
end
 return
```

ImPrint

```
function ZPlanes = ImPrint_1_4(Dat)
% print the assembled decomposed planes or maximum intensity projection
% Dimitrios Tzeranis, June 2011 - august 2012
fprintf('=== Printing assembled decomposed images ===\n')
 if Dat.Proc.ParProc & Dat.ImAsbl.ImPrintHSV
    fprintf('\t\tapply distributed computing\n')
end
    %% === initialization
ZPlanes2Print = Dat.ImAsbl.ImPrintZRange(1):Dat.ImAsbl.ImPrintZRange(2);
ZPlanes = [];
 RGBpallete = [ 1 0 0;  0 1 0 ;  0 0 1 ;  1 1 0 ;  0.5 0 1 ;  1 0.5 0 ;  0 1 1];
HSVpallete = zeros(7,1);
for ctR = 1:7
    temp = rgb2hsv(RGBpallete(ctR,:)); HSVpallete(ctR) = temp(1); %save only H value
end
if ~isfield(Dat.ImAsbl,'ImPrintScale')
    Dat.ImAsbl.ImPrintScale = 100;
else
    if isempty(Dat.ImAsbl.ImPrintScale);   Dat.ImAsbl.ImPrintScale = 100;   end
end
 if isempty(Dat.ImAsbl.ImPrintComp);  Dat.ImAsbl.ImPrintComp = false; end
 if isempty(Dat.ImAsbl.ImPrintCmpnd);     Dat.ImAsbl.ImPrintCmpnd = false; end
%% === print images
fprintf('\n\tPlane:')
```

```
for ctZ = ZPlanes2Print
    fprintf(' %s',num2str(ctZ))
    if ctZ==0
FileDir = [Dat.GMaxIPr.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',Dat.GMaxIPr.SvName,'.mat'];
    else
        FileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
            Dat.ImAsbl.SvName,'_',Padding(ctZ,Dat.Nom.PadNum),'.mat'];
    end
    if ~isempty(ls(FileDir))
        ZPlanes = [ZPlanes,ctZ];          load(FileDir)
        if ctZ==0
            Image2Print = MaxIntImage;
            clear MaxIData MaxIntImage
        else
            Image2Print = AsblCompConMap;
            clear AsblCompConMap
        end
    else
        disp(['this file was not found: ',FileDir])
    end
    NumberRows = size(Image2Print{1},1);      NumberCols = size(Image2Print{1},2);
    for ctD = 1 % :Dat.ImDcmp.DomNum %FOCUS ON THE FIRST DOMAIN
        VisThr  = Dat.ImAsbl.VisThr{ctD};       VisChan = Dat.ImAsbl.VisChan{ctD};
              if Dat.ImAsbl.ImPrintHSV
            % generate the HSV matrix and then convert to RGB
            Indx =  find(VisChan>0);              Ch2Vis = VisChan(Indx);
            Th2Use = VisThr(:,Indx);              colorsi = HSVpallete(Indx);
Im2Process = reshape(Image2Print{ctD}(:,:,Ch2Vis),[NumberRows*NumberCols,length(Ch2Vis)]);
            for ctCH = 1:length(Ch2Vis)
Im2Process(:,ctCH) = (Im2Process(:,ctCH)-Th2Use(2,ctCH))/(Th2Use(1,ctCH)-Th2Use(2,ctCH));
                % print individual components
                if Dat.ImAsbl.ImPrintComp
CompIm2Print = reshape(Im2Process(:,ctCH),[NumberRows,NumberCols]);
CompIm2Print(CompIm2Print<0) = 0;    CompIm2Print(CompIm2Print>1) = 1;
                    ElemName = ['_E',num2str(VisChan(ctCH)),...
                        'T',num2str(VisThr(2,ctCH)),'-',num2str(VisThr(1,ctCH))];
SaveFileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
Dat.ImAsbl.SvName,'_z',Padding(ctZ,Dat.Nom.PadNum),'_Dom',num2str(ctD),ElemName];
   if (Dat.ImAsbl.ImPrintScale==100)
                        imwrite(CompIm2Print,[SaveFileDir,'.png'],'png')
                    else
                        imwrite(imresize(CompIm2Print,Dat.ImAsbl.ImPrintScale/100),...
 [SaveFileDir,'_sc',num2str(Dat.ImAsbl.ImPrintScale),'.png'],'png')
                    end
                end
            end
            % print compound images
            if Dat.ImAsbl.ImPrintCmpnd
                [maxIm2Process,maxIm2PrIndx] = max(Im2Process,[],2);
maxIm2Process(maxIm2Process<0) = 0;    maxIm2Process(maxIm2Process>1) = 1;
                % convert to RGB
Im2PrintHSV = hsv2rgbDIM([colorsi(maxIm2PrIndx),ones(length(maxIm2PrIndx),1),maxIm2Process(:)]);
                Im2PrintHSV = reshape(Im2PrintHSV,[NumberRows,NumberCols,3]);
                 ElemName = '_';
                for ctE = 1:length(Ch2Vis)
                    ElemName = [ElemName,'E',num2str(Ch2Vis(ctE))];
                    ElemName = [ElemName,'T',num2str(VisThr(2,ctE)),'-',num2str(VisThr(1,ctE))];
                end
SaveFileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
Dat.ImAsbl.SvName,'_z',Padding(ctZ,Dat.Nom.PadNum),'_Dom',num2str(ctD),ElemName,'HSV'];
                if (Dat.ImAsbl.ImPrintScale==100)
                    imwrite(Im2PrintHSV,[SaveFileDir,'.png'],'png')
                else
                    imwrite(imresize(Im2PrintHSV,Dat.ImAsbl.ImPrintScale/100),...
[SaveFileDir,'_sc',num2str(Dat.ImAsbl.ImPrintScale),'.png'],'png')
                end
            end
        else %RGB
            Im2PrintRGB = zeros(NumberRows,NumberCols,3);
            for ctCH = 1:3
if VisChan(ctCH) && (VisThr(1,ctCH)>=0) && (VisThr(2,ctCH)>=0) && (VisThr(1,ctCH)-VisThr(2,ctCH))
```

```
Im2PrintRGB(:,:,ctCH) = (double(Image2Print{ctD}(:,:,VisChan(ctCH)))-
VisThr(2,ctCH))/(VisThr(1,ctCH)-VisThr(2,ctCH));
                end
            end
            Im2PrintRGB(Im2PrintRGB>1) = 1;     Im2PrintRGB(Im2PrintRGB<0) = 0;
            if Dat.ImAsbl.ImPrintComp
                for ctCH = 1:3
ElemName = ['_E',num2str(VisChan(ctCH)),'T',num2str(VisThr(2,ctCH)),'-',num2str(VisThr(1,ctCH))];
                    SaveFileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
Dat.ImAsbl.SvName,'_z',Padding(ctZ,Dat.Nom.PadNum),'_Dom',num2str(ctD),ElemName];
                    if (Dat.ImAsbl.ImPrintScale==100)
                        imwrite(Im2PrintRGB(:,:,ctCH),[SaveFileDir,'.png'],'png')
                    else
imwrite(imresize(Im2PrintRGB(:,:,ctCH),Dat.ImAsbl.ImPrintScale/100),...
[SaveFileDir,'_sc',num2str(Dat.ImAsbl.ImPrintScale),'.png'],'png')
                    end
                end
            end
            % print compound images
            if Dat.ImAsbl.ImPrintCmpnd
                ElemName = '_';
                for ctE = 1:3
                    ElemName = [ElemName,'E',num2str(VisChan(ctE))];
                    ElemName = [ElemName,'T',num2str(VisThr(2,ctE)),'-',num2str(VisThr(1,ctE))];
                end
                SaveFileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
Dat.ImAsbl.SvName,'_z',Padding(ctZ,Dat.Nom.PadNum),'_Dom',num2str(ctD),ElemName];
                if (Dat.ImAsbl.ImPrintScale==100)
                    imwrite(Im2PrintRGB,[SaveFileDir,'.png'],'png')
                else
imwrite(imresize(Im2PrintRGB,Dat.ImAsbl.ImPrintScale/100),[SaveFileDir,'_sc',num2str(Dat.ImAsbl.I
mPrintScale),'.png'],'png')
                end
            end
        end
    end
end % z plane
fprintf('\n')
fprintf('\t\t..... done\n\n')
return
function [rout,g,b] = hsv2rgbDIM(hin)
h = 6.*hin(:,1); s = hin(:,2);v = hin(:,3); k = floor(h);
 rout = zeros(size(hin,1),3,'single');
rout(:,1) = (k==0 | k==6 | k==5) + (k==1).*(1-s.*(h-k)) + (k==2 | k==3).*(1-s)  + (k==4).*(1-
s.*(1-h+k));
rout(:,2) = (k==0 | k==6).*(1-s.*(1-h+k)) + (k==1 | k==2) + (k==3).*(1-s.*(h-k)) + (k==4 |
k==5).*(1-s);
rout(:,3) = (k==0 | k==6 | k==1).*(1-s) + (k==2).*(1-s.*(1-h+k)) + (k==3 | k==4)    + (k==5).*(1-
s.*(h-k));
 rout = bsxfun(@times, v./max(rout(:)), rout);
return
```

DomEvidenceCalc

```
function ZPlanes = DomEvidenceCalc_2_2(Dat)
% Image segmentation using maximum a-posteriori model selection
% Dimitris Tzeranis, August 2012 - June 2013
fprintf('=== Calculating Domain Evidence ===\n')

Ns = 100; NumSig = 2;
% load spectrum estimation data
FileDirSpcClb = [Dat.SpcClb.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',Dat.SpcClb.SvName];
load(FileDirSpcClb) % load pL0 pDL pK
SpcEstData    = cell(4,1);
SpcEstData{1} = pL0; SpcEstData{2} = pDL;
SpcEstData{3} = Dat.SpcClb.SHGch; SpcEstData{4} = pK;
SpcEstData{5} = Dat.Img.OptEff; SpcEstData{6} = Dat.Img.SpecResp;
SpcEstData{7} = Dat.Img.PMTDiscRelGains;
 % create folder if necessary
[unused1,unused2,unused3] = mkdir(Dat.ImSegm.Dir);
clear unused1 unused2 unused3
```

```
 % === identify which frames & planes to process
XWinSz = Dat.ImSegm.XRange(2) - Dat.ImSegm.XRange(1) + 1;
YWinSz = Dat.ImSegm.YRange(2) - Dat.ImSegm.YRange(1) + 1;
FileNum = zeros(1,YWinSz*XWinSz); ctFN = 0;
for ctR = Dat.ImSegm.YRange(1):Dat.ImSegm.YRange(2)
    for ctC = Dat.ImSegm.XRange(1):Dat.ImSegm.XRange(2)
        ctF = ctC + (ctR-1)*Dat.Img.Xsteps;
        ctFN = ctFN +1;          FileNum(ctFN) = ctF-1;
    end
end
if Dat.DatCnv.ZBinSw;
    ZPlanes = 1;
else
    ZPlanes = Dat.ImSegm.ZRange(1):Dat.ImSegm.ZRange(2);
end
RowNum = Dat.Img.NumRow; ColNum = Dat.Img.NumCol;
 %% ====== main loop
FileNameConstR = [Dat.DatCnv.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_'];
FileNameConstS = [Dat.Nom.MainDir,Dat.Nom.DASH];
FileNameConstW = [Dat.ImSegm.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
                  Dat.ImSegm.SvName,'_'];
ChNum  = Dat.Img.ChNum;ChUsed = Dat.Img.ChUsed;NumChUsed = length(ChUsed);
 PadNum = Dat.Nom.PadNum;
SpcClbSHGch      = Dat.SpcClb.SHGch; ImDcmpWinSw      = Dat.ImDcmp.WinSw;
ImDcmpCompRead  = Dat.ImDcmp.CompRead; ImDcmpCompFName = Dat.ImDcmp.CompFName;
VoidDomSw = Dat.ImDcmp.VoidDomSw;DomNum         = Dat.ImDcmp.DomNum;
ImDcmpDomName = Dat.ImDcmp.DomName; ImDcmpDomComp = Dat.ImDcmp.DomComp;
ImDcmpDomCompNum = Dat.ImDcmp.DomCompNum;
DomCompMat  = Dat.ImDcmp.DomCompMat; DomCompLim  = Dat.ImDcmp.DomCompLim;
DomPrior    = Dat.ImSegm.DomPrior; DomTransMat = Dat.ImSegm.DomTransMat;
 if isempty(DomPrior);    DomPrior = ones(DomNum+VoidDomSw,1);end
DomPrior = DomPrior/sum(DomPrior);
VoidSpec = ones(NumChUsed,1)/NumChUsed; VoidRate = Dat.Img.Noise;
ImDcmpAlgor  = Dat.ImDcmp.Algor; ImSegmSaveSw = Dat.ImSegm.SaveSw;
ParProc      = Dat.Proc.ParProc; % NeighbSegmSw = false;
InclPiorSw   = true;
if ParProc %distributed computing enabled
    fprintf('\tdistributed computing enabled \n')
else
    fprintf('\tdistributed computing not enabled \n')
end
 %% ================== calculate normalized spectra matrix (row-dependent)
fprintf('\tpreliminary calculations | ')  SpectraRows    = cell(DomNum,1);
UssRow         = cell(DomNum,1);PseudoInvSSRow = cell(DomNum,1);
 for ctD = 1:DomNum
    DomComp = ImDcmpDomComp{ctD};     DomCompNum = ImDcmpDomCompNum(ctD);
 DomCompID    = find(DomCompMat(:,ctD));     DomCompRead  = ImDcmpCompRead(DomCompID);
    DomCompFName = ImDcmpCompFName{DomCompID};
    SpectraRowsDM = zeros(NumChUsed,DomCompNum,RowNum);
    UssRowDM = zeros(NumChUsed,DomCompNum,RowNum);
    PseudoInvSSRowDM = zeros(DomCompNum,NumChUsed,RowNum);
    if ParProc
        parfor ctR = 1:RowNum % iterate over rows
            SS = zeros(ChNum,DomCompNum);
            for ctE = 1:DomCompNum
                ElemID = DomComp{ctE};
                if DomCompRead(ctE)
TempSpec = LStackSpectrum(readconfocal16pics([FileNameConstS,DomCompFName{ctE}]));
                    if strcmpi(ElemID,'collagen')
                        TempSpec(1:SpcClbSHGch) = 0;
                    end
                    SS(:,ctE) = TempSpec/sum(TempSpec);
                else
                    SS(:,ctE) = SpectrumEstim_2_3(ElemID,SpcEstData,1,ctR);
                end
            end
             SS = SS(ChUsed,:);
            for ctE = 1:DomCompNum
                SS(:,ctE) = SS(:,ctE)/sum(SS(:,ctE));
            end
            SpectraRowsDM(:,:,ctR) = SS;
```

```
            [Uss,Sss,Vss]      = svd(SS);
            UssRowDM(:,:,ctR) = Uss(:,1:DomCompNum);
            PseudoInvSSRowDM(:,:,ctR) = ...
Vss*diag(1./diag(Sss(1:DomCompNum,1:DomCompNum)))*(Uss(:,1:DomCompNum))';
        end
    else
        for ctR = 1:RowNum % iterate over rows
            SS = zeros(ChNum,DomCompNum);
            for ctE = 1:DomCompNum
                ElemID = DomComp{ctE};
                if DomCompRead(ctE)
TempSpec = LStackSpectrum(readconfocal16pics([FileNameConstS,DomCompFName{ctE}]));
                    if strcmpi(ElemID,'collagen')
                        TempSpec(1:SpcClbSHGch) = 0;
                    end
                    SS(:,ctE) = TempSpec/sum(TempSpec);
                else
                    SS(:,ctE) = SpectrumEstim_2_3(ElemID,SpcEstData,1,ctR);
                end
            end
            % keep only the channels that contain data
            SS = SS(ChUsed,:);
            for ctE = 1:DomCompNum
                SS(:,ctE) = SS(:,ctE)/sum(SS(:,ctE));
            end
            SpectraRowsDM(:,:,ctR) = SS;
             [Uss,Sss,Vss]      = svd(SS);
            UssRowDM(:,:,ctR) = Uss(:,1:DomCompNum);
 PseudoInvSSRowDM(:,:,ctR) = ...
Vss*diag(1./diag(Sss(1:DomCompNum,1:DomCompNum)))*(Uss(:,1:DomCompNum))';
        end
    end
    SpectraRows{ctD}     = SpectraRowsDM;    UssRow{ctD}          = UssRowDM;
    PseudoInvSSRow{ctD} = PseudoInvSSRowDM;
end
clear SpectraRowsDM UssRowDM PseudoInvSSRowDM
 %% ==================== prepare data structures for EvidenceIntCalc_1_1
% initialize domain data structure
DomI = struct('SS',[],'SSPinv',[],'DomCompNum',[],'DomCompLim',[],...
    'polSam',[],'VLIM',[]);
for ctD = 1:DomNum
    % number of components per domain
    DomI(ctD).DomCompNum = ImDcmpDomCompNum(ctD);
    % component limits per domain
    DomI(ctD).DomCompLim = DomCompLim{ctD};
    % random sampling for each domain
    DomI(ctD).polSam = sobolset(ImDcmpDomCompNum(ctD),'Skip',1e3,'Leap',1e2);
    % boundary volume
    DomI(ctD).VLIM = prod(diff(DomCompLim{ctD}'));
end
% noise data structure
NoiseI = struct('VoidSpec',VoidSpec,'VoidSpecPinv',[],'VoidRate',VoidRate,...
    'VoidDomSw',VoidDomSw);
% noise Spectral matrix pseudoinverse
[Uss,Sss,Vss] = svd(VoidSpec);
NoiseI.VoidSpecPinv = Vss*1./Sss(1)*(Uss(:,1))';
% Monte Carlo Integration Parameters
MCIPar = struct('Ns',Ns,'NumSig',NumSig);
clear ctD
% fprintf('\tpreliminary calculations done \n')
%% ==================== MAIN LOOP for calculating the evidence of each frame
fprintf(' evidence calculations')
if ParProc %distributed computing enabled
    if length(FileNum)>1
        parfor ctFF = 1:length(FileNum) % iterate for all z-stacks
            ctF = FileNum(ctFF);
            if mod(ctF,10) == 0
                fprintf('.')
            end
            DomIL = DomI;
            for ctZZ = 1:length(ZPlanes);
```

