Integrative approaches for systematic reconstruction of regulatory circuits in mammals

by

Ana Paula Santos Botelho Oliveira Leite

Licenciatura in Mathematics, Universidade do Minho (2002)
Mestrado in Mathematical Engineering, Universidade do Porto (2006)

Submitted to the Computational and Systems Biology Initiative
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computational and Systems Biology

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2012

Author ...
Computational and Systems Biology Initiative
August 30, 2012

Certified by ...
Aviv Regev, Ph.D.
Associate Professor of Biology
Thesis Supervisor

Accepted by ...
Christopher B. Burge, Ph.D.
Associate Professor of Biology and Biological Engineering
Director, Computational and Systems Biology Ph.D. Program

João Botelho Oliveira Leite

1932-2011

In loving memory of my father

Integrative approaches for systematic reconstruction of regulatory circuits in mammals

by

Ana Paula Santos Botelho Oliveira Leite

Submitted to the Computational and Systems Biology Initiative
on August 30, 2012, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computational and Systems Biology

Abstract

The reconstruction of regulatory networks is one of the most challenging tasks in systems biology. Although some models for inferring regulatory networks can make useful predictions about the wiring and mechanisms of molecular interactions, these approaches are still limited and there is a strong need to develop increasingly universal and accurate approaches for network reconstruction. This problem is particularly challenging in mammals, due to the higher complexity of mammalian regulatory networks and limitations in experimental manipulation. In this thesis, I present three systematic approachs to reconstruct, analyse and refine models of gene regulation. In Chapter 1, I devise a method for deriving an observational model from temporal genomic profiles. I use it to choose targets for perturbation experiments in order to determine a network controlling the responses of mouse primary dendritic cells to stimulation with pathogen components. In Chapter 2, I introduce the algorithm Exigo, for identifying essential interactions in regulatory networks reconstructed from experimental data where regulators have been silenced, using a network reduction strategy. Exigo outperforms previous approaches on simulated data, uncovers the core network structure when applied to real networks derived from perturbation studies in mammals, and improves the performance of network inference methods. Lastly, I introduce in Chapter 3 an approach to learn a module network from multiple high-throughput assays. Analysis of a diffuse large B-cell lymphoma dataset identifies candidate regulator genes, microRNAs and copy number aberrations with biological, and possibly therapeutic, importance.

Thesis Supervisor: Aviv Regev, Ph.D.
Title: Associate Professor of Biology

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Aviv Regev, for welcoming me into her lab and sharing with me her experience, knowledge and insights (Aviv, I hope to make you proud in my future research endeavors). Second, I'd like to express tremendous gratitude to my thesis committee, Professors Bonnie Berger and Ernest Fraenkel, for sharing with me their time and knowledge, listening to my ideas, and pointing me in helpful directions. I am also thankful to Professor Chris Burge for his guidance and support.

This journey started at Instituto Gulbenkian de Ciência, in the context of its pioneering PhD Program in Computational Biology. In this regard, I am thankful to Marie-France Sagot and Jorge Carneiro for their mentorship. I must also thank Fundacão para a Ciência e a Tecnologia for funding my work.

I want to thank all past and present members of the Regev lab for the good working atmosphere and the scientific - and sometimes not so scientific - discussions. In particular, I want to thank Dawn, Ilan and Noa, for being so helpful when I joined the lab; Sushmita and Alon for sharing with me their knowledge and experience; Kendra for all her dedication; and Jay, Jenna, Courtney, Michelle, Jason and Nathalie for all the joyful time together. The work in this thesis would not have been possible without collaborations with Ido, Max and Stefano, who taught me how science can be fun.

On a personal note, I want to acknowledge my Portuguese friends. Special thanks go to Joana, Cláudia, Paulinha, Ana Sofia, Lígia, Garfo, Truta, Ana Rita, Ana Luísa, Ísis, Sandra, Sílvia, Victor, Paula and Zé, but there are many more. Your friendship is truly a blessing. Also, I have no words to describe how thankful I am to Elsa, Francisco and David. I will always remember the moments we have shared together.

I have saved the most important acknowledgements for last. I owe everything that I am to my parents, Ana Maria and João. Their support has been unsurpassable and unflagging, their love unbounded and unconditional. Finally, my husband Rogério has simply been the greatest person I've ever met.

Contents

Chapter 1

Introduction

Chapter 1

Introduction

1.1 Overview

The reconstruction of regulatory networks is one of the most challenging tasks in systems biology. Although some models for inferring regulatory networks can make useful predictions about the wiring and mechanisms of molecular interactions, these approaches are still limited and there is a strong need to develop increasingly universal and accurate approaches for network reconstruction. This problem is particularly challenging in mammals, due to the higher complexity of mammalian regulatory networks and limitations in experimental manipulation.