```
                ctZ = ZPlanes(ctZZ);
     FileName = [FileNameConstR,Padding(ctF,PadNum),'_',Padding(ctZ,PadNum)];
                RawFrame = LoadMatFile(FileName);
                 RawFrame = RawFrame(:,:,ChUsed);
                 FrameMean = zeros(RowNum,ColNum,NumChUsed);
                for ctCH = 1:NumChUsed
FrameMean(:,:,ctCH) = round(imfilter(double(RawFrame(:,:,ctCH)),...
                        fspecial('average',3),'replicate'));
                end
                % ===  initialize output
                % component maps
                CompConMap = cell(VoidDomSw+DomNum,1);
                if VoidDomSw
                    CompConMap{1} = zeros(RowNum,ColNum,'single');
                end
                for ctD = 1:DomNum
                    CompConMap{VoidDomSw+ctD} = ...
                        zeros(RowNum,ColNum,ImDcmpDomCompNum(ctD),'single');
                end
                 % domain log probability
                DomLogProb = -inf(RowNum,ColNum,VoidDomSw+DomNum);
                 % loop over rows
                for ctR = 1:RowNum
                    % ===== ===== read & process raw data yy
                    if ImDcmpWinSw % use low-pass filtering
                        yy = squeeze(FrameMean(ctR,:,:))';
                    else
                        yy = squeeze(double(RawFrame(ctR,:,:)))';
                    end
                     % those pixels that have zero counts, use mean
                    yySUM = sum(yy); % calculate sum
                    yy(:,yySUM==0) = squeeze(FrameMean(ctR,yySUM==0,:))';
                     % ====== ===== update DomIL with row-specific data
                    for ctD = 1:DomNum
                        DomIL(ctD).SS = SpectraRows{ctD}(:,:,ctR);
                        DomIL(ctD).SSPinv = PseudoInvSSRow{ctD}(:,:,ctR);
                    end
                      % ====== ===== calculate ML & evidence integrals
[DomLogProbL,CompConMapL] = EvidenceIntCalc_1_1(yy,DomIL,NoiseI,MCIPar);
                     DomLogProb(ctR,:,:) = DomLogProbL;
                    for ctD = 1:(DomNum+VoidDomSw)
                        CompConMap{ctD}(ctR,:,:) = CompConMapL{ctD};
                    end
                 end % row
                %% add prior if necessary
                if InclPiorSw
DomLogProb = DomLogProb +
repmat(reshape(DomPrior,[1,1,DomNum+VoidDomSw]),[size(DomLogProb,1),size(DomLogProb,2)]);
                end
                %% check whether all domains have -inf
  PxAllInf = find(all(isinf(DomLogProb),3));         PxAllInf = PxAllInf(:);
                 if ~isempty(PxAllInf)
                    DomLogProb = reshape(DomLogProb,[RowNum*ColNum,VoidDomSw+DomNum]);
                     [rr,cc] = ind2sub([RowNum ColNum],PxAllInf);
     rN = repmat(rr,[1 4]) + repmat([-1 1 0 0],[length(PxAllInf),1]);% T B R L
     cN = repmat(cc,[1 4]) + repmat([0 0 1 -1],[length(PxAllInf),1]);% T B R L
                    Nghb2Use = (rN>=1) & (rN<=RowNum) & (cN>=1) & (cN<=ColNum);
                    for ctPx = 1:length(PxAllInf)
                        NghbPx = sub2ind([RowNum
ColNum],rN(ctPx,Nghb2Use(ctPx,:)),cN(ctPx,Nghb2Use(ctPx,:)));
                        NghbPx = setdiff(NghbPx,PxAllInf);
                        if ~isempty(NghbPx)
                            DomLogProb(PxAllInf(ctPx),:) = max(DomLogProb(NghbPx,:));
                        end
                        if all(isinf(DomLogProb(PxAllInf(ctPx),:)))
                            if VoidDomSw
                                DomLogProb(PxAllInf(ctPx),1) = 1;
                                DomLogProb(PxAllInf(ctPx),2:(1+DomNum)) = 0;
                            else
                                DomLogProb(PxAllInf(ctPx),:) = 0;
                            end
```

```
                    end
                end
                DomLogProb = reshape(DomLogProb,[RowNum,ColNum,VoidDomSw+DomNum]);
            end
            %% ===== normalize domain probability results
DomProbN = single(exp(DomLogProb - repmat(max(DomLogProb,[],3),[1,1,DomNum+VoidDomSw])));
   tmp = DomProbN(:,:,1);      tmp(logical(all(isinf(DomLogProb),3))) = 1;
   tmp(sum(DomProbN,3)==0) = 1;         DomProbN(:,:,1) = tmp;
   EvidenceDom = single(DomProbN./repmat(sum(DomProbN,3),[1,1,DomNum+VoidDomSw]));
            if ImSegmSaveSw
  SaveFileDir = [FileNameConstW,Padding(ctF,PadNum),'_',Padding(ctZ,PadNum)];
               SaveSegmData(CompConMap,EvidenceDom,SaveFileDir);
            end
         end % plane
      end % file (z-stack)
   else % when there is only one z-stack, parfor over z
      fprintf('.')
         parfor ctZZ = 1:length(ZPlanes);
            DomIL = DomI;
FileName = [FileNameConstR,Padding(FileNum(1),PadNum),'_',Padding(ZPlanes(ctZZ),PadNum)];
            RawFrame = LoadMatFile(FileName);
            RawFrame = RawFrame(:,:,ChUsed);
            FrameMean = zeros(RowNum,ColNum,NumChUsed);
            for ctCH = 1:NumChUsed
FrameMean(:,:,ctCH) = round(imfilter(double(RawFrame(:,:,ctCH)),...
                  fspecial('average',3),'replicate'));
            end
            CompConMap = cell(VoidDomSw+DomNum,1);
      if VoidDomSw;  CompConMap{1} = zeros(RowNum,ColNum,'single');          end
            for ctD = 1:DomNum
               CompConMap{VoidDomSw+ctD} = ...
                  zeros(RowNum,ColNum,ImDcmpDomCompNum(ctD),'single');
            end
             DomLogProb = -inf(RowNum,ColNum,VoidDomSw+DomNum);
             for ctR = 1:RowNum
               if ImDcmpWinSw % use low-pass filtering
                  yy = squeeze(FrameMean(ctR,:,:))';
               else
                  yy = squeeze(double(RawFrame(ctR,:,:)))';
               end
                yySUM = sum(yy); % calculate sum
               yy(:,yySUM==0) = squeeze(FrameMean(ctR,yySUM==0,:))';
                for ctD = 1:DomNum
                  DomIL(ctD).SS = SpectraRows{ctD}(:,:,ctR);
                  DomIL(ctD).SSPinv = PseudoInvSSRow{ctD}(:,:,ctR);
               end
    [DomLogProbL,CompConMapL] = EvidenceIntCalc_1_1(yy,DomIL,NoiseI,MCIPar);
                DomLogProb(ctR,:,:) = DomLogProbL;
               for ctD = 1:(DomNum+VoidDomSw)
                  CompConMap{ctD}(ctR,:,:) = CompConMapL{ctD};
               end
            end % row
             if InclPiorSw
DomLogProb = DomLogProb +
repmat(reshape(DomPrior,[1,1,DomNum+VoidDomSw]),[size(DomLogProb,1),size(DomLogProb,2)]);
            end
            PxAllInf = find(all(isinf(DomLogProb),3));
            PxAllInf = PxAllInf(:);
             if ~isempty(PxAllInf)
               % reshape the DomLogProb matrix
               DomLogProb = reshape(DomLogProb,[RowNum*ColNum,VoidDomSw+DomNum]);
               [rr,cc] = ind2sub([RowNum ColNum],PxAllInf);
rN = repmat(rr,[1 4]) + repmat([-1 1 0 0],[length(PxAllInf),1]);% T B R L
cN = repmat(cc,[1 4]) + repmat([0 0 1 -1],[length(PxAllInf),1]);% T B R L
               Nghb2Use = (rN>=1) & (rN<=RowNum) & (cN>=1) & (cN<=ColNum);
               for ctPx = 1:length(PxAllInf)
NghbPx = sub2ind([RowNum ColNum],rN(ctPx,Nghb2Use(ctPx,:)),cN(ctPx,Nghb2Use(ctPx,:)));
                  NghbPx = setdiff(NghbPx,PxAllInf);
                  if ~isempty(NghbPx)
                     DomLogProb(PxAllInf(ctPx),:) = max(DomLogProb(NghbPx,:));
                  end
```

```
                        if all(isinf(DomLogProb(PxAllInf(ctPx),:)))
                            if VoidDomSw
                                DomLogProb(PxAllInf(ctPx),1) = 1;
                                DomLogProb(PxAllInf(ctPx),2:(1+DomNum)) = 0;
                            else
                                DomLogProb(PxAllInf(ctPx),:) = 0;
                            end
                        end
                    end
                    DomLogProb = reshape(DomLogProb,[RowNum,ColNum,VoidDomSw+DomNum]);
                end
                %% ===== normalize domain probability results
                DomProbN = single(exp(DomLogProb -
repmat(max(DomLogProb,[],3),[1,1,DomNum+VoidDomSw])));
tmp = DomProbN(:,:,1);                tmp(logical(all(isinf(DomLogProb),3))) = 1;
tmp(sum(DomProbN,3)==0) = 1;                DomProbN(:,:,1) = tmp;
EvidenceDom = single(DomProbN./repmat(sum(DomProbN,3),[1,1,DomNum+VoidDomSw]));
                if ImSegmSaveSw
SaveFileDir = [FileNameConstW,Padding(FileNum(1),PadNum),'_',Padding(ZPlanes(ctZZ),PadNum)];
                    SaveSegmData(CompConMap,EvidenceDom,SaveFileDir);
                end

            end % plane

    end
else % distributed computing not enabled
    for ctFF = 1:length(FileNum) % iterate for all z-stacks
        ctF = FileNum(ctFF);
        if mod(ctF,10) == 0;            fprintf('.');        end
        DomIL = DomI;
        for ctZZ = 1:length(ZPlanes);
            ctZ = ZPlanes(ctZZ);
            load([FileNameConstR,Padding(ctF,PadNum),'_',Padding(ctZ,PadNum)])
            RawFrame = RawFrame(:,:,ChUsed);
            FrameMean = zeros(RowNum,ColNum,NumChUsed);
            for ctCH = 1:NumChUsed
                FrameMean(:,:,ctCH) = round(imfilter(double(RawFrame(:,:,ctCH)),...
                    fspecial('average',3),'replicate'));
            end
            CompConMap = cell(VoidDomSw+DomNum,1);
            if VoidDomSw;    CompConMap{1} = zeros(RowNum,ColNum,'single');     end
            for ctD = 1:DomNum
CompConMap{VoidDomSw+ctD} = zeros(RowNum,ColNum,ImDcmpDomCompNum(ctD),'single');
            end
            DomLogProb = -inf(RowNum,ColNum,VoidDomSw+DomNum);
            for ctR = 1:RowNum
                if ImDcmpWinSw % use low-pass filtering
                    yy = squeeze(FrameMean(ctR,:,:))';
                else
                    yy = squeeze(double(RawFrame(ctR,:,:)))';
                end
                yySUM = sum(yy);% calculate sum
                yy(:,yySUM==0) = squeeze(FrameMean(ctR,yySUM==0,:))';
                for ctD = 1:DomNum
                    DomIL(ctD).SS = SpectraRows{ctD}(:,:,ctR);
                    DomIL(ctD).SSPinv = PseudoInvSSRow{ctD}(:,:,ctR);
                end
 [DomLogProbL,CompConMapL] = EvidenceIntCalc_1_1(yy,DomIL,NoiseI,MCIPar);
                 DomLogProb(ctR,:,:) = DomLogProbL;
                for ctD = 1:(DomNum+VoidDomSw)
                    CompConMap{ctD}(ctR,:,:) = CompConMapL{ctD};
                end
    end % row
            if InclPiorSw
DomLogProb = DomLogProb + repmat(reshape(DomPrior,[1,1,DomNum+VoidDomSw]),...
                    [size(DomLogProb,1),size(DomLogProb,2)]);
            end
            PxAllInf = find(all(isinf(DomLogProb),3));  PxAllInf = PxAllInf(:);
            if ~isempty(PxAllInf)
                DomLogProb = reshape(DomLogProb,[RowNum*ColNum,VoidDomSw+DomNum]);
                [rr,cc] = ind2sub([RowNum ColNum],PxAllInf);
```

```
rN = repmat(rr,[1 4]) + repmat([-1 1 0 0],[length(PxAllInf),1]);% T B R L
cN = repmat(cc,[1 4]) + repmat([0 0 1 -1],[length(PxAllInf),1]);% T B R L
                Nghb2Use = (rN>=1) & (rN<=RowNum) & (cN>=1) & (cN<=ColNum);
                for ctPx = 1:length(PxAllInf)
NghbPx = sub2ind([RowNum ColNum],rN(ctPx,Nghb2Use(ctPx,:)),cN(ctPx,Nghb2Use(ctPx,:)));
                    NghbPx = setdiff(NghbPx,PxAllInf);
                    if ~isempty(NghbPx)
                        DomLogProb(PxAllInf(ctPx),:) = max(DomLogProb(NghbPx,:));
                    end
                    if all(isinf(DomLogProb(PxAllInf(ctPx),:)))
                        if VoidDomSw
                            DomLogProb(PxAllInf(ctPx),1) = 1;
                            DomLogProb(PxAllInf(ctPx),2:(1+DomNum)) = 0;
                        else
                            DomLogProb(PxAllInf(ctPx),:) = 0;
                        end
                    end
                end
                DomLogProb = reshape(DomLogProb,[RowNum,ColNum,VoidDomSw+DomNum]);
            end
DomProbN = single(exp(DomLogProb - repmat(max(DomLogProb,[],3),[1,1,DomNum+VoidDomSw])));
            tmp = DomProbN(:,:,1);  tmp(logical(all(isinf(DomLogProb),3))) = 1;
            tmp(sum(DomProbN,3)==0) = 1;          DomProbN(:,:,1) = tmp;
 EvidenceDom = single(DomProbN./repmat(sum(DomProbN,3),[1,1,DomNum+VoidDomSw]));
            if ImSegmSaveSw
SaveFileDir = [FileNameConstW,Padding(ctF,PadNum),'_',Padding(ctZ,PadNum)];
               save(SaveFileDir,'CompConMap','EvidenceDom')
            end
        end % plane
    end % z-stack
end
fprintf('\n')
fprintf('\t... done\n\n')
return


function Err = funML(x,Param)
 % yy = Param{1};
% S  = Param{2};
% DomCompNum = Param{3};
 Err = -ones(Param{3},1);
 x = x(:);
for cti = 1:Param{3}
    for ctCH = 1:length(Param{1})
        Err(cti) = Err(cti) + Param{1}(ctCH)*Param{2}(ctCH,cti)/(Param{2}(ctCH,:)*x);
    end
end
 function RawFrame = LoadMatFile(FileName)
load(FileName)
return

function y = SaveSegmData(CompConMap,EvidenceDom,SaveFileDir)
save(SaveFileDir,'CompConMap','EvidenceDom')
return
```

AssembleDomEvid

```
function ZPlanes = AssembleDomEvid_1_8(Dat)
% assemble decomposed images based on the optimum image registration
% Dimitrios Tzeranis, Aug 2012 - June 2013
fprintf('=== Assemblying Evidence Frames ===\n')
 DEBUGMODEsw = false; % if true save 'DomProbMap','DomProbMapPROC'
 [unused1,unused2,unused3] = mkdir(Dat.ImAsbl.Dir);
ZPlanes = Dat.ImSegm.ZRangePrint(1):Dat.ImSegm.ZRangePrint(2);
if Dat.ImDcmp.VoidDomSw
    NumEle = [1 Dat.ImDcmp.DomCompNum];
else
    NumEle = Dat.ImDcmp.DomCompNum;
end
DomNum = Dat.ImDcmp.DomNum;VoidDomSw = Dat.ImDcmp.VoidDomSw;
for ctZ = ZPlanes
```

```
    fprintf('\tPlane %s: Assemble Evidence |',num2str(ctZ))
    if ctZ % non-zero z
        if (Dat.Img.Xsteps==1) && (Dat.Img.Ysteps==1)     % case of a z-stack
            load([Dat.ImSegm.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
                Dat.ImSegm.SvName,'_',Padding(0,Dat.Nom.PadNum),'_',...
                Padding(ctZ,Dat.Nom.PadNum)]); % reads CompConMap and EvidenceDom'
            DomProbMap = EvidenceDom;              AsblCompConMap = CompConMap;
            clear CompConMap EvidenceDom
        else
            XRange = Dat.ImReg.XRange;              YRange = Dat.ImReg.YRange;
            XWinSz = XRange(2) - XRange(1) + 1; YWinSz = YRange(2) - YRange(1) + 1;
            FileNum = zeros(1,YWinSz*XWinSz);              ctFN = 0;
            for ctR = YRange(1):YRange(2)
                for ctC = XRange(1):XRange(2)
                    ctF = ctC + (ctR-1)*Dat.Img.Xsteps;    ctFN = ctFN +1;
                    FileNum(ctFN) = ctF;
                end
            end
            load([Dat.ImReg.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_ImageRegOverview']);
            posiO = RegImagDat.posiO;              dispO = RegImagDat.dispO;
            insertO = RegImagDat.insertO;              rangeO = RegImagDat.rangeO;
            clear RegImagDat
            maxR = max(rangeO(1:2:end));              minR = min(insertO(1:2:end));
            maxC = max(rangeO(2:2:end));              minC = min(insertO(2:2:end));
            if minR~=1
                TotalPixX = maxR - minR + 1;
                rangeO(1:2:end) = rangeO(1:2:end) - minR + 1;
                insertO(1:2:end) = insertO(1:2:end) - minR + 1;
            else
                TotalPixX = maxR;
            end
            TotalPixY = maxC;
            if minC<=0
                TotalPixY = TotalPixY - minC + 1;
                rangeO(2:2:end) = rangeO(2:2:end) - minC + 1;
                insertO(2:2:end) = insertO(2:2:end) - minC + 1;
            end
            DomProbMap = zeros(TotalPixY,TotalPixX,VoidDomSw+DomNum);
            AsblCompConMap = cell(VoidDomSw+DomNum,1);
            for ctD = 1:(VoidDomSw+DomNum)
                AsblCompConMap{ctD} = zeros(TotalPixY,TotalPixX,NumEle(ctD),'single');
            end
            for ctF = 1:XWinSz*YWinSz
                ctR = floor((ctF-1)/XWinSz) + 1;
                ctC = ctF - (ctR-1)*XWinSz;    addF = [2*ctF-1,2*ctF];
                dO = dispO(addF);
load([Dat.ImSegm.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
Dat.ImSegm.SvName,'_',Padding(FileNum(ctF)-1,Dat.Nom.PadNum),'_',...
Padding(ctZ,Dat.Nom.PadNum)]); % reads 'CompConMap','EvidenceDom'
                DecImF = EvidenceDom;                ComImF = CompConMap;
                 if Dat.ImReg.RemLSw
                    if strcmpi(Dat.Img.Instr,'MMM')
                        DecImF = DecImF(8:end,:,:);
                        for ctD = 1:(VoidDomSw+DomNum)
                            ComImF{ctD} = ComImF{ctD}(8:end,:,:);
                        end
                     else
                        % original microscope: remove left pixels
                        DecImF = DecImF(:,Dat.ImReg.RemLPixNum + 1:end,:);
                        for ctD = 1:(VoidDomSw+DomNum)
         ComImF{ctD} = ComImF{ctD}(:,Dat.ImReg.RemLPixNum + 1:end,:);
                        end
                    end
                end
                dimF = size(DecImF);
                clear CompConMap EvidenceDom
                if (ctR == 1) && (ctC == 1)
                elseif (ctR == 1) && (ctC > 1)
                    % === === === L-R
                    ctL = ctF - 1;
load([Dat.ImSegm.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
```

```
Dat.ImSegm.SvName,'_',Padding(FileNum(ctL)-1,Dat.Nom.PadNum),'_',...
Padding(ctZ,Dat.Nom.PadNum)]); % reads 'CompConMap','EvidenceDom'
                    DecImL = EvidenceDom;      ComImL = CompConMap;
                    if  Dat.ImReg.RemLSw;
                        DecImL = DecImL(:,Dat.ImReg.RemLPixNum + 1:end,:);
                        for ctD = 1:(VoidDomSw+DomNum)
ComImL{ctD} = ComImL{ctD}(:,Dat.ImReg.RemLPixNum + 1:end,:);
                        end
                    end
                    dimL = size(DecImL);
                    [cs1,cs2] = CalculateWindowCS2(dimL,dimF,1,dO);
MaskLR = (linspace(0,1,cs1(4)-cs1(3)+1)'*ones(1,cs1(2)-cs1(1)+1))';
                    for ctE = 1:(VoidDomSw+DomNum)
      DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE) = ...
        round((1-MaskLR).*double(DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE)) + ...
              MaskLR.*double(DecImL(cs1(1):cs1(2),cs1(3):cs1(4),ctE)));
                    end
                    for ctD = 1:(VoidDomSw+DomNum)
                        for ctE = 1:NumEle(ctD)
                            ComImF{ctD}(cs2(1):cs2(2),cs2(3):cs2(4),ctE) = ...
round((1-MaskLR).*single(ComImF{ctD}(cs2(1):cs2(2),cs2(3):cs2(4),ctE)) + ...
     MaskLR.*single(ComImL{ctD}(cs1(1):cs1(2),cs1(3):cs1(4),ctE)));
                        end
                    end
                    clear CompConMap EvidenceDom
                elseif (ctR > 1) && (ctC == 1)
                    % === === === T-B
                    ctT = ctF - XWinSz;
load([Dat.ImSegm.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
Dat.ImSegm.SvName,'_',Padding(FileNum(ctT)-1,Dat.Nom.PadNum),'_',...
Padding(ctZ,Dat.Nom.PadNum)]); % reads 'CompConMap','EvidenceDom'
                    DecImT = EvidenceDom;                    ComImT = CompConMap;
                    if Dat.ImReg.RemLSw
                        if strcmpi(Dat.Img.Instr,'MMM')
                            DecImT = DecImT(8:end,:,:);
                            for ctD = 1:(VoidDomSw+DomNum)
                                ComImT{ctD} = ComImT{ctD}(8:end,:,:);
                            end
                        else
                            % original microscope: remove left pixels
                            DecImT = DecImT(:,Dat.ImReg.RemLPixNum + 1:end,:);
                            for ctD = 1:(VoidDomSw+DomNum)
       ComImT{ctD} = ComImT{ctD}(:,Dat.ImReg.RemLPixNum + 1:end,:);
                            end
                        end
                    end

                    dimT = size(DecImT);
                    [cs1,cs2] = CalculateWindowCS2(dimT,dimF,2,dO);
                    MaskTB = (ones(cs1(4)-cs1(3)+1,1)*linspace(0,1,cs1(2)-cs1(1)+1))';
                    for ctE = 1:(VoidDomSw+DomNum)
DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE) = ...
round((MaskTB).*double(DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE)) + ...
 (1-MaskTB).*double(DecImT(cs1(1):cs1(2),cs1(3):cs1(4),ctE)));
                    end
                    for ctD = 1:(VoidDomSw+DomNum)
                        for ctE = 1:NumEle(ctD)
                            ComImF{ctD}(cs2(1):cs2(2),cs2(3):cs2(4),ctE) = ...

round((MaskTB).*single(ComImF{ctD}(cs2(1):cs2(2),cs2(3):cs2(4),ctE)) + ...
 (1-MaskTB).*single(ComImT{ctD}(cs1(1):cs1(2),cs1(3):cs1(4),ctE)));
                        end
                    end
                    clear CompConMap EvidenceDom
                else
                    % === === ===L-R
                    ctL = ctF - 1;
load([Dat.ImSegm.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
Dat.ImSegm.SvName,'_',Padding(FileNum(ctL)-1,Dat.Nom.PadNum),'_',...
Padding(ctZ,Dat.Nom.PadNum)]);
                    DecImL = EvidenceDom;          ComImL = CompConMap;
```

```
                    if Dat.ImReg.RemLSw
                        if strcmpi(Dat.Img.Instr,'MMM')
                            DecImL = DecImL(8:end,:,:);
                            for ctD = 1:(VoidDomSw+DomNum)
                                ComImL{ctD} = ComImL{ctD}(8:end,:,:);
                            end
                        else
                            DecImL = DecImL(:,Dat.ImReg.RemLPixNum + 1:end,:);
                            for ctD = 1:(VoidDomSw+DomNum)
          ComImL{ctD} = ComImL{ctD}(:,Dat.ImReg.RemLPixNum + 1:end,:);
                            end
                        end
                    end
                     dimL = size(DecImL);
                     [cs1,cs2] = CalculateWindowCS2(dimL,dimF,1,dO);
        MaskLR = (linspace(0,1,cs1(4)-cs1(3)+1)'*ones(1,cs1(2)-cs1(1)+1))';
                    for ctE = 1:(VoidDomSw+DomNum)
      DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE) = ...
    round((1-MaskLR).*double(DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE)) + ...
                            MaskLR.*double(DecImL(cs1(1):cs1(2),cs1(3):cs1(4),ctE)));
                    end
                    for ctD = 1:(VoidDomSw+DomNum)
                        for ctE = 1:NumEle(ctD)
   ComImF{ctD}(cs2(1):cs2(2),cs2(3):cs2(4),ctE) = ...
       round((1-MaskLR).*single(ComImF{ctD}(cs2(1):cs2(2),cs2(3):cs2(4),ctE)) + ...
              MaskLR.*single(ComImL{ctD}(cs1(1):cs1(2),cs1(3):cs1(4),ctE)));
                        end
                    end
                    clear CompConMap EvidenceDom
                    % === === === T-B
                    ctT = ctF - XWinSz;
                    addT = [2*ctT-1,2*ctT];
                    dO =  posiO(addF) - posiO(addT);
load([Dat.ImSegm.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
Dat.ImSegm.SvName,'_',Padding(FileNum(ctT)-1,Dat.Nom.PadNum),'_',...
Padding(ctZ,Dat.Nom.PadNum)]); % reads 'CompConMap','EvidenceDom'
                    DecImT = EvidenceDom;      ComImT = CompConMap;
                    if Dat.ImReg.RemLSw
                        if strcmpi(Dat.Img.Instr,'MMM')
                            DecImT = DecImT(8:end,:,:);
                            for ctD = 1:(VoidDomSw+DomNum)
                                ComImT{ctD} = ComImT{ctD}(8:end,:,:);
                            end
                        else
                            DecImT = DecImT(:,Dat.ImReg.RemLPixNum + 1:end,:);
                            for ctD = 1:(VoidDomSw+DomNum)
                 ComImT{ctD} = ComImT{ctD}(:,Dat.ImReg.RemLPixNum + 1:end,:);
                            end
                        end
                    end

                    dimT = size(DecImT);
                    [cs1,cs2] = CalculateWindowCS2(dimT,dimF,2,dO);
                    MaskTB = (ones(cs1(4)-cs1(3)+1,1)*linspace(0,1,cs1(2)-cs1(1)+1))';
                    for ctE = 1:(VoidDomSw+DomNum)
          DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE) = ...
round((MaskTB).*double(DecImF(cs2(1):cs2(2),cs2(3):cs2(4),ctE)) + ...
  (1-MaskTB).*double(DecImT(cs1(1):cs1(2),cs1(3):cs1(4),ctE)));
                    end
                    for ctD = 1:(VoidDomSw+DomNum)
                        for ctE = 1:NumEle(ctD)
             ComImF{ctD}(cs2(1):cs2(2),cs2(3):cs2(4),ctE) = ...
round((MaskTB).*single(ComImF{ctD}(cs2(1):cs2(2),cs2(3):cs2(4),ctE)) + ...
      (1-MaskTB).*single(ComImT{ctD}(cs1(1):cs1(2),cs1(3):cs1(4),ctE)));
                        end
                    end
                    clear CompConMap EvidenceDom
                end
   DomProbMap(insertO(addF(2)):rangeO(addF(2)),...
                    insertO(addF(1)):rangeO(addF(1)),:) = DecImF;
                for ctD = 1:(VoidDomSw+DomNum)
```

```
                    AsblCompConMap{ctD}(insertO(addF(2)):rangeO(addF(2)),...
                    insertO(addF(1)):rangeO(addF(1)),:) = ComImF{ctD};
                end
            end
             DomProbMap = single(DomProbMap);
            for ctD = 1:(VoidDomSw+DomNum)
                AsblCompConMap{ctD} = single(AsblCompConMap{ctD});
            end
        end % is there stage raster scanning
    else % zero z
        load([Dat.GMaxIPr.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',Dat.GMaxIPr.SvName])
        FrmCrds = MaxICrds(:,:,1);  RowCrds = MaxICrds(:,:,2);
         ColCrds = MaxICrds(:,:,3);
        clear MaxIntImage MaxICrds
load([Dat.ImReg.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',Dat.ImReg.SvName,'_ImageExt'])
        TotalPixY = size(MaxIData,1);
        TotalPixX = size(MaxIData,2);
        DomProbMap = zeros(TotalPixY*TotalPixX,VoidDomSw+DomNum);
        AsblCompConMap = cell(VoidDomSw+DomNum,1);
        for ctD = 1:(VoidDomSw+DomNum)
            AsblCompConMap{ctD} = zeros(TotalPixY*TotalPixX,NumEle(ctD),'single');
        end
        for ctF = unique(FrmCrds)'
            PxF = find((FrmCrds==ctF) & ImageExt);
            for ctZZ = setdiff(unique(MaxIData(PxF))',0)
                load([Dat.ImSegm.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
                    Dat.ImSegm.SvName,'_',Padding(ctF,Dat.Nom.PadNum),'_',...
                    Padding(ctZZ,Dat.Nom.PadNum)]); % reads 'CompConMap','EvidenceDom'
                EvidenceDom = reshape(EvidenceDom,[256*256,size(EvidenceDom,3)]);
                for ctD = 1:(VoidDomSw+DomNum)
                    CompConMap{ctD} = reshape(CompConMap{ctD},[256*256,NumEle(ctD)]);
                end
                PxFZ = find((FrmCrds==ctF) & MaxIData==ctZZ & ImageExt);
               PxFZl = sub2ind([256,256],double(RowCrds(PxFZ)),double(ColCrds(PxFZ)));
                DomProbMap(PxFZ,:) = EvidenceDom(PxFZl,:);
                for ctD = 1:(VoidDomSw+DomNum)
                    AsblCompConMap{ctD}(PxFZ,:) = CompConMap{ctD}(PxFZl,:);
                end
            end
        end
  DomProbMap = single(reshape(DomProbMap,[TotalPixY,TotalPixX,VoidDomSw+DomNum]));
        for ctD = 1:(VoidDomSw+DomNum)
  AsblCompConMap{ctD} = single(reshape(AsblCompConMap{ctD},[TotalPixY,TotalPixX,NumEle(ctD)]));
        end
    end
    %% Morphological Processing
    fprintf(' Morphological Processing |')
    [sx,sy,sz] = size(DomProbMap); % (sz = VoidDomSw+DomNum)
    DomProbMap = reshape(DomProbMap,[sx*sy,sz]);
    DomProbMapsum = sum(DomProbMap,2);     Indx = find(DomProbMapsum>1);
    for ct = Indx'
        DomProbMap(ct,:) = DomProbMap(ct,:)/DomProbMapsum(ct);
    end
    DomProbMap = reshape(DomProbMap,[sx,sy,sz]);
    % === morphological processing
    % not applied to void domain
    se1 = strel('disk',1);
    se2 = strel('disk',1);%2
    DomProbMapPROC = DomProbMap;
    for ct = (1+VoidDomSw):(DomNum+VoidDomSw)
 DomProbMapPROC(:,:,ct) = medfilt2(imclose(imreconstruct(imopen(DomProbMap(:,:,ct),se1),...
            imclose(DomProbMap(:,:,ct),se1)),se2),[3 3]);
    end
    DomProbMapPROC = reshape(DomProbMapPROC,[sx*sy,sz]);
    DomProbMapPROCsum  = sum(DomProbMapPROC,2);
    Indx = find(DomProbMapPROCsum>1);
    for ct = Indx'
        DomProbMapPROC(ct,:) = DomProbMapPROC(ct,:)./DomProbMapPROCsum(ct);
    end
    DomProbMapPROC = reshape(DomProbMapPROC,[sx,sy,sz]);
    fprintf(' Save')
```

```
    SaveFileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,...
        '_DomEvidAsbl_',Padding(ctZ,Dat.Nom.PadNum)];
    save(SaveFileDir,'AsblCompConMap','DomProbMapPROC')
    if DEBUGMODEsw
        SaveFileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,...
            '_DomProbMapData_',Padding(ctZ,Dat.Nom.PadNum)];
        save(SaveFileDir,'DomProbMap')
    end
    if Dat.ImSegm.SvImgSw
        fprintf(' | Print')
        SvImgCh = Dat.ImSegm.SvImgCh;          SvImgCh(SvImgCh>DomNum+VoidDomSw) = nan;
        SvImgCh2Prnt = SvImgCh;         SvImgCh2Del = find(isnan(SvImgCh));
        ChNames = '';
        for ctCH = 1:3
            if ~isnan(SvImgCh(ctCH))
                ChNames = [ChNames,num2str(SvImgCh(ctCH))];
            else
                ChNames = [ChNames,'-'];
                SvImgCh2Prnt(ctCH) = 1;
            end
        end
        SaveFileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
Dat.ImAsbl.SvName,'_z',Padding(ctZ,Dat.Nom.PadNum),'_DomProbMaps',ChNames];
        SaveFileProcDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
Dat.ImAsbl.SvName,'_z',Padding(ctZ,Dat.Nom.PadNum),'_DomProbMapsPROC',ChNames];
        if (Dat.ImAsbl.ImPrintScale==100)
            tmp1 = DomProbMap(:,:,SvImgCh2Prnt);            tmp1(:,:,SvImgCh2Del) = 0;
            imwrite(tmp1,[SaveFileDir,'.png'],'png')
               tmp1 = DomProbMapPROC(:,:,SvImgCh2Prnt);   tmp1(:,:,SvImgCh2Del) = 0;
            imwrite(tmp1,[SaveFileProcDir,'.png'],'png')
        else
            tmp1 = DomProbMap(:,:,SvImgCh2Prnt);
            tmp1(:,:,SvImgCh2Del) = 0;
            imwrite(imresize(tmp1,Dat.ImAsbl.ImPrintScale/100),...
                [SaveFileDir,'_sc',num2str(Dat.ImAsbl.ImPrintScale),'.png'],'png')
            tmp1 = DomProbMapPROC(:,:,SvImgCh2Prnt);
            tmp1(:,:,SvImgCh2Del) = 0;
            imwrite(imresize(tmp1,Dat.ImAsbl.ImPrintScale/100),...
                [SaveFileProcDir,'_sc',num2str(Dat.ImAsbl.ImPrintScale),'.png'],'png')
        end
    end
    fprintf('\n')
end % z plane
fprintf('\t\t..... done\n\n')
 return
```

ImageSegm (Local)

```
function ZPlanes = ImSegmLoc_1_3(Dat)
% local (pixel-wide) Bayesian image segmentation
% Dimitrios Tzeranis, November 2012 - June 2013
fprintf('=== Local Image Segmentation (Bayesian) ===\n')
%% initialization
[unused1,unused2,unused3] = mkdir(Dat.ImAsbl.Dir);
 ZPlanes = Dat.ImSegm.ZRangePrint(1):Dat.ImSegm.ZRangePrint(2);
 NumEle = [1 Dat.ImDcmp.DomCompNum];
 % load ImageExt
LoadFileDir = [Dat.ImReg.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',Dat.ImReg.SvName,'_ImageExt'];
load(LoadFileDir) % load 'ImageExt'
% IndExt = find(~ImageExt); % find which nodes are not really image
for ctZ = ZPlanes %%%
    fprintf('\tPlane %s: ',num2str(ctZ));    fprintf('Classify |')
    % ===== pre-process data
    % load data
OpenFileDir =
[Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_DomEvidAsbl_',Padding(ctZ,Dat.Nom.PadNum)];
    load(OpenFileDir) % loads 'DomProbMapPROC' 'AsblCompConMap'
    clear AsblCompConMap
    % normalize pixels of DomProbMapPROC whose sum >0 and <1
    [sx,sy,sz] = size(DomProbMapPROC);
    DomProbMapPROC = reshape(DomProbMapPROC,[sx*sy,sz]);
```

```
    DomProbMapPROCsum = sum(DomProbMapPROC,2);
    IndxPROCsum  = find(DomProbMapPROCsum>0 & DomProbMapPROCsum<1);
    IndxPROCzero = find(DomProbMapPROCsum==0);
    DomProbMapPROC2Norm = DomProbMapPROC(IndxPROCsum,:);
    DomProbMapPROCsum   = DomProbMapPROCsum(IndxPROCsum);
    parfor ct = 1:length(IndxPROCsum)
        DomProbMapPROC2Norm(ct,:) = DomProbMapPROC2Norm(ct,:)/DomProbMapPROCsum(ct);
    end
    DomProbMapPROC(IndxPROCsum,:) = DomProbMapPROC2Norm;
    clear DomProbMapPROCsum IndxPROCsum DomProbMapPROC2Norm DomProbMapPROCsum
    % take care of pixels whose sum==0
    DomProbMapPROCZero = zeros(length(IndxPROCzero),sz,'single');
    parfor ct = 1:length(IndxPROCzero)
        [Row,Col] = ind2sub([sx sy],IndxPROCzero(ct));
        nghb = IndxPROCzero(ct) + [-sx -1 1 sx]; % L T B R
        nghb([Col==1 Row==1 Row==sx Col==sy]) = [];
        temp = DomProbMapPROC(nghb,:);
        if any(sum(temp,2))
            temp = sum(temp);
            DomProbMapPROCZero(ct,:) = temp/sum(temp);
        end
    end
    DomProbMapPROC(IndxPROCzero,:) = DomProbMapPROCZero;
    DomProbMapPROC = reshape(DomProbMapPROC,[sx,sy,sz]);
    % ===== initial segmentation
    ImSegmLoc    = zeros(sx,sy,'uint8');
    % define segmentation law
    ImSegmLocLaw = 'max';
    switch ImSegmLocLaw
        case 'max'
            [x1,ImSegmLoc] = max(DomProbMapPROC,[],3);
            clear x1
        case 'thres'
            Hierarchy = [4 2 3 1];            SegmThres = [0.1 0.1 0.1 0.1];
            DomProbMapPROCtemp = DomProbMapPROC;
            for ctD = 1:sz
                MaskD = DomProbMapPROCtemp(:,:,Hierarchy(ctD))>=SegmThres(ctD);
                ImSegmLoc(MaskD) = Hierarchy(ctD);
                for ctDD = setdiff(1:sz,Hierarchy(ctD))
                    temp = DomProbMapPROCtemp(:,:,ctDD);
                    temp(MaskD) = 0;    DomProbMapPROCtemp(:,:,ctDD) = temp;
                end
            end
            clear DomProbMapPROCtemp
        otherwise
            disp('3')
    end
    %% === save data
    fprintf(' Save')
    SaveFileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,...
        '_ImSegmLoc_',Padding(ctZ,Dat.Nom.PadNum)];
    save(SaveFileDir,'ImSegmLoc')
    %% === print images
    if Dat.ImSegm.SvImgSw
        fprintf(' | Print')
        if isempty(Dat.ImDcmp.VoidDomSw);    Dat.ImDcmp.VoidDomSw = 1;      end
        % generate HSV image to print
        ColorBar =
floor(linspace(1,Dat.ImDcmp.DomNum+Dat.ImDcmp.VoidDomSw+1,size(ImSegmLoc,1)+1));
        ColorBar(end) = [];
        ImSegmLoc = [ImSegmLoc,nan(size(ImSegmLoc,1),20),repmat(ColorBar',[1,30])];
        szI1 = size(ImSegmLoc,1);
        if Dat.ImDcmp.VoidDomSw
           Im2Print = hsv2rgb((double(ImSegmLoc)-2)/(Dat.ImDcmp.DomNum),...
               ones(size(ImSegmLoc)),double([ImageExt,ones(szI1,50)]).*(ImSegmLoc>1));
        else
           Im2Print = hsv2rgb((double(ImSegmLoc))/(Dat.ImDcmp.DomNum+1),...
               ones(size(ImSegmLoc)),double([ImageExt,ones(szI1,50)]).*(ImSegmLoc>1));
        end
        % domain names
        if Dat.ImDcmp.VoidDomSw
```

```
            ChNames = 'void_';
        else
            ChNames = '';
        end
        for ctD = 1:Dat.ImDcmp.DomNum
            ChNames = [ChNames,Dat.ImDcmp.DomName{ctD},'_'];
        end
        % file name
        SaveFileSegmDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
            Dat.ImAsbl.SvName,'_z',Padding(ctZ,Dat.Nom.PadNum),'_ImSegmLoc_',ChNames(1:end-1)];
        % save
        if (Dat.ImAsbl.ImPrintScale==100)
            imwrite(Im2Print,[SaveFileSegmDir,'.png'],'png')
        else
            imwrite(imresize(Im2Print,Dat.ImAsbl.ImPrintScale/100),...
                [SaveFileSegmDir,'_sc',num2str(Dat.ImAsbl.ImPrintScale),'.png'],'png')
        end
    end
    fprintf('\n')
end % z plane
fprintf('\t\t..... done\n\n')
return
```

ImageSegm (Global)

```
function ZPlanes = ImSegmGlob_1_4(Dat)
% global image segmentation by MRF
% Dimitrios Tzeranis, November 2012 - June 2013
fprintf('=== Image Segmentation ===\n')
%% load MRF model parameters
if isempty(Dat.ImDcmp.VoidDomSw);    Dat.ImDcmp.VoidDomSw = 1;end
psi = Dat.ImSegm.DomTransMat;psi(psi==0) = eps; % PSI matrix
if isempty(Dat.ImSegm.ImDlSz);  Dat.ImSegm.ImDlSz = 2;end
MRFDilSz = Dat.ImSegm.ImDlSz; % dilation size
% Boundary Map
if isempty(Dat.ImSegm.EnableBndr)
    Dat.ImSegm.EnableBndr = true(Dat.ImDcmp.VoidDomSw + Dat.ImDcmp.DomNum);
end
BoundaryMap = Dat.ImSegm.EnableBndr;
%% initialization
[unused1,unused2,unused3] = mkdir(Dat.ImAsbl.Dir);
ZPlanes = Dat.ImSegm.ZRangePrint(1):Dat.ImSegm.ZRangePrint(2);
NumEle = [1 Dat.ImDcmp.DomCompNum];
LoadFileDir = [Dat.ImReg.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',Dat.ImReg.SvName,'_ImageExt'];
load(LoadFileDir)
for ctZ = ZPlanes
    fprintf('\tPlane %s: ',num2str(ctZ))
    OpenFileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,...
        '_DomEvidAsbl_',Padding(ctZ,Dat.Nom.PadNum)];
    load(OpenFileDir) % loads 'DomProbMapPROC' 'AsblCompConMap'
    clear AsblCompConMap
    % load local segmentation
    OpenFileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,...
        '_ImSegmLoc_',Padding(ctZ,Dat.Nom.PadNum)];
    load(OpenFileDir) % loads 'ImSegmLoc'
    ImSegm = ImSegmLoc;
    clear ImSegmLoc
    %% === data tuning
    fprintf('. ')
    [sx,sy,sz] = size(DomProbMapPROC);
    % ===== normalize the pixels of DomProbMapPROC whose sum >0 and <1
    DomProbMapPROC = reshape(DomProbMapPROC,[sx*sy,sz]);
    DomProbMapPROCsum = sum(DomProbMapPROC,2);
    IndxPROCsum = find(DomProbMapPROCsum>0 & DomProbMapPROCsum<1);
    IndxPROCzero= find(DomProbMapPROCsum==0);
    DomProbMapPROC2Norm = DomProbMapPROC(IndxPROCsum,:);
    DomProbMapPROCsum   = DomProbMapPROCsum(IndxPROCsum);
    parfor ct = 1:length(IndxPROCsum)
        DomProbMapPROC2Norm(ct,:) = ...
            DomProbMapPROC2Norm(ct,:)/DomProbMapPROCsum(ct);
    end
```

```
    DomProbMapPROC(IndxPROCsum,:) = DomProbMapPROC2Norm;
    clear DomProbMapPROCsum IndxPROCsum DomProbMapPROC2Norm DomProbMapPROCsum
    % ===== take care of pixels whose sum==0
    fprintf('. ')
    DomProbMapPROCZero = zeros(length(IndxPROCzero),sz,'single');
    parfor ct = 1:length(IndxPROCzero)
        [Row,Col] = ind2sub([sx sy],IndxPROCzero(ct));
        nghb = IndxPROCzero(ct) + [-sx -1 1 sx]; % L T B R
        nghb([Col==1 Row==1 Row==sx Col==sy]) = [];
        temp = DomProbMapPROC(nghb,:);
        if any(sum(temp,2))
            temp = sum(temp);
            DomProbMapPROCZero(ct,:) = temp/sum(temp);
        end
    end
    DomProbMapPROC(IndxPROCzero,:) = DomProbMapPROCZero;
    DomProbMapPROC = reshape(DomProbMapPROC,[sx,sy,sz]);
    %% ===== identify boundaries to refine MRF
    fprintf('. ')
    sel = strel('disk',1);
    MaskDi = false(sx,sy,sz);
    for ctD = 1:sz
        if BoundaryMap(ctD,ctD)
            MaskDi(:,:,ctD) = imdilate(ImSegm==ctD,sel);
        end
    end
    MaskDC = false(sx,sy);
    for ctD1 = 1:sz
        BoundaryMapt = BoundaryMap(ctD1,:);
        BoundaryMapt(1:ctD1) = false;
        for ctD2 = find(BoundaryMapt)
            MaskDC = MaskDC | (MaskDi(:,:,ctD1)&MaskDi(:,:,ctD2));
        end
    end
    MaskDC = (imdilate(MaskDC & ImageExt,strel('disk',MRFDilSz)) | (ImSegm==0)) & ImageExt ;
    clear MaskDi
    %% === identify areas for MRF
    Areas4MRF = bwconncomp(MaskDC,4);
    if Areas4MRF.NumObjects
        fprintf('%s regions identified.. ',num2str(Areas4MRF.NumObjects))
        % organize trees
        Areas4MRFSize = cellfun('length',Areas4MRF.PixelIdxList);
        % trees of size 1
        IndxArea1Px = find(Areas4MRFSize==1);
        Indx4Forest = setdiff(1:Areas4MRF.NumObjects,IndxArea1Px);
        % group objects into Forrests
        ForrestSizeLim = 20000;
        PixelCount = cumsum(Areas4MRFSize(Indx4Forest));
        PixelCount = PixelCount(:)'/ForrestSizeLim;
        ForrestIndx = cell(ceil(PixelCount(end)),1);
        SizeForrest = zeros(ceil(PixelCount(end)),1);
        for ctG = 1:ceil(PixelCount(end))
            ForrestIndx{ctG} = Indx4Forest(find(PixelCount>(ctG-1) & PixelCount<=ctG));
            SizeForrest(ctG) = sum(Areas4MRFSize(ForrestIndx{ctG}));
        end
        ForrestIndx(find(SizeForrest==0)) = [];
        SizeForrest(find(SizeForrest==0)) = [];
        %% === process single-pixel objects
        for ctAREA = IndxArea1Px
            [RWG,CWG] = ind2sub([sx,sy],Areas4MRF.PixelIdxList{ctAREA}); %row-column cords global
            % Corners of image
ImCnr = [max([min(RWG)-1,1]);max([min(CWG)-1,1]);min([max(RWG)+1,sx]);min([max(CWG)+1,sy])];
            szR   = ImCnr(3) - ImCnr(1) + 1;          szC   = ImCnr(4) - ImCnr(2) + 1;
            MaskIndx = sub2ind([szR,szC],RWG - ImCnr(1) + 1,CWG - ImCnr(2) + 1);
            MaskL = false(szR,szC);           MaskL(MaskIndx) = true;
            ImageExtL = ImageExt(ImCnr(1):ImCnr(3),ImCnr(2):ImCnr(4)); % local ImageExt mask
            % neighbor pixels
            ImSegmL = ImSegm(ImCnr(1):ImCnr(3),ImCnr(2):ImCnr(4));
  phiObsIndx = ImSegmL( find(((imdilate(MaskL,strel('disk',1)) & (~MaskL)) & ImageExtL)) );
            % generate phi
            phi = squeeze(DomProbMapPROC(RWG,CWG,:))' + sum(psi(phiObsIndx,:));
```

```
        % if the area has a single pixel, solution is trivial
        [tmp1,tmp2] = max(phi);
        ImSegm(Areas4MRF.PixelIdxList{ctAREA}) = tmp2;
    end
    clear RWG CWG ImCnr szR szC MaskIndx MaskL ImageExtL ImSegmL phiObsIndx phi tmp1 tmp2
    DomProbMapPROC = reshape(DomProbMapPROC,[sx*sy,sz]);
    %% === MRF SEGMENTATION
    fprintf('solving MRF in %s forests',num2str(length(ForrestIndx)))
    for ctG = 1:length(ForrestIndx)
        fprintf('.')
        Obj2Proc = ForrestIndx{ctG};          NumTree  = length(Obj2Proc);
        % initialize
        MaskIndx = cell(NumTree,1);           phi      = cell(NumTree,1);
        nghb     = cell(NumTree,1);           Q        = cell(NumTree,1);
        N  = zeros(1,NumTree);
        for ctT = 1:NumTree
            ctAREA = Obj2Proc(ctT);
            % ===== ===== setup MRF Model for each area
            MaskIndxG = Areas4MRF.PixelIdxList{ctAREA}; % index in global coords
            [RWG,CWG] = ind2sub([sx,sy],MaskIndxG); %row-column cords global
            MaskIndx{ctT} = MaskIndxG;
            % Corners of image
ImCnr = [max([min(RWG)-1,1]);max([min(CWG)-1,1]);min([max(RWG)+1,sx]);min([max(CWG)+1,sy])];
            szR   = ImCnr(3) - ImCnr(1) + 1;          szC   = ImCnr(4) - ImCnr(2) + 1;
            RWL = RWG - ImCnr(1) + 1; % rows in local coords
            CWL = CWG - ImCnr(2) + 1; % cols in local coords
            MaskIndxL = sub2ind([szR,szC],RWL,CWL); % object index in local coords
            clear RWG CWG
            MaskL = false(szR,szC);
            MaskL(MaskIndxL) = true;
            ImageExtL = ImageExt(ImCnr(1):ImCnr(3),ImCnr(2):ImCnr(4)); % local ImageExt mask
            NotExtIndx = find(~ImageExtL);
            % % % generate phi
            DomProbMapPROCL = DomProbMapPROC(MaskIndxG,:); % processed image, local
            DomProbMapPROCL(DomProbMapPROCL==0) = eps;
            phi{ctT} = DomProbMapPROCL;
            clear DomProbMapPROCL
            % % % observables
            ObsPx = (imdilate(MaskL,strel('disk',1)) & (~MaskL)) & ImageExtL;
            ObsPxIndx = find(ObsPx);  % boundary index in local coords
            ImSegmL = ImSegm(ImCnr(1):ImCnr(3),ImCnr(2):ImCnr(4));
            phiObsIndx = ImSegmL(ObsPxIndx);
            N(ctT)  = length(MaskIndxL);
            No = length(ObsPxIndx);
            clear ObsPx ImSegmL
            % % % generate nghb, Q
            nghb{ctT}  = zeros(N(ctT),4);
            Q{ctT}     = zeros(N(ctT),2,'uint32');
            for ct = 1:N(ctT)
                [Row,Col] = ind2sub([szR,szC],MaskIndxL(ct)); % [Row,Col]
                Q{ctT}(ct,:) = [Row,Col];
                Neigb = MaskIndxL(ct) + [-szR -1 1 szR]; % L T B R
                Neigb([Col==1,Row==1,Row==szR,Col==szC]) = NaN;
                % remove neighbors not included in ImageExtL
                [tmp1,tmp2,tmp3] = intersect(Neigb,NotExtIndx);
                Neigb(tmp2) = NaN;     nghb{ctT}(ct,:) = Neigb([4 1 3 2]); % R L B T
            end
            clear tmp1 tmp2 tmp3
            % % % incorporate observed pixels in the model
            for ct = 1:No
                [Row,Col] = ind2sub([szR,szC],ObsPxIndx(ct));
                Neigbo = ObsPxIndx(ct) + [-szR -1 1 szR];
                Neigbo([Col==1,Row==1,Row==szR,Col==szC]) = [];
                [Neigbo,iNeigbo,iMaskIndx] = intersect(Neigbo,MaskIndxL);
                for ctNE = 1:length(iMaskIndx)
                    % calculate nghb
                    nhgbi = nghb{ctT}(iMaskIndx(ctNE),:);
                    nhgbi(find(nhgbi==ObsPxIndx(ct))) = NaN;
                    nghb{ctT}(iMaskIndx(ctNE),:) = nhgbi;
                    % update phi
                    phi{ctT}(iMaskIndx(ctNE),:) = phi{ctT}(iMaskIndx(ctNE),:) + ...
```

```
                        psi(phiObsIndx(ct),:);
                    end
                end
                clear nhgbi
                nghb{ctT} = changem(nghb{ctT},MaskIndxG,MaskIndxL);
            end
            % PREPARE FOREST
            MaskIndxForrest = cell2mat(MaskIndx);
            phiForrest      = cell2mat(phi);
            nghbForrest = changem(cell2mat(nghb),...
                1:length(MaskIndxForrest),MaskIndxForrest);
            QForrest    = cell2mat(Q);
            Forrest     = cell(NumTree,1);
            ctCur = [0;cumsum(cellfun('length',MaskIndx))];
            for ctT = 1:NumTree
                Forrest{ctT} = uint32(ctCur(ctT)+1):ctCur(ctT+1);
            end
            % ===== ===== run MRF in forest data
            MaxNumIt = 120;
            UConv  = 5;
            PrntDU = false;
     ParProcSw = sum(cellfun('length',Forrest))>500; % apply parallel processing if >500 pixels
            ImSegm(MaskIndxForrest) = bp_map_par_DT_8_9(phiForrest, psi, ...
                nghbForrest, QForrest, MaxNumIt,UConv,PrntDU,ParProcSw,Forrest);
        end
    else
        fprintf('No regions identified. ')
    end
    %% === save data
    fprintf('| Save ')
    SaveFileDir =
[Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_ImSegm_',Padding(ctZ,Dat.Nom.PadNum)];
    save(SaveFileDir,'ImSegm')
    %% === print images
    if Dat.ImSegm.SvImgSw
        fprintf('| Print')
        if isempty(Dat.ImDcmp.VoidDomSw);            Dat.ImDcmp.VoidDomSw = 1;        end
        % generate HSV image to print
        ColorBar = floor(linspace(1,Dat.ImDcmp.DomNum+Dat.ImDcmp.VoidDomSw+1,size(ImSegm,1)+1));
        ColorBar(end) = [];
        ImSegm = [ImSegm,nan(size(ImSegm,1),20),repmat(ColorBar',[1,30])]; szI1 = size(ImSegm,1);
        if Dat.ImDcmp.VoidDomSw
           Im2Print = hsv2rgb((double(ImSegm)-2)/(Dat.ImDcmp.DomNum),...
               ones(size(ImSegm)),double([ImageExt,ones(szI1,50)]).*(ImSegm>1));
        else
           Im2Print = hsv2rgb((double(ImSegm))/(Dat.ImDcmp.DomNum+1),...
               ones(size(ImSegm)),double([ImageExt,ones(szI1,50)]).*(ImSegm>1));
        end
        % domain names
        if Dat.ImDcmp.VoidDomSw
            ChNames = 'void_';
        else
            ChNames = '';
        end
        for ctD = 1:Dat.ImDcmp.DomNum
            ChNames = [ChNames,Dat.ImDcmp.DomName{ctD},'_'];
        end
        % file name
        SaveFileSegmDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
            Dat.ImAsbl.SvName,'_z',Padding(ctZ,Dat.Nom.PadNum),'_ImSegmGlob_',ChNames];
        % save
        if (Dat.ImAsbl.ImPrintScale==100)
            imwrite(Im2Print,[SaveFileSegmDir,'.png'],'png')
        else
            imwrite(imresize(Im2Print,Dat.ImAsbl.ImPrintScale/100),...
                [SaveFileSegmDir,'_sc',num2str(Dat.ImAsbl.ImPrintScale),'.png'],'png')
        end
    end
    fprintf('\n')
end % z plane
fprintf('\t\t..... done\n\n')
```

```
return
```

bp_map_par_DT_8_9

```
function [map_seq, map_val, m, midx,UH] = bp_map_par_DT_8_9(phi, psi, nghb, Q,
MaxNumIt,UConv,PrntDU,ParProcSw,Forest)
% This function performs max-sum algorithmin a tree graph where only
% pairwise cliques exist in a parallel fashion.
% ========================================================================
% INPUT
%      phi:  Nxcard   array    - Node potentials
%      psi:  cardxcard array   - Pairwise potentials
%      nghb: Nx4 array         - Node connectivity.  (R L B T)
%                                Assumes 4-connectivity. use NaN if a
%                                neighbor does not exist
%     Q:  Nx2 array            - [Column, Row] coords of each node for
%                                asynchronous belief propagation
%     Forest: Fx1 cell array   - each cell contains the nodes of a tree
%                                if empty, then forest contains 1 tree
%
% OUTPUT
%  map_seq:  Nx1 matrix        - MAP sequence
%                                Sequence of variable values corresponding
%                                to the maximum value of this function
%  map_val:  1x1 scalar        - MAP value
%                                Maximum value of this function
%        m:  Nx4 cell matrix   - Messages (assume 4-nighborhood)
%                                Rows represent the origin of a message,
%                                and columns the destination of a message.
%        midx:Nx4 cell matrix  - argmax(x) of optimal messages (assume 4-nighborhood)
%                                Rows represent the origin of a message,
%                                and columns the destination of a message.
%
% VARIABLES
%        N:  1x1   scalar      - Number of nodes (variables)
%     card:  1x1   scalar      - Cardinality (alphabet) of the variables
%                                (assumed the same for all variables)
%                                Each variable x_i takes values in {1, ..., card}
% Author: geopapa, edits by Dimitrios Tzeranis
% ========================================================================
% Algorithm:
% 1. Initialization
%    for i=1:N
%       for j=1:N(i)
%           m_{i->j}^{0}(x_j) = 0
%       end
%    end
% 2. Apply iteratively until convergence  (for a tree N iterations would
%                                         suffice, where N is the number of
%                                         nodes)
%    for t = 1:iter
%       for i=1:N
%            blf_{i}^{t}(x_i) = log(phi(x_i)) + sum_{k \in N(i)} m_{k->i}^{t}(x_i)
%           for j \in N(i)
%               m_{i->j}^{t+1}(x_j)   = max_{x_i}    { log(psi(x_i,x_j) + blf_{i}^{t}(x_i) -
m_{j->i}^{t}(x_i) }
%               idx_{i->j}^{t+1}(x_j) = argmax_{x_i} { log(psi(x_i,x_j) + blf_{i}^{t}(x_i) -
m_{j->i}^{t}(x_i) }
%            end
%       end
%    end
% 3. Compute the marginals
%       for i=1:N
%           b_i(x_i) = log(phi(x_i)) + sum_{k \in N(i)} m_{k->i}^{t}(x_i)
%       end
% 4. Backtrack to recover an MAP configuration
%    Choose an arbitrary i for a root
%       x_i^{map} = argmax_{x_i} { b_i(x_i) }
%       for j \in N(i)
%           x_j^{map} = idx_{j->i}(x_i^{map})
%       end
```

```
%   Continue backtracking until all nodes' maximum values have been tracked.
% 5. Log-Likelihood and likelihood of maximum configuration are
%    log(p(x_1^{map}, .., x_N^{map})) = b_i{x_i^{map}
%    p(x_1^{map}, ..., x_N^{map})      = exp(b_i{x_i^{map})  ,
%    respectively.
% ====================================================================
%%  INITIALIZATION
% ========== system dimensions
[N,card] = size(phi);    % N: number of nodes, % card: card
% ========== Process Neighbors
% number of neighbors for each node
% first column : RIGHT neighbor. second column: LEFT neighbor. third column:  BOTTOM neighbor
% fourth column: TOP neighbor
% % number of neighbors per node
% nghbNum = uint8(sum(~isnan(nghb),2));
% initialize messages: m(i,j): message sent by pixel i to its j-th neighbor
m = cell(N,4); % messages mij between neighbors
% intialize Orind: m(Orind(cti,ctj)) is the message sent by the j-th
% neighbor of pixel i to i
OriInd = zeros(N,4,'uint32'); % index coordinate of j-th neighbor of node i in the m matrices
% node i has n neighbors. k = nghb{i}(j) is the node that corresponds to the
% j-th neighbor of i. the message from the j-th neighbor of i to node i is
% located at m(Orind(cti,ctj))

if ParProcSw
    parfor cti = 1:N % initialize m and OriInd
        nghbL = nghb;        phiL  = phi;
        NzNb = find(~isnan(nghbL(cti,:))); % find neighbor directions for pixel i
        mCS    = repmat({zeros(card,1,'single')},1,4); % add zeros in non-existing messages
 OriCS  = zeros(1,4,'uint32'); % index of neighbour j of node i in the i-th line of the m matrix
        % directions where i has a neighbor
        for ctj = NzNb
            temp = phiL(nghbL(cti,ctj),:)';          temp(temp<0.1) = 0.1;
            mCS(ctj) = {temp/sum(temp)};
            OriCS(ctj) = find(nghbL(nghbL(cti,ctj),:)==cti);
        end
        OriCS(NzNb) = sub2ind([N,4],uint32(nghbL(cti,NzNb)),OriCS(NzNb));
        OriInd(cti,:) = OriCS;        m(cti,:) = mCS;
    end
else
    for cti = 1:N % initialize m and OriInd
        nghbL = nghb;        phiL  = phi;
         NzNb = find(~isnan(nghbL(cti,:))); % find neighbor directions for pixel i
mCS    = repmat({zeros(card,1,'single')},1,4); % add zeros in non-existing messages
OriCS  = zeros(1,4,'uint32'); % index of neighbour j of node i in the i-th line of the m matrix

        % directions where i has a neighbor
        for ctj = NzNb
            temp = phiL(nghbL(cti,ctj),:)';
            temp(temp<0.1) = 0.1;
            mCS(ctj) = {temp/sum(temp)};
            OriCS(ctj) = find(nghbL(nghbL(cti,ctj),:)==cti);
        end
        OriCS(NzNb) = sub2ind([N,4],uint32(nghbL(cti,NzNb)),OriCS(NzNb));
        OriInd(cti,:) = OriCS;
        m(cti,:) = mCS;
    end

end
clear OriCS nghbi temp
% initialize argmax of optimal messages
midx = cell(N,4);
% initialize energy calculations
Uhis    = zeros(1,MaxNumIt);UhisM   = zeros(1,MaxNumIt); Uhisabs = zeros(1,MaxNumIt);
% initialize QQ
QQ = cell(1,2);QQ{1} = unique(Q(:,1)); QQ{2} = unique(Q(:,2));
QQL = [length(QQ{1}) , length(QQ{2})];
Nodes2ProcQ = cell(2,max(QQL));
for ctA = 1:2
    Nodes2ProcQA = cell(1, max( length(QQ{1}) , length(QQ{2}) ));
    for ctS = 1:length(QQ{ctA})
```

```
        Nodes2ProcQA(ctS) = {  uint32(find(Q(:,ctA)==(QQ{ctA}(ctS))))  };
    end
    Nodes2ProcQ(ctA,:) = Nodes2ProcQA;
end
clear QQ Q
%%  Iteratively update m estimates
% initialize switches
NextIterSw = true; LastIter   = false; ctIt = 0;
if PrntDU;     fprintf('Iter#\tdU[k]\tdU[k-1]:\n'); end
while NextIterSw
    ctIt = ctIt + 1;
    % if this is the last iteration do not do more
    if LastIter;      NextIterSw = false;    end
    % initiate dU
    dU = 0;    dUabs = 0;
    for ctA = 1:2      % AXIS (1: process rows, 2: process columns)        %$8
        dirA = (ctA-1)*2 + [1 2];
        dirNotA = setdiff(1:4,dirA);
        Nghb2A = zeros(2,1,'uint8');
        Nghb2A(1) = (ctA-1)*2 + 1; % which neighbor is being processed (RLBT)
        Nghb2A(2) = ctA*2 +1  - 1; % which neighbor to add during msg calculation
        mNEW  = cell(QQL(ctA),1);     mmNEW = cell(QQL(ctA),1);
        Nodes2ProcQA = Nodes2ProcQ(ctA,:);
        if ParProcSw
            % parallel processing enabled: each core processes a row/column
            parfor ctS = 1:QQL(ctA) % for each row/column
                % generate data used in each processor
                dirNotAL = dirNotA;          Nghb2AL  = Nghb2A;
                % generate data used in each processor
                nghbcs   = nghb;       nghbcs   = nghbcs(Nodes2ProcQA{ctS},:);
                blfC     = phi;           blfC     = log(blfC(Nodes2ProcQA{ctS},:));
                OriIndL  = OriInd;           OriIndL  = OriIndL(Nodes2ProcQA{ctS},:);
                mL       = m;
                for ctN = 1:2 %ctA=1: B/T, ctA=2: R/L
                    % which pixels have an appropriate normal neighbor
                    OriIndC = find(~isnan(nghbcs(:,dirNotAL(ctN))));
                    % add to phi
                    if ~isempty(OriIndC)
          blfC(OriIndC,:) = blfC(OriIndC,:) + cell2mat(mL(OriIndL(OriIndC,dirNotAL(ctN)))')';
                    end
                end
                % prepare zig-zag belief update for each single row/column
                OriInd2A = cell(2,1);
             OriInd2A{1} = find(~isnan(nghbcs(:,Nghb2AL(1))))'; % pixels that have R/B neighbor
          OriInd2A{2} = find(~isnan(nghbcs(:,Nghb2AL(2))))'; % pixels that have L/T neighbor
                OriInd2A{2} = OriInd2A{2}(end:-1:1);
                 % initialize messages
                msgUPD = mL(Nodes2ProcQA{ctS},dirA);
                if LastIter
                    mmidx = cell(length(Nodes2ProcQA{ctS}),2);
                else
                    mmidx = [];
                end
                % zig-zag belief update for each single row/column
                for ctF = 1:2 % 1: back (R (ctA=1) or B(ctA=2)). 2: forth (L (ctA=1) or T(ctA=2))
                    for ctP = OriInd2A{ctF}
                        if isnan(nghbcs(ctP,Nghb2AL(3-ctF)))
                            blf =  blfC(ctP,:)';
                        else
                            blf =  blfC(ctP,:)' + msgUPD{ctP+(-3+2*ctF),ctF};
                        end
                        if LastIter
                            % update m and save midx
                            [max_val, max_idx] = max(log(psi') + repmat(blf,1,card)', [], 2);
                            mmidx{ctP,ctF} = max_idx;
                        else
                            % update m only
                            max_val = max(log(psi') + repmat(blf,1,card)', [], 2);
                        end
                        % normalize & save message
                        msgUPD{ctP,ctF}= max_val - min(max_val);
```

```
                end % pixels of this row/column
            end %back and forth
            % each processor exports its findings
            mNEW{ctS}  = msgUPD;
            if LastIter
                mmNEW{ctS} = mmidx;
            end
            % calculate energy decrease
            z2 = cell2mat(msgUPD) - cell2mat(mL(Nodes2ProcQA{ctS},dirA));
            dU    = dU    + sum(z2(:));    dUabs = dUabs + sum(abs(z2(:)));
        end
    else
        % parallel processing not enabled
        for ctS = 1:QQL(ctA) % for each row/column
            % generate data used in each processor
            dirNotAL = dirNotA;
            Nghb2AL  = Nghb2A;
            % generate data used in each processor
            nghbcs   = nghb;       nghbcs   = nghbcs(Nodes2ProcQA{ctS},:);
            blfC     = phi;         blfC     = log(blfC(Nodes2ProcQA{ctS},:));
            OriIndL  = OriInd;      OriIndL  = OriIndL(Nodes2ProcQA{ctS},:);
            mL       = m;
            for ctN = 1:2 %ctA=1: B/T, ctA=2: R/L
                % which pixels have an appropriate normal neighbor
                OriIndC = find(~isnan(nghbcs(:,dirNotAL(ctN))));
                % add to phi
                if ~isempty(OriIndC)
 blfC(OriIndC,:) = blfC(OriIndC,:) + cell2mat(mL(OriIndL(OriIndC,dirNotAL(ctN)))')';
                end
            end
            % prepare zig-zag belief update for each single row/column
            OriInd2A = cell(2,1);
OriInd2A{1} = find(~isnan(nghbcs(:,Nghb2AL(1))))'; % pixels that have R/B neighbor
OriInd2A{2} = find(~isnan(nghbcs(:,Nghb2AL(2))))'; % pixels that have L/T neighbor
            OriInd2A{2} = OriInd2A{2}(end:-1:1);
             % initialize messages
            msgUPD = mL(Nodes2ProcQA{ctS},dirA);
            if LastIter
                mmidx = cell(length(Nodes2ProcQA{ctS}),2);
            else
                mmidx = [];
            end
            % zig-zag belief update for each single row/column
            for ctF = 1:2 % 1: back (R (ctA=1) or B(ctA=2)). 2: forth (L (ctA=1) or T(ctA=2))
                for ctP = OriInd2A{ctF}
                    if isnan(nghbcs(ctP,Nghb2AL(3-ctF)))
                        blf =  blfC(ctP,:)';
                    else
                        blf =  blfC(ctP,:)' + msgUPD{ctP+(-3+2*ctF),ctF};
                    end
                    if LastIter
                        % update m and save midx
                        [max_val, max_idx] = max(log(psi') + repmat(blf,1,card)', [], 2);
                        mmidx{ctP,ctF} = max_idx;
                    else
                        % update m only
                        max_val = max(log(psi') + repmat(blf,1,card)', [], 2);
                    end
                    % normalize & save message
                    msgUPD{ctP,ctF}= max_val - min(max_val);
                end % pixels of this row/column
            end %back and forth
            % each processor exports its findings
            mNEW{ctS}  = msgUPD;
            if LastIter
                mmNEW{ctS} = mmidx;
            end
            % calculate energy decrease
            z2 = cell2mat(msgUPD) - cell2mat(mL(Nodes2ProcQA{ctS},dirA));
            dU    = dU    + sum(z2(:));     dUabs = dUabs + sum(abs(z2(:)));
        end
```

```
        end
        % assemble updates & calculate errors
        for ctS = 1:QQL(ctA) % for each row/column
            % nodes that belong to this row/column
            m(Nodes2ProcQA{ctS},dirA) = mNEW{ctS};
            if LastIter
                midx(Nodes2ProcQA{ctS},dirA) = mmNEW{ctS};
            end
        end
    end % axis
    % save energy history
    Uhis(ctIt)  = dU;
    UhisM(ctIt) = sum(Uhis(max(1,ctIt-5):ctIt))/(ctIt+1-max(1,ctIt-5));
    Uhisabs(ctIt) = dUabs;
    % one more last iteration?
    if ctIt>2
        if (ctIt >= MaxNumIt) || all(abs(UhisM((ctIt-2):ctIt))<UConv)
            LastIter = true;
        end
    end
    % print energy history
    if ~rem(ctIt,3) && PrntDU
        disp([num2str(ctIt),'  ',num2str(UhisM(ctIt),3),'  ',num2str(UhisM(ctIt-1),3)])
    end
    % if this is last iteration store energy history
    if LastIter
        Uhis    = Uhis(1:ctIt);   UhisM   = UhisM(1:ctIt);   Uhisabs = Uhisabs(1:ctIt);
    end
end
UH = cell(1,3); UH{1} = Uhis; UH{2} = UhisM;  UH{3} = Uhisabs;
clear prev_m blf mmidx mm
% calculate roots
if isempty(Forest)
    % Choose a root randomly
    root = unidrnd(N);    Treeroot = root;
    % find marginal for root
    broot = log(phi(root,:))' + sum(cell2mat(m(OriInd(root,~isnan(nghb(root,:))))),2);
else
    NumTrees = size(Forest,1);
    root    = zeros(1,NumTrees); Treeroot= zeros(1,NumTrees); broot   = zeros(card,NumTrees);
    for ctF = 1:NumTrees % loop over trees
        % Choose a root randomly
        SzTree = length(Forest{ctF});
        Treeroot(ctF) = unidrnd(SzTree);
        root(ctF) = Forest{ctF}(Treeroot(ctF));
        % find marginal for root
        broot(:,ctF) = log(phi(root(ctF),:))' +
sum(cell2mat(m(OriInd(root(ctF),~isnan(nghb(root(ctF),:))))),2);
    end
end
 [tmp, idx_mroot] = max(broot); % Find x_{root}^{map}
clear tmp
% Backtracking
map_seq = btrack9(root, idx_mroot, nghb, OriInd, midx, Forest);
%% Evaluate MAP log-likelihood
lg_map_val = sum(log(phi(sub2ind([N,card],(1:N)',map_seq))));
for cti = 1:N
    NzNb = find(~isnan(nghb(cti,:)));
    lg_map_val = lg_map_val + 0.5*sum(log(psi(sub2ind([card
card],map_seq(cti)*ones(length(NzNb),1),map_seq(nghb(cti,NzNb))))));
end
map_val = exp(lg_map_val); % Evaluate MAP likelihood
return

function map_seq = btrack9(root, idx_mroot, nghb, OriInd, midx, Forest)
% This function recovers the maximum sequence of values by backtracking.
% INPUT
%         N:  1x1   scalar      - Number of nodes (variables)
%      root:  1x1   scalar      - Root
% map_idx_root:  1x1   scalar      - Index corresponding to the maximum value of
%                                the chosen root
```

```
%       nghb:  Nx1 cell array    - Neighbors
%                                  Each element lists the neighbors of the
%                                  corresponding variable
%       midx:  Nx4 cell matrix   - Message maximum indices
%                                  Each element corresponds to the index
%                                  of the origin node that maximizes the
%                                  value of the message of the target node
% OUTPUT
%  map_seq:  Nx1 matrix          - Sequence of variable (node) values
%                                  corresponding to the optimal value
% Original Author: geopapa, Modifications by Dimitris Tzeranis
N = size(nghb,1);map_seq = zeros(N,1);stack = zeros(1,N);
if isempty(Forest)
    NumTrees = 1;
else
    NumTrees = length(Forest);
end
% loop over trees
for ctF = 1:NumTrees
    map_seq(root(ctF)) = idx_mroot(ctF);
    stack(1) = root(ctF);    stackLoc = 1;
    while stackLoc
        cti = stack(stackLoc);
        stack(stackLoc) = 0;        stackLoc = stackLoc - 1;
        nghbi = nghb(cti,:);        NzNb = find(~isnan(nghbi));
        for ctj = NzNb
            map_seq(nghbi(ctj)) = midx{OriInd(cti,ctj)}(map_seq(cti));
            nghb(nghbi(ctj),nghb(nghbi(ctj),:)==cti) = NaN;
        end
        temp = setdiff(nghb(cti,:),stack(1:stackLoc));
        temp(isnan(temp)) = [];
        tempL = length(temp);
        stack((stackLoc+1):(stackLoc+tempL)) = temp;
        stackLoc = stackLoc + tempL;
    end
end
return
```

CellScObj

```
function ZPlanes = CellScObj_1_8(Dat)
% identify cell objects in segmented images of cells
% Dimitrios Tzeranis, September 2012 - Jan 2013
% ========================================================================
%                              CODE OVERVIEW
% parameter input
% loop over Z planes
%     read plane data
%     generate cytoplasm, cell, nuclei,matrix masks
%     initialize cell objects
%     probe user to add/remove cells or stop
%         update & display cell-scaffold image w highlighted chosen cells
%         user picks to add/remove cells
%         if input==add
%             process local image
%             identify nuclei objects
%             identify peaks via hmin calculation
%             edit seeds
%             segment cytoplasm using min(grayscale-weighted distance)
%             store segmented cytoplasms that contain nuclei
%         elseif input==remove
%             pick an area to process
%             identify & save nucleus
%         else
%             exit while loop
%         end
%     end
%     for [process each cell object]
%       identify nucleus pixels
%       identify cell pixels
%       identify cytoplasm pixels
```

```
%       nucleus centroid
%       identify cell-matrix & cell-void boundaries
%     end
% end
fprintf('\tProcessing segmenting cell-scaffold images')
  [unused1,unused2,unused3] = mkdir(Dat.CellScObj.Dir);
 VoidDomain   = 1;
NuclDomain   = Dat.CellScObj.NuclDom; %2
CytoDomain   = Dat.CellScObj.CytoDom; %3
MatrixDomain = Dat.CellScObj.ScafDom; %4
CellTrkComp = 2;  % fluorescent component that contains cell-tracker
NucStainComp = 1; % fluorescent component that contains nuclei
 PxSzIm = Dat.Img.PxSz*(40/Dat.Img.Magn)*(Dat.Img.Res/32);
 disk1 = strel('disk',1); % disk for dilation operations
ZPlanes = Dat.CellScObj.ZRange(1):Dat.CellScObj.ZRange(2);
 fprintf('\n\tPlane:')
%% main loop
for ctZZ = 1:length(ZPlanes);
    ctZ = ZPlanes(ctZZ);     fprintf(' %s',num2str(ctZ))
        % ===== read and process plane data
FileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,...
        '_ImSegm_',Padding(ctZ,Dat.Nom.PadNum)];
    load(FileDir) % reads 'ImSegm'
    FileDir = [Dat.ImAsbl.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_',...
        Dat.ImAsbl.SvName,'_',Padding(ctZ,Dat.Nom.PadNum)];
    load(FileDir) % reads 'AsblCompConMap' (full unmixing)
    AsblCompConMap = AsblCompConMap{1};
     NumRowsG = size(ImSegm,1);    NumColsG = size(ImSegm,2);
    % ===== generate cytoplasm, cell, nuclei, matrix, void masks
    MaskCytoG = ImSegm == CytoDomain;    MaskNuclG = ImSegm == NuclDomain;
    MaskCellG = MaskCytoG | MaskNuclG;     MaskMatrG = ImSegm == MatrixDomain;
    MaskVoidG = ImSegm == VoidDomain;
        % ===== initialize temporary cell objects
    CellOBJupd = struct('PxNcl',[],'PxCyto',[],'PxMatrixBnd',[],'PxVoidBnd',[],...
        'PxCellBnd',[],'Centroid',[]);
    NumCellOBJupd = 0;
    %% ===== probe user to add/remove cells or stop (done)
    fig1 = figure;     ContinueProc = true;
    while ContinueProc
        % ===== create & display cell-scaffold image with highlighted chosen cells
        % create
        Im4Pick = single(1/2*cat(3,MaskCytoG,MaskMatrG,MaskNuclG));
        Im4Pick = reshape(Im4Pick,[NumRowsG*NumColsG 3]);
        for ctCELL = 1:NumCellOBJupd
            NumCellPxNcl = length(CellOBJupd(ctCELL).PxNcl);
            Im4Pick(CellOBJupd(ctCELL).PxNcl,1) = zeros(NumCellPxNcl,1,'single');
Im4Pick(CellOBJupd(ctCELL).PxNcl,2) = ones(NumCellPxNcl,1,'single')*(rand*0.1+0.05);
Im4Pick(CellOBJupd(ctCELL).PxNcl,3) = ones(NumCellPxNcl,1,'single')*(rand*0.4+0.6);
            NumCellPxCyto = length(CellOBJupd(ctCELL).PxCyto);
Im4Pick(CellOBJupd(ctCELL).PxCyto,1) = ones(NumCellPxCyto,1,'single')*(rand*0.4+0.6);
Im4Pick(CellOBJupd(ctCELL).PxCyto,2) = ones(NumCellPxCyto,1,'single')*(rand*0.2+0.05);
            Im4Pick(CellOBJupd(ctCELL).PxCyto,3) = zeros(NumCellPxCyto,1,'single');
        end
        Im4Pick = reshape(Im4Pick,[NumRowsG NumColsG 3]);
        figure(fig1)
        imshow(Im4Pick)
        for ctCELL = 1:NumCellOBJupd
            text(round(CellOBJupd(ctCELL).Centroid(2)),...
                round(CellOBJupd(ctCELL).Centroid(1)),num2str(ctCELL),'Color',[1 1 1])
        end
        % ===== user input (add/remove/done)
        title('"a": add cells (mono-nuclei), "b": add cells (nuclei split), "r": remove cells,
"s": done','FontSize',16,'Color',[1 0 0])
        AcceptCell = input('type "a" to add cells (mono-nuclei), "b" to add cells (nuclei split),
"r" to remove cells, or "s" if done   ','s');         title('')
        if strcmpi(AcceptCell,'a') | strcmpi(AcceptCell,'b')
 title('pick a new cell and press enter. press enter without clicking to finish',...
                'FontSize',16,'Color',[1 0 0])
            [ColcM,RowcM] = getpts;
            if ~isempty(RowcM)
                for ctPP = 1:length(RowcM) % loop over all points chosen
```

```
                    Colc = round(ColcM(ctPP));      Rowc = round(RowcM(ctPP));
                    if ~MaskCellG(Rowc,Colc)
                        continue
                        disp('invalid location identified. need to click on a cell')
                    end
                    % ===== identify local window
                     temp  = false([NumRowsG,NumColsG]);    temp(Rowc,Colc) = true;
CellCytoD = imreconstruct(temp, MaskCellG); % identify nuclei-cytoplasm region
                    clear temp
 [CellCytoRowG,CellCytoColG] = ind2sub([NumRowsG,NumColsG],find(CellCytoD));
                    CellCytoRowLim = [min(CellCytoRowG),max(CellCytoRowG)];
                    CellCytoColLim = [min(CellCytoColG),max(CellCytoColG)];
                    sRL = CellCytoRowLim(2) - CellCytoRowLim(1) + 1;
                    sCL = CellCytoColLim(2) - CellCytoColLim(1) + 1;
CellCytoD = CellCytoD(CellCytoRowLim(1):CellCytoRowLim(2),CellCytoColLim(1):CellCytoColLim(2));
CytoD = CellCytoD &
MaskCytoG(CellCytoRowLim(1):CellCytoRowLim(2),CellCytoColLim(1):CellCytoColLim(2));
NucD = CellCytoD & ...
MaskNuclG(CellCytoRowLim(1):CellCytoRowLim(2),CellCytoColLim(1):CellCytoColLim(2));
ScafD = MaskMatrG(CellCytoRowLim(1):CellCytoRowLim(2),CellCytoColLim(1):CellCytoColLim(2));
                    clear CellCytoRowG CellCytoColG
                    % =====  process local image
                    % (low-pass filter cell-tracker sugnal)
FltCellTrkImage =
filter2(fspecial('gaussian',20,10),double(AsblCompConMap(CellCytoRowLim(1):CellCytoRowLim(2),
CellCytoColLim(1):CellCytoColLim(2),CellTrkComp)));
                    ImGradAbsLLim = 0.1;
ImInt = filter2(fspecial('gaussian',5,3),
double(AsblCompConMap(CellCytoRowLim(1):CellCytoRowLim(2),...
CellCytoColLim(1):CellCytoColLim(2),CellTrkComp)) );
ImGradAbs  = cat(3,0.5*(-[zeros(sRL,1),ImInt(:,1:end-1)] + [ImInt(:,2:end),zeros(sRL,1)]),0.5*(-
[zeros(1,sCL);ImInt(1:end-1,:)] + [ImInt(2:end,:);zeros(1,sCL)]));
ImGradAbs = sqrt(sum(ImGradAbs.^2,3)); ImGradAbs(ImGradAbs<=ImGradAbsLLim) = 0;
                    clear ImInt
                    % ===== identify nucleus objects
                    MinNucDiam = 4; % lower limit of nucleus diameter [um]
                    MinNucAreaPx = round(pi/4*(MinNucDiam/PxSzIm)^2);
                    NucObj = bwconncomp(NucD,4);   NucIndx = cell(0);      NucNum = 0;
                    for ctC = 1:NucObj.NumObjects
                        if length(NucObj.PixelIdxList{ctC})>MinNucAreaPx
                            if strcmpi(AcceptCell,'b') % split nuclei
            hminNUCLim = 8;    % hmin limit for nuclei peaks inned dist
            minPeakInDist = 5; % min distance of a nucleus peak from nucleus border
         MinNucPkDist = 10; % min distance between neighboring cytoplasm peaks [Px]
    % === identify cell possible cell objects based on peak of outer distance function
        NucIm = false(sRL,sCL);      NucIm(NucObj.PixelIdxList{ctC}) = true;
                                NucIm = imfill(NucIm,4,'holes'); % fill small holes
                                DistN = bwdist(~NucIm,'Quasi-Euclidean');
                                % find peaks based on outer distance
PeaksNuc = imregionalmax(imhmin(DistN,min([hminNUCLim,0.75*max(DistN(:))])));
                                NucSeedIndx = find(PeaksNuc);
                                % sort peaks by inner distance
                                [x1,NucPeakSort] = sort(DistN(NucSeedIndx));
                                NucSeedIndx = NucSeedIndx(NucPeakSort);
                                % keep peaks located enough inside nucleus
                                Peak2Del = DistN(NucSeedIndx)<minPeakInDist;
                                if all(Peak2Del),Peak2Del(end) = false;end
                                NucSeedIndx(Peak2Del) = [];
                                %  coordinates of initial estimates
  [PeakNucRow,PeakNucCol] = ind2sub([sRL,sCL],NucSeedIndx);
              PeaksNucRT = [PeakNucRow,PeakNucCol]';
                                % check if peaks are close to each other
                                checkV = false(1,length(NucSeedIndx));
                                for ctP = 1:(length(NucSeedIndx)-1)
disti = sqrt(sum((repmat(PeaksNucRT(:,ctP),1,length(NucSeedIndx)-ctP) -
PeaksNucRT(:,ctP+1:end)).^2));
                                    if any(disti<MinNucPkDist)
              closepeak = [ctP,ctP + find(disti<MinNucPkDist)];
                                        checkV(ctP) = true;
PeaksNucRT(:,closepeak) = repmat(round(mean(PeaksNucRT(:,closepeak),2)),1,length(closepeak));
                                    end
```

```
                                end
  NucSeedIndx(checkV) = [];         PeaksNucRT(:,checkV)= [];
                                NumNucFnd = length(NucSeedIndx);
                                % ===== display to user and ask feedback
                                if NumNucFnd>1
                                    FigPickNuc = figure;
[NucIR,NucIC] = ind2sub([sRL,sCL],NucObj.PixelIdxList{ctC});
subplot(1,2,1),imshow(NucIm(min(NucIR):max(NucIR),min(NucIC):max(NucIC)))
          tmp = DistN;                  tmp(NucSeedIndx) = 0;
subplot(1,2,2),imshow(tmp(min(NucIR):max(NucIR),min(NucIC):max(NucIC)))
                                    caxis([0 20]),colormap hot
                                    for ctP = 1:NumNucFnd
text(PeaksNucRT(2,ctP)-min(NucIC),PeaksNucRT(1,ctP)-min(NucIR)+2,num2str(ctP))
                                    end
                                    clear tmp NucIR NucIC
title('provide nuclei peaks to keep'),    KeepPeakChek = true;
                                    while KeepPeakChek
Peaks2Keep = input('Provide numbers of nuclei peaks to keep  ','s');
                                        if isempty(Peaks2Keep)
                                            Peaks2Keep = [];
                                            KeepPeakChek = false;
                                        else
                                            if ~isempty(str2num(Peaks2Keep))
                Peaks2Keep = round(str2num(Peaks2Keep));
                Peaks2Keep = Peaks2Keep(Peaks2Keep<=NumNucFnd);
                                                KeepPeakChek = false;
                                            end
                                        end
                                    end
                                    close(FigPickNuc)
                                    NucSeedIndx = NucSeedIndx(Peaks2Keep);
                                    PeaksNucRT  = PeaksNucRT(:,Peaks2Keep);
                                    NumNucFnd = length(NucSeedIndx);
                                end
                                if NumNucFnd>1
                                    % ===== segment nuclei region based on seeds
                                    DistNLIM = 10; LoopChk = true; LoopNum = 0;
                                    while LoopChk & LoopNum<5
                                        LoopNum = LoopNum + 1;
                                        DistN(DistN>DistNLIM) = DistNLIM;
                                        DistN = DistNLIM - DistN;
                                        NucIm = false(sRL,sCL);
                                        NucIm(NucObj.PixelIdxList{ctC}) = true;
                                        Dist = zeros(sRL,sCL,NumNucFnd);
                                        for ctS = 1:NumNucFnd
                                            % calculate gray-weighted distance
Dist(:,:,ctS) = graydist(double(DistN),PeaksNucRT(2,ctS),PeaksNucRT(1,ctS),'quasi-
euclidean').*double(NucIm);
                                        end
 [x1,NucSegm] = min(reshape(Dist,[sRL*sCL,NumNucFnd]),[],2);
                                        NucSegm = reshape(NucSegm,[sRL,sCL]).*NucIm;
                                        CheckV = zeros(NumNucFnd,1);
                                        for ctSS = 1:NumNucFnd
                            CheckV(ctSS) = sum(sum( NucSegm == ctSS  ))>0;
                                        end
                                        if all(CheckV)
                                            LoopChk = false;
                                        else
                                            DistNLIM = DistNLIM + 5;
                                        end
                                    end
                                    Nuc2Use = find(CheckV>0);
                                    NucSeedIndx = NucSeedIndx(Nuc2Use);
                                    PeaksNucRT  = PeaksNucRT(:,Nuc2Use);
                                    NumNucFnd   = length(NucSeedIndx);
                                     NucPxFnd = cell(NumNucFnd,1);
                                    for ctN = 1:NumNucFnd
                                        NucPxFnd{ctN} = find(NucSegm == Nuc2Use(ctN));
                                    end
                        disp(['cell split into ',num2str(NumNucFnd),' parts'])
                                elseif NumNucFnd==1
```

```
                    NumNucFnd = 1;        NucPxFnd = {NucObj.PixelIdxList{ctC}};
                                else
                                    NumNucFnd = 0;  NucPxFnd = {};
                                end
                            else  % do not split nuclei.. single nucleus
                                NumNucFnd = 1; NucPxFnd = {NucObj.PixelIdxList{ctC}};
                            end
                            for ctNUc = 1:NumNucFnd
                                NucNum = NucNum + 1;
                                NucIndx{NucNum} = NucPxFnd{ctNUc};
                            end
                        end
                    end
                    clear FltNucImage NucIm  NucSegm Dist DistN x1 NucPxFnd
                    % generate Peak indices
                    PeaksInNuc = zeros(2,NucNum);
                    for ctC = 1:NucNum
                        [rr,cc] = ind2sub([sRL,sCL],NucIndx{ctC});
                        PeaksInNuc(:,ctC) = round([mean(rr);mean(cc)]);
                    end
                    clear rr cc
                    if NucNum>1
                        NucPix = zeros(sRL,sCL);
                        for ctC = 1:NucNum
                            NucPix(NucIndx{ctC}) = 0.4+0.6*rand;
                        end
                        NucPix2Ev = cat(3,CytoD,CytoD*0,NucPix);
                        figN2c = figure;    imshow(NucPix2Ev)
                        for ctC = 1:NucNum
text(PeaksInNuc(2,ctC),PeaksInNuc(1,ctC),num2str(ctC),'Color',[1 1 1])
                        end
                        title('provide nuclei to keep')
                        KeepPeakChek = true;
                        while KeepPeakChek
 Peaks2Keep = input('Provide numbers of nuclei peaks to keep  ("a" for all)    ','s');
                            if isempty(Peaks2Keep)
                                Peaks2Keep = [];
                            else
                                if strcmpi(Peaks2Keep,'a')
                                    Peaks2Keep = 1:NucNum;
                                    KeepPeakChek = false;
                                elseif ~isempty(str2num(Peaks2Keep))
                                    Peaks2Keep = str2num(Peaks2Keep);
                                    Peaks2Keep = Peaks2Keep(Peaks2Keep<=NucNum);
                                    KeepPeakChek = false;
                                end
                            end
                        end
                        NucIndx = NucIndx(Peaks2Keep);  NucNum = length(NucIndx);
                        PeaksInNuc = PeaksInNuc(:,Peaks2Keep);
                        close(figN2c)
                        clear NucPix2Ev NucPix
                    end
       PeaksInNucIndx = sub2ind([sRL,sCL],PeaksInNuc(1,:),PeaksInNuc(2,:));
                    if NucNum>0
                        % ===== identify & process non-nuclei peaks
          hminCytoLim = 5; % hmin limit for cyto peaks (grayscale values)
            MinCytoPkDist = 20; % min distance between neighboring cytoplasm peaks [Px]
          minPeakONucDist = 10; % minimum distance of non-nuclei peaks from nuclei
           PeaksIm = imregionalmax(imhmin(FltCellTrkImage,hminCytoLim).*CellCytoD);
                        % do not consider peaks located inside or very close to nuclei
          PeaksIm = PeaksIm.*(~imdilate(NucD,strel('disk',minPeakONucDist)));
                        PeaksOutNucIndx = find(PeaksIm);
                         [PeakRow,PeakCol] = ind2sub([sRL,sCL],PeaksOutNucIndx);
                        PeaksR = [PeakRow,PeakCol]';
                        % treat peaks very close to each other as one
                        checkV = false(1,length(PeakRow));
                        for ctP = 1:(length(PeakRow)-1)
disti = sqrt(sum((repmat(PeaksR(:,ctP),1,length(PeakRow)-ctP) - PeaksR(:,ctP+1:end)).^2));
                            if any(disti<MinCytoPkDist)
                                closepeak = [ctP,ctP + find(disti<MinCytoPkDist)];
```

```
                                checkV(ctP) = true;
PeaksR(:,closepeak) = repmat(round(mean(PeaksR(:,closepeak),2)),1,length(closepeak));
                            end
                        end
                        PeaksOutNucIndx(checkV) = [];
                        NumSdONuc = length(PeaksOutNucIndx);
                        clear PeaksR PeaksIm
                        if NumSdONuc
                            % ===== user picks which seeds to keep
                            fUfbk = figure;          subplot(2,1,1)
                            imshow(single(cat(3,CytoD,ScafD,NucD)))
                            subplot(2,1,2),     tmp = FltCellTrkImage.*CytoD;
                            tmp = tmp(end:-1:1,:);      contour3(tmp,10)
                            hold on
                            for ctP = 1:NumSdONuc
                                [pkr,pkc] = ind2sub([sRL,sCL],PeaksOutNucIndx(ctP));
            plot3(pkc,sRL-pkr+1,tmp(sRL-pkr+1,pkc)+0.1,'bo','LineWidth',2)
                                text(pkc+3,sRL-pkr+3,num2str(ctP))
                            end
                            view([0 90]),   grid off,    clear  pkr pkc tmp
                            title(' Provide numbers of non-nuclei peaks to keep')
                            PeakChek = true;
                            while PeakChek
 Peak2Keep = input(['Provide numbers of non-nuclei peaks to keep ("a" for all)    '],'s');
                                if isempty(Peak2Keep)
                                    Peak2Keep = []; PeakChek = false;
                                else
                                    if strcmpi(Peak2Keep,'a')
                                        Peak2Keep = 1:NumSdONuc;   PeakChek = false;
                                    elseif ~isempty(str2num(Peak2Keep))
                                        Peak2Keep = round(str2num(Peak2Keep));
                                        Peak2Keep = Peak2Keep(Peak2Keep<=NumSdONuc);
                                        PeakChek = false;
                                    end
                                end
                            end
                            close(fUfbk)
                            PeaksOutNucIndx = PeaksOutNucIndx(Peak2Keep);
                            NumSdONuc   = length(PeaksOutNucIndx);
                        else
                            PeaksOutNucIndx = [];
                            NumSdONuc = 0;
                        end
                         % ===== prepare seeds
                        % number of seeds
   NumSeed = NucNum + NumSdONuc; % first nuclei-seed, then non-nuclei seed
                        PeaksIndxT = [PeaksInNucIndx(:)',PeaksOutNucIndx(:)'];
                        [PeakRowT,PeakColT] = ind2sub([sRL,sCL],PeaksIndxT);
                        PeaksRT = [PeakRowT(:),PeakColT(:)]';
                        % ===== cytoplasm segmentation
                        DistPLIM = 20; % distance transform pixel distance
                        lambdaDi = 0.4; % contribution of distance function
                        lambdaDe = 0.3; % contribution of derivative function
                        lambdaCa = 1-lambdaDi - lambdaDe;
                        if lambdaCa<0
                            lambdaDi = lambdaDi/(1-lambdaCa);
                            lambdaDe = lambdaDe/(1-lambdaCa);  lambdaCa = 0;
                        end
                        % generate distance transform
                        DistP = bwdist(~CellCytoD,'cityblock');
                        DistP(DistP>DistPLIM) = DistPLIM;  DistP = DistPLIM - DistP;
                        % ===== calculate distance function for all seeds
                        Dist = zeros(sRL,sCL,NumSeed);
                        for ctC = 1:NumSeed
                            % function contributions
                            % eucledian distance contribution
                            DelayM = lambdaCa*single(CellCytoD);
                            % outer distance contribution
                            DelayM = DelayM + lambdaDi*DistP;
                            % derivative magnitude contribution
                            DelayM = DelayM + lambdaDe*ImGradAbs;
```

```
                            % nuclei corrections
                            for ctCC = 1:NucNum
                                if ctC == ctCC
                                    DelayM(NucIndx{ctCC}) = eps;
                                else
                                    DelayM(NucIndx{ctCC}) = inf;
                                end
                            end
                            % non-cell area correction
                            DelayM(CellCytoD==0) = inf;
                            % calculate gray-weighted distance
 Dist(:,:,ctC) = graydist(DelayM,PeaksRT(2,ctC),PeaksRT(1,ctC),'quasi-euclidean');
                        end
                        clear DelayM
                        % at each pixel, find corresponding seed for min distance
                        [x1,NucCytoSegm] = min(reshape(Dist,[sRL*sCL,NumSeed]),[],2);
                        NucCytoSegm = reshape(NucCytoSegm,[sRL,sCL]).*CellCytoD;
                        clear x1
                        % ===== store segmented cytoplasm that correspond to nuclei
                        for ctCELL = 1:NucNum
                            NumCellOBJupd = NumCellOBJupd + 1;

 [RowL,ColL] = ind2sub([sRL,sCL],find((NucCytoSegm==ctCELL).*NucD));
                            rCi = RowL - 1 + CellCytoRowLim(1);
                            cCi = ColL - 1 + CellCytoColLim(1);
   CellOBJupd(NumCellOBJupd).PxNcl = sub2ind([NumRowsG,NumColsG],rCi,cCi);
    CellOBJupd(NumCellOBJupd).Centroid = [round(mean(rCi)),round(mean(cCi))];
     [RowL,ColL] = ind2sub([sRL,sCL],find((NucCytoSegm==ctCELL).*CytoD));
      CellOBJupd(NumCellOBJupd).PxCyto = sub2ind([NumRowsG,NumColsG],...
            RowL - 1 + CellCytoRowLim(1),ColL - 1 + CellCytoColLim(1));
                            end
                        end
                    end
                end
                clear RowL ColL rCi cCi CytoD ScafD NucD CellCytoD NucCytoSegm
                clear ImGradAbs FltCellTrkImage DistP Dist
            elseif strcmpi(AcceptCell,'r')
title('type the cells to remove in format [n m l]','FontSize',16,'Color',[1 0 0])
                DelPeakChek = true;
                while DelPeakChek
    Nuc2Rem = input('provide numbers of cells to remove  (e.g. 1 5 9)  ','s');
                    if isempty(Nuc2Rem)
                        Nuc2Rem = [];
                        DelPeakChek = false;
                    else
                        if ~isempty(str2num(Nuc2Rem))
   Nuc2Rem = round(str2num(Nuc2Rem));     Nuc2Rem = Nuc2Rem(Nuc2Rem<=NumCellOBJupd);
                            DelPeakChek = false;
                        end
                    end
                end
                title('');       CellOBJupd(Nuc2Rem) = [];
                NumCellOBJupd = length(CellOBJupd);
                clear Nuc2Rem
            elseif strcmpi(AcceptCell,'s')
                ContinueProc = false;
            else
                ContinueProc = true;
            end
        end
        close(fig1)
        clear AcceptCell AsblCompConMap
        %% process cell objects to identify cell-matrix, cell-cell, cell-void interfaces
        % check cell objects
        NumCellOBJupd = length(CellOBJupd);  CheckCell = false(1,NumCellOBJupd);
        for ctCELL = 1:NumCellOBJupd
            CheckCell(ctCELL) = (~isempty(CellOBJupd(ctCELL).PxNcl)) &
(~isempty(CellOBJupd(ctCELL).PxCyto));
        end
        CellOBJupd = CellOBJupd(find(CheckCell));     NumCellOBJupd = length(CellOBJupd);
        clear CheckCell
```

```
    CellsOBJ = struct('Connectivity',4,'ImageSize',[NumRowsG NumColsG],...
        'NumObj', 0,'Cell',struct(...
'PxNcl',[],'PxCyto',[],'PxMatrixBnd',[],'PxVoidBnd',[],'PxCellBnd',[],'Centroid',[]));
    CellsOBJ.Cell   = CellOBJupd;  CellsOBJ.NumObj = length(CellsOBJ.Cell);
    Im4Pick = single(1/3*cat(3,MaskCytoG,MaskMatrG,MaskNuclG));
    Im4Pick = reshape(Im4Pick,[NumRowsG*NumColsG 3]);
    for ctCELL = 1:CellsOBJ.NumObj  % for each cell object
        CellPxNcl = CellsOBJ.Cell(ctCELL).PxNcl;
        NumCellPxNcl = length(CellPxNcl);
        CellPxCyto = CellsOBJ.Cell(ctCELL).PxCyto;
        NumCellPxCyto= length(CellPxCyto);
        CytoIm = false(NumRowsG,NumColsG);
        CytoIm(CellPxCyto) = true;
CellsOBJ.Cell(ctCELL).PxMatrixBnd = find(CytoIm &imdilate(MaskMatrG,strel('disk',1)));
CellsOBJ.Cell(ctCELL).PxVoidBnd = find(CytoIm & imdilate(MaskVoidG,strel('disk',1)));
CellsOBJ.Cell(ctCELL).PxCellBnd = find(CytoIm & imdilate(MaskCytoG & (~CytoIm),strel('disk',1)));
        Im4Pick(CellPxNcl,1) = zeros(NumCellPxNcl,1,'single');
        Im4Pick(CellPxNcl,2) =  ones(NumCellPxNcl,1,'single')*(rand*0.1+0.05);
        Im4Pick(CellPxNcl,3) =  ones(NumCellPxNcl,1,'single')*(rand*0.4+0.6);
        Im4Pick(CellPxCyto,1) =  ones(NumCellPxCyto,1,'single')*(rand*0.4+0.6);
        Im4Pick(CellPxCyto,2) =  ones(NumCellPxCyto,1,'single')*(rand*0.1+0.05);
        Im4Pick(CellPxCyto,3) = zeros(NumCellPxCyto,1,'single');
    end
    Im4Pick = reshape(Im4Pick,[NumRowsG NumColsG 3]);
    if Dat.CellScObj.SvImgSw | Dat.CellScObj.SaveSw
        SaveFileDir = [Dat.CellScObj.Dir,Dat.Nom.DASH,Dat.Nom.fname,...
            '_CellScObj_z',Padding(ctZ,Dat.Nom.PadNum)];
    end
    if Dat.CellScObj.SvImgSw
        ImPickedCells = reshape(1/3*cat(3,MaskCytoG,MaskMatrG,MaskNuclG),...
            [NumRowsG*NumColsG,3]);
        ImPickedCells = rgb2hsv(ImPickedCells);
        ImPickedCells(:,3) = ImPickedCells(:,3)/2;
        ImPickedCells = hsv2rgb(ImPickedCells);
        for ctCELL = 1:CellsOBJ.NumObj
            ImPickedCells(CellsOBJ.Cell(ctCELL).PxNcl,3) = 1;
            ImPickedCells(CellsOBJ.Cell(ctCELL).PxCyto,1) = rand*0.5+0.5;
            ImPickedCells(CellsOBJ.Cell(ctCELL).PxMatrixBnd,1) = 1;
            ImPickedCells(CellsOBJ.Cell(ctCELL).PxMatrixBnd,2) = 1;
            ImPickedCells(CellsOBJ.Cell(ctCELL).PxMatrixBnd,3) = 0;
            ImPickedCells(CellsOBJ.Cell(ctCELL).PxVoidBnd,1) = 1;
            ImPickedCells(CellsOBJ.Cell(ctCELL).PxVoidBnd,2) = 0;
            ImPickedCells(CellsOBJ.Cell(ctCELL).PxVoidBnd,3) = 1;
            ImPickedCells(CellsOBJ.Cell(ctCELL).PxCellBnd,1) = 0;
            ImPickedCells(CellsOBJ.Cell(ctCELL).PxCellBnd,2) = 1;
            ImPickedCells(CellsOBJ.Cell(ctCELL).PxCellBnd,3) = 1;
        end
        ImPickedCells = reshape(ImPickedCells,[NumRowsG,NumColsG,3]);
        fig2 = figure;   imshow(ImPickedCells)
        for ctCELL = 1:CellsOBJ.NumObj
text(CellsOBJ.Cell(ctCELL).Centroid(2),CellsOBJ.Cell(ctCELL).Centroid(1),num2str(ctCELL),'FontSiz
e',48,'Color',[1 1 1])
        end
        saveas(fig2,[SaveFileDir,'.png'],'png')
        close (fig2)
    end
    if Dat.CellScObj.SaveSw;    save(SaveFileDir,'CellsOBJ');   end
end
fprintf('\n\t\t..... done\n\n')
return
```

CellScMetr

```
function ZPlanes = CellScMetr_1_2(Dat)
% calculate cell metrics of cell objects in cell-matrix mages
% Dimitrios Tzeranis, Sep 2012 - Jan 2013
fprintf('=== Calculate metrics of cell objects in cell-scaffold images ===\n')
[unused1,unused2,unused3] = mkdir(Dat.CellScObj.Dir);
clear unused1 unused2 unused3
VoidDomain   = 1;NuclDomain   = Dat.CellScObj.NuclDom; %2
CytoDomain   = Dat.CellScObj.CytoDom; %3
```

```
MatrixDomain = Dat.CellScObj.ScafDom; %4
CellTrkComp  = 2;  % fluorescent component that contains cell-tracker
NucStainComp = 1; % fluorescent component that contains nuclei

ZPlanes = Dat.CellScObj.CellMtrZRange(1):Dat.CellScObj.CellMtrZRange(2);
PxSzA = Dat.Img.PxSz*(40/Dat.Img.Magn)*(Dat.Img.Res/32);
% === metric description
% nucleus
NumMtNuc  = 12;
IDNU      = cell(1,NumMtNuc);DescrNU   = cell(1,NumMtNuc);
IDNU{1}   = 'NA';DescrNU{1}='Nucleus Area [um2]';
IDNU{2}   = 'NP';DescrNU{2}='Nucleus Perimeter [um]';
IDNU{3}   = 'NSF';DescrNU{3}='Nucleus Shape Factor';
IDNU{4}   = 'NE';DescrNU{4}='Nucleus Eccentricity';
IDNU{5}   = 'NTIA';DescrNU{5}='Nucleus Total Intensity Average [P.C.]';
IDNU{6}   = 'NTIS';DescrNU{6}='Nucleus Total Intensity Std [P.C.]';
IDNU{7}   = 'NHIA';DescrNU{7}='Nucleus Hoechst Intensity Average [P.C.]';
IDNU{8}   = 'NHIS';DescrNU{8}='Nucleus Hoechst Intensity Std [P.C.]';
IDNU{9}   = 'NCIA';DescrNU{9}='Nucleus CMTMR Intensity Average [P.C.]';
IDNU{10}   = 'NCIS';DescrNU{10}='Nucleus CMTMR Intensity Std [P.C.]';
IDNU{11}   = 'NGCO';DescrNU{11}='Nucleus Grayscale Centroid Offset [um]';
IDNU{12}   = 'NCO';DescrNU{12}='Nucleus Cell Centroids Offset [um]';
% cytoplasm
NumMtCyto = 15;
IDCY      = cell(1,NumMtCyto);DescrCY   = cell(1,NumMtCyto);
IDCY{1}   = 'CA';DescrCY{1}='Cytoplasm Area [um2]';
IDCY{2}   = 'CP';DescrCY{2}='Cell Perimeter [um]';
IDCY{3}   = 'CSF';DescrCY{3}='Cell Shape Factor';
IDCY{4}   = 'CE';DescrCY{4}='Cell Eccentricity';
IDCY{5}   = 'CPNPR'DescrCY{5}= 'Cell  Perimeter to Nucleus Perimeter Ratio';
IDCY{6}   = 'CTIA';DescrCY{6}='Cytoplasm Total Intensity Average [P.C.]';
IDCY{7}   = 'CTIS';DescrCY{7}='Cytoplasm Total Intensity Std [P.C.]';
IDCY{8}   = 'CAIA';DescrCY{8}='Cytoplasm Alexa488 Intensity Average [P.C.]';
IDCY{9}   = 'CAIS';DescrCY{9}='Cytoplasm Alexa488 Intensity Std [P.C.]';
IDCY{10}   = 'CCIA';DescrCY{10}='Cytoplasm CMTMR Intensity Average [P.C.]';
IDCY{11}   = 'CCIS';DescrCY{11}='Cytoplasm CMTMR Intensity Std [P.C.]';
IDCY{12}   = 'CNIAR';DescrCY{12}='Cytoplasm to Nucleus Intensity Average Ratio';
IDCY{13}   = 'CNCAR';DescrCY{13}='Cytoplasm to Nucleus CMTMR Average Ratio';
IDCY{14}   = 'CGCO';DescrCY{14}='Cytoplasm Grayscale Centroid Offset [um]';
IDCY{15}   = 'CCO';DescrCY{15}='Cytoplasm Cell Centroids Offset [um]';
% cell interface metrics
NumMtIntrf = 18;
IDCM      = cell(1,NumMtIntrf);DescrCM    = cell(1,NumMtIntrf);
IDCM{1}    = 'CMPA';DescrCM{1} = 'Cell-Matrix Proximity Area [um2]';
IDCM{2}    = 'CMPAF';DescrCM{2} = 'Cell-Matrix Proximity Area Fraction';
IDCM{3}    = 'CMPAIA';DescrCM{3} = 'Cell-Matrix Proximity Alexa488 Intensity Average';
IDCM{4}    = 'CMPAIS';DescrCM{4} = 'Cell-Matrix Proximity Alexa488 Intensity Std';
IDCM{5}    = 'CMCCO';DescrCM{5} = 'Cell-Matrix Proximity Cell Centroids Offset ';
IDCM{6}    = 'CVPA';DescrCM{6} = 'Cell-Void Proximity Area [um2]';
IDCM{7}    = 'CVPAF';DescrCM{7} = 'Cell-Void Proximity Area Fraction';
IDCM{8}    = 'CVPAIA';DescrCM{8} = 'Cell-Void Proximity Alexa488 Intensity Average';
IDCM{9}    = 'CVPAIS';DescrCM{9} = 'Cell-Void Proximity Alexa488 Intensity Std';
IDCM{10}   = 'CVCCO';DescrCM{10}= 'Cell-Void Proximity Cell Centroids Offset ';
IDCM{11}   = 'CCPA';DescrCM{11}= 'Cell-Cell Proximity Area [um2]';
IDCM{12}   = 'CCPAF';DescrCM{12}= 'Cell-Cell Proximity Area Fraction';
IDCM{13}   = 'CCPAIA';DescrCM{13}= 'Cell-Cell Proximity Alexa488 Intensity Average';
IDCM{14}   = 'CCPAIS';DescrCM{14}= 'Cell-Cell Proximity Alexa488 Intensity Std';
IDCM{15}   = 'CCCCO';DescrCM{15}= 'Cell-Cell Proximity Cell Centroids Offset';
IDCM{16}   = 'CMPF';DescrCM{16}= 'Cell-Matrix Interface / Perimeter Fraction ';
IDCM{17}   = 'CVPF';DescrCM{17}= 'Cell-Void Interface / Perimeter Fraction ';
IDCM{18}   = 'CCPF';DescrCM{18}= 'Cell-Cell Interface / Perimeter Fraction ';
Metr = struct('ID',[IDNU,IDCY,IDCM],'Description',[DescrNU,DescrCY,DescrCM]);
CellMetrics = [];
for ctZZ = 1:length(ZPlanes)
    ctZ = ZPlanes(ctZZ);
    fprintf('\t\tPlane: %s: ',num2str(ctZ))

CellObjFileDir =
[Dat.CellScObj.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_CellScObj_z',Padding(ctZ,Dat.Nom.PadNum)];
    load(CellObjFileDir) % loads  CellsOBJ & ImPickedCells
    clear ImPickedCells
```

```
DecompFileDir = [Dat.ImAsbl.Dir,
Dat.Nom.DASH,Dat.Nom.fname,'_DomEvidAsbl_',Padding(ctZ,Dat.Nom.PadNum)];
    load(DecompFileDir)
    clear DomProbMap DomProbMapPROC ImSegm
    for ctD = 1:(1+Dat.ImDcmp.DomNum)
        [sx,sy,sz] = size(AsblCompConMap{ctD});
        AsblCompConMap{ctD} = reshape(AsblCompConMap{ctD},[sx*sy,sz]);
    end
    NumRow = CellsOBJ.ImageSize(1);     NumCol = CellsOBJ.ImageSize(2);
    CellMetrPlane = zeros(NumMtNuc + NumMtCyto + NumMtIntrf , CellsOBJ.NumObj);
    fprintf('Cell: ')
    for ctCL = 1:CellsOBJ.NumObj
        % === === === === === === === === === PRE-PROCESSING
        fprintf('%s ',num2str(ctCL))
        % === === === LOCAL COORDINATES
        PxCL = unique([CellsOBJ.Cell(ctCL).PxCyto;CellsOBJ.Cell(ctCL).PxNcl]);
        [RowCL,ColCL] = ind2sub([NumRow,NumCol],PxCL);
        minRowCL = min(RowCL);          maxRowCL = max(RowCL);
        minColCL = min(ColCL);          maxColCL = max(ColCL);
        NumRowCL1 = maxRowCL-minRowCL + 5;     NumColCL1 = maxColCL-minColCL + 5;
        % === ===  PROCESSING NUCLEUS DATA
        [RowNU,ColNU] = ind2sub([NumRow,NumCol],CellsOBJ.Cell(ctCL).PxNcl);
        RowNU1 = RowNU - minRowCL + 3;          ColNU1 = ColNU - minColCL + 3;
        PxNU1 = sub2ind([NumRowCL1,NumColCL1],RowNU1,ColNU1);
        ImNU = false(NumRowCL1,NumColCL1);          ImNU(PxNU1) = true;
        [RowNUp,ColNUp] = ind2sub([NumRowCL1,NumColCL1],find(bwperim(ImNU)));
        EliNU = FitEllipse(RowNUp,ColNUp);
        NucInt = AsblCompConMap{NuclDomain}(CellsOBJ.Cell(ctCL).PxNcl,:);
        NucBWCntr = [mean(RowNU1);mean(ColNU1)];
   NucINCntr = sum([RowNU1.*sum(NucInt,2),ColNU1.*sum(NucInt,2)])'/sum(sum(NucInt));
        % === === PROCESSING CYTOPLASM DATA
        [RowCY,ColCY] = ind2sub([NumRow,NumCol],CellsOBJ.Cell(ctCL).PxCyto);
        RowCY1 = RowCY - minRowCL + 3          ColCY1 = ColCY - minColCL + 3;
        PxCY1 = sub2ind([NumRowCL1,NumColCL1],RowCY1,ColCY1);
        CYInt = AsblCompConMap{CytoDomain}(CellsOBJ.Cell(ctCL).PxCyto,:);
        CYBWCntr = [mean(RowCY1);mean(ColCY1)];
        CYINCntr = sum([RowCY1.*sum(CYInt,2),ColCY1.*sum(CYInt,2)])'/sum(sum(CYInt));
        RowCL1 = RowCL - minRowCL + 3;          ColCL1 = ColCL - minColCL + 3;
        PxCL1 = sub2ind([NumRowCL1,NumColCL1],RowCL1,ColCL1);
        ImCL = false(NumRowCL1,NumColCL1)          ImCL(PxCL1) = true;
        [RowCEp,ColCEp] = ind2sub([NumRowCL1,NumColCL1],find(bwperim(ImCL)));
        EliCE = FitEllipse(RowCEp,ColCEp);
        partNU = AsblCompConMap{NuclDomain}(CellsOBJ.Cell(ctCL).PxNcl,:);
        partNU = [partNU,zeros(size(partNU,1),1)];
        partCY = AsblCompConMap{CytoDomain}(CellsOBJ.Cell(ctCL).PxCyto,:);
        partCY = [zeros(size(partCY,1),1),partCY];
        CLInt  = [partNU;partCY];
        CLBWCntr = [mean(RowCL1);mean(ColCL1)];
        clear CLInt partCY partNU
        % === ===  PROCESSING CELL-MATRIX BOUNDARY DATA
        [RowCM,ColCM] = ind2sub([NumRow,NumCol],CellsOBJ.Cell(ctCL).PxMatrixBnd);
        RowCM1 = RowCM - minRowCL + 3;          ColCM1 = ColCM - minColCL + 3;
        CMInt = AsblCompConMap{CytoDomain}(CellsOBJ.Cell(ctCL).PxMatrixBnd,:);
        CMBWCntr = [mean(RowCM1);mean(ColCM1)];
        clear RowCM ColCM RowCM1 ColCM1
        % === ===  PROCESSING CELL-VOID BOUNDARY DATA
        [RowCV,ColCV] = ind2sub([NumRow,NumCol],CellsOBJ.Cell(ctCL).PxVoidBnd);
        RowCV1 = RowCV - minRowCL + 3;          ColCV1 = ColCV - minColCL + 3;
        CVInt = AsblCompConMap{CytoDomain}(CellsOBJ.Cell(ctCL).PxVoidBnd,:);
        CVBWCntr = [mean(RowCV1);mean(ColCV1)];
        clear RowCV ColCV RowCV1 ColCV1
        % === ===  PROCESSING CELL-CELL BOUNDARY DATA
      [RowCC,ColCC] = ind2sub([NumRow,NumCol],CellsOBJ.Cell(ctCL).PxCellBnd);
        RowCC1 = RowCC - minRowCL + 3;          ColCC1 = ColCC - minColCL + 3;
        CCInt = AsblCompConMap{CytoDomain}(CellsOBJ.Cell(ctCL).PxCellBnd,:);
        CCBWCntr = [mean(RowCC1);mean(ColCC1)];
        clear RowCC ColCC RowCC1 ColCC1
        %% === === === === === === === === === CALCULATE METRICS
        % === === NUCLEUS METRICS
        NUMetrics = zeros(NumMtNuc,1);
        % NA: Nucleus Area
```

```
NUMetrics(1) = length(CellsOBJ.Cell(ctCL).PxNcl)*(PxSzA)^2;
% NP: Nucleus Perimeter
NUMetrics(2) = sum(sum(bwperim(ImNU)))*PxSzA;
% NSF: Nucleus Shape Factor
NUMetrics(3) = 4*pi*NUMetrics(1)/(NUMetrics(2))^2;
% NE: Nucleus Eccentricity
NUMetrics(4) = EliNU(1)/EliNU(2);
% NTIA: Nucleus Total Intensity Average
NUMetrics(5) = mean(sum(NucInt,2));
% NTIS: Nucleus Total Intensity Std
NUMetrics(6) = std(sum(NucInt,2));
% NHIA: Nucleus Hoechst Intensity Average
NUMetrics(7) = mean(NucInt(:,1));
% NHIS: Nucleus Hoechst Intensity Std
NUMetrics(8) = std(NucInt(:,1));
% NCIA: Nucleus CMTMR Intensity Average
NUMetrics(9) = mean(NucInt(:,2));
% NCIS: Nucleus CMTMR Intensity Std
NUMetrics(10) = std(NucInt(:,2));
% NGCO: Nucleus Grayscale Centroid Offset [um]
NUMetrics(11) = norm((NucBWCntr - NucINCntr))*PxSzA;
% NCO: Nucleus Cell Centroids Offset [um]
NUMetrics(12) = norm((NucBWCntr - CLBWCntr))*PxSzA;
clear ImNU NucInt NucBWCntr NucINCntr
% === === CYTOPLASM METRICS
CYMetrics = zeros(NumMtCyto,1);
% CA: Cytoplasm Area
CYMetrics(1) = length(CellsOBJ.Cell(ctCL).PxCyto)*(PxSzA)^2;
% CP: Cell Perimeter
CYMetrics(2) = sum(sum(bwperim(ImCL)))*PxSzA;
% CSF: Cell Shape Factor
CYMetrics(3) = 4*pi*(NUMetrics(1) + CYMetrics(1))/(CYMetrics(2))^2;
% CE: Cell Eccentricity
CYMetrics(4) = EliCE(1)/EliCE(2);
% CPNPR: Cell  Perimeter to Nucleus Perimeter Ratio
CYMetrics(5) = CYMetrics(2)/NUMetrics(2);
% CTIA: Cytoplasm Total Intensity Average
CYMetrics(6) = mean(sum(CYInt,2));
% CTIS: Cytoplasm Total Intensity Std
CYMetrics(7) = std(sum(CYInt,2));
% CAIA: Cytoplasm Alexa488 Intensity Average
CYMetrics(8) = mean(CYInt(:,2));
% CAIS: Cytoplasm Alexa488 Intensity Std
CYMetrics(9) = std(CYInt(:,2));
% CCIA: Cytoplasm CMTMR Intensity Average
CYMetrics(10) = mean(CYInt(:,1));
% CCIS: Cytoplasm CMTMR Intensity Std
CYMetrics(11) = std(CYInt(:,1));
% CNIAR: Cytoplasm to Nucleus Intensity Average Ratio
CYMetrics(12) = CYMetrics(6)/NUMetrics(5);
% CNCAR: Cytoplasm to Nucleus CMTMR Average Ratio
CYMetrics(13) = CYMetrics(10)/NUMetrics(9);
% CGCO: Cytoplasm Grayscale Centroid Offset [um]
CYMetrics(14) = norm((CYBWCntr - CYINCntr))*PxSzA;
% CCO: Cytoplasm Cell Centroids Offset [um]
CYMetrics(15) = norm((CYBWCntr - CLBWCntr))*PxSzA;
clear CYInt CYBWCntr CYINCntr ImCL
% === === CELL INTERFACE METRICS
CIMetrics = zeros(NumMtIntrf,1);
% CMPA: Cell-Matrix Proximity Area [um2]
CIMetrics(1) = length(CellsOBJ.Cell(ctCL).PxMatrixBnd)*(PxSzA)^2;
% CMPAF: Cell-Matrix Proximity Area Fraction
CIMetrics(2) = CIMetrics(1)/(CYMetrics(1) + NUMetrics(1));
% CMPAIA: Cell-Matrix Proximity Alexa488 Intensity Average
CIMetrics(3) = mean(CMInt(:,2));
% CMPAIS: Cell-Matrix Proximity Alexa488 Intensity Std
CIMetrics(4) = std(CMInt(:,2));
% CMCCO: Cell-Matrix Proximity Cell Centroids Offset
CIMetrics(5) = norm((CMBWCntr - CLBWCntr))*PxSzA;
% CVPA: Cell-Void Proximity Area [um2]
CIMetrics(6) = length(CellsOBJ.Cell(ctCL).PxVoidBnd)*(PxSzA)^2;
```

```
        % CVPAF: Cell-Void Proximity Area Fraction
        CIMetrics(7) = CIMetrics(6)/(CYMetrics(1) + NUMetrics(1));
        % CVPAIA: Cell-Void Proximity Alexa488 Intensity Average
        CIMetrics(8) = mean(CVInt(:,2));
        % CVPAIS: Cell-Void Proximity Alexa488 Intensity Std
        CIMetrics(9) = std(CVInt(:,2));
        % CVCCO: Cell-Void Proximity Cell Centroids Offset
        CIMetrics(10) = norm((CVBWCntr - CLBWCntr))*PxSzA;
        % CCPA: Cell-Cell Proximity Area [um2]
        CIMetrics(11) = length(CellsOBJ.Cell(ctCL).PxCellBnd)*(PxSzA)^2;
        if CIMetrics(11)>0
            % CCPAF: Cell-Cell Proximity Area Fraction
            CIMetrics(12) = CIMetrics(11)/(CYMetrics(1) + NUMetrics(1));
            % CCPAIA: Cell-Cell Proximity Alexa488 Intensity Average
            CIMetrics(13) = mean(CCInt(:,2));
            % CCPAIS: Cell-Cell Proximity Alexa488 Intensity Std
            CIMetrics(14) = std(CCInt(:,2));
            % CCCCO: Cell-Cell Proximity Cell Centroids Offset
            CIMetrics(15) = norm((CCBWCntr - CLBWCntr))*PxSzA;
        else
            % CCPAF: Cell-Cell Proximity Area Fraction
            CIMetrics(12) = 0;
            % CCPAIA: Cell-Cell Proximity Alexa488 Intensity Average
            CIMetrics(13) = 0;
            % CCPAIS: Cell-Cell Proximity Alexa488 Intensity Std
            CIMetrics(14) = 0;
            % CCCCO: Cell-Cell Proximity Cell Centroids Offset
            CIMetrics(15) = 0;
        end
        % CMPF: Cell-Matrix Interface / Perimeter Fraction
        CIMetrics(16) = CIMetrics(1)/(CIMetrics(1) + CIMetrics(6) + CIMetrics(11));
        % CVPF: Cell-Void Interface / Perimeter Fraction
        CIMetrics(17) = CIMetrics(6)/(CIMetrics(1) + CIMetrics(6) + CIMetrics(11));
        % CCPF: Cell-Cell Interface / Perimeter Fraction
        CIMetrics(18) = CIMetrics(11)/(CIMetrics(1) + CIMetrics(6) + CIMetrics(11));
        clear CLBWCntr CMInt CVInt CCBWCntr
        % === === assemble
        CellMetrPlane(:,ctCL) = [NUMetrics ; CYMetrics ; CIMetrics];
    end
    fprintf('\n')
    CellMetrics = [CellMetrics,CellMetrPlane];
end

if Dat.CellScObj.SaveSw
SaveFileDir = [Dat.CellScObj.Dir,Dat.Nom.DASH,Dat.Nom.fname,'_CellMetrics'];
    save(SaveFileDir,'CellMetrics','Metr')
end
fprintf('\t\t..... done\n\n')
return

function ELI = FitEllipse(x,y)
% fitting data into ellipsoid using the algorithm described in the paper
% 'NUMERICALLY STABLE DIRECT LEAST SQUARES FITTING OF ELLIPSES' by Halir
% and Flusser, 1998
% ---------------------------
% INPUTS
% x,y: vectors that contain data
%
% OUTPUTS
% ELI = [Rx;Ry;fi;ro(:)];
% Rx: long axis radius
% Ry: small axis radius
% fi: angle
% ro: offset position
% ---------------------------
% Dimitrios Tzeranis Sept 2012
% ============ Step 1: calculate a polynomial a
D1 = [x.^2, x.*y, y.^2]; % quadratic part of the design matrix
D2 = [x, y, ones(size(x))]; % linear part of the design matrix
T = -(D2'*D2)\(D1'*D2)'; % for getting a2 from a1
M = D1'*D1 + (D1'*D2)*T; % reduced scatter matrix
```

```
M = [M(3,:)./2; -M(2,:); M(1,:)./2]; % premultiply by inv(C1)
[evec, eval] = eig(M); % solve eigensystem
cond = 4*evec(1, :).*evec(3,:) - evec(2,:).^2; % evaluate aï¿½Ca
a1 = evec(:, find(cond > 0)); % eigenvector for min. pos. eigenvalue
a = [a1; T*a1]; % ellipse coefficients
if a(1)<0;  a=-a; end
% ============ Step 2: calculate parameters
fi = atan(a(2)/(a(1)-a(3)))/2;
ro = -0.5*([a(1),a(2)/2;a(2)/2,a(3)]\[a(4);a(5)]);
Lx = ((a(1)-a(3))+(a(1)+a(3))*cos(2*fi))/(2*cos(2*fi));
Ly = (-(a(1)-a(3))+(a(1)+a(3))*cos(2*fi))/(2*cos(2*fi));
Re = [cos(fi),-sin(fi);sin(fi),cos(fi)];
G = ro'*Re*diag([Lx,Ly])*Re'*ro-a(6);
if Lx>Ly
    temp = [Lx,Ly];    Lx = temp(2);
    Ly = temp(1);    fi = fi + pi/2;
end
if fi>pi/2
    fi = fi - pi;
elseif fi<-pi/2
    fi = fi + pi;
end
Rx = sqrt(G/Lx);
Ry = sqrt(G/Ly);
ELI = [Rx;Ry;fi;ro(:)];
return
```

Literature Cited (Appendices F to L)

GST Gene Fusion System, Amersham Biosciences

Wizard Plus SV Minipreps DNA Purification System, Technical Bulletin, Promega, 2007

Harley BAC. *Cell-Matrix Interactions : Collagen-GAG Scaffold Fabrication, Characterization, and Measurement of Cell Migratory and Contractile Behavior via Confocal Microscopy*, PhD Thesis, Department of Mechanical Engineering, MIT, 2006.

Olde Damink LHH, Dijkstra PJ, van Luyn MJA, Van Wachem PB, Nieuwenhuis P and Feijen J. Cross-linking of dermal sheep collagen using a water soluble carbodiimide. *Biomaterials* **17**: 765-773, 1996.

Pace C. N., et al. 1995, How to measure and predict the molar absorption coefficient of a protein, *Prot. Sci.* **4**: 2411-2423

Wilson K., and Walker J. 2000, Principles and techniques of practical biochemistry, fifth edition, Cambridge university press.

Current Protocols in Molecular Biology, John Willey and Sons, 2007.

Protein assay compatibility table, Tech Tip #68, Pierce Thermoscientific, 2012.