In this thesis, I present a broadly-applicable approach to reconstruct, analyze and refine models of gene regulation. In Chapter 1, I introduce existing approaches and their limitations. In Chapter 2, I devise a method for deriving an observational model from genome-wide temporal expression profiles and use it to choose targets for perturbation experiments, in a study of the network controlling the response of Dendritic Cells (DCs) to stimulation with pathogen components. In Chapter 3, I develop an algorithm for distinguishing relevant from irrelevant interactions in the perturbation network and to refine the initial model. In Chapter 4, I reconstruct and analyse an integrative regulatory model of Diffuse Large B-Cell Lymphoma (DLBCL) and use it to suggest perturbation experiments. I conclude this thesis in Chapter 5 with a summary of the chapters and a discussion of future work.

1.2 Regulatory network reconstruction

Regulatory networks are the information processing devices of cells, transforming cellular signals into coherent transcriptional responses. However, even for well characterized systems we do not fully understand how a regulatory network processes signals, encodes the relevant information at different layers of the network, and achieves the fine-tuned change in expression in each target gene. In particular, reconstruction of regulatory networks in mammalian cells has been a challenge [1], given the system complexity and the obstacle of performing targeted genetic perturbations on a large scale in a mammalian animal model.

There are three main approaches to reconstruct regulatory networks: (1) observational models look for statistical associations between regulatory elements (in promoter sequences [2]) or factors (in mRNA expression [3]) and target gene expression; (2) perturbational models examine the effect of genetic perturbation (deletion [4], overexpression [5], knockdown [6], or natural genetic variation [7]) on gene expression; and (3) physical models examine direct binding between regulatory proteins and promoters or enhancers. In many cases, heterogeneous models attempt to integrate one or more of these approaches.

Inferring regulation from observational models

Two major 'observational' strategies have been used to associate regulators with putative targets on a genome scale [8]: (1) *cis*-regulatory models, which consider the presence of predicted transcription factor (TF) binding sites in the promoters of target genes [1, 8]; and (2) *trans*-regulatory models, that infer interactions based on correlations between regulator and target expression [1, 3, 8]. The latter approach relies on the fact that many *trans*-regulators are embedded within transcriptional feedback and feedforward regulatory loops [3, 9], and hence their own expression may be transcriptionally regulated within processes that they control. These two types of models have some inherent limitations. In particular, *cis*-regulatory models cannot predict expression because they are not condition specific, and *trans*-regulatory

models only account for changes in mRNA levels, not activity. It is, though, still challenging to estimate the success of most of the models. Two major reasons for this are the lack of systematic experimental follow-up and the limited scale of most datasets. It is thus important to establish standard methodologies and data sets on which the performance of different models can be compared.

Module-based algorithms

Many of the observational models developed rely on the assumption that gene modules (rather than individual genes) are the units of gene regulation. The rationale for this grouping is based on several examples in which the same regulatory circuits coordinate activation or repression of a regulon of genes that are involved in the same process (e.g., all ribosomal protein genes are regulated by common transcription factors). Module-based models have been extensively used in genomics ([3, 10]) because they achieve greater statistical power by using fewer parameters.

Module Networks algorithm

Module Networks [3] is a procedure based on probabilistic graphical models [11]. It takes as input a data set of gene expression profiles and a large precompiled set of candidate regulator genes, and determines both the partition of genes to modules and the regulatory programs, that explain the behavior of the genes in the modules as a function of the expression level of sets of regulators (Fig. 1-1).

In Module Networks, the action of regulators on each module is described by a logical program represented as a decision tree . Each node in the tree consists of a regulatory gene and each leaf of a regulation context, that is, a configuration of the regulator genes, determined by the path from the root to the leaf. A regulation context determines the gene expression probabilistically. For example, in Fig. 1-1, when regulator gene A is downregulated and regulator gene B is upregulated, the module genes are repressed.

A module network is learned by both partitioning the genes into modules and learning the Bayesian network between the module nodes. This is accomplished

using an Expectation-Maximization (EM) algorithm. In the M-step, it learns the best decision tree for each module, given the partition of genes into modules. In other words, given the module assignments, the algorithm builds the regulatory program for each module by choosing the regulator whose split best predicts the behavior of the module genes, creating a node on the decision tree for this regulator and then recursing on the two branches. In the E-step, the gene pool is partitioned into modules, given the regulatory programs. Basically, for each gene in a module, we can obtain a probability value that a gene expression value was obtained by a particular regulatory program. In this case, the algorithm simply assigns the gene to the module that best predicts it. Each reassignment therefore is guaranteed to improve the overall predictiveness.

The Module Networks algorithm has been applied to yeast [3] and mammals [12], showing its ability to make reasonable predictions and to provide a deeper understanding of the functionality of a regulatory network. By pooling many similar genes together, the module network framework significantly increases the statistical power to identify regulatory influences. For that reason, it has inspired the development of other algorithms. For example, Segal *et al.* [13] present a probabilistic model over both gene expression and sequence data to identify transcriptional modules and the regulatory motif binding sites that control their regulation within a given set of experiments; Geronemo [14] automatically constructs a set of co-regulated genes whose regulation can involve both sequence variations and expression of regulators; LIRNET [15] solves a SNP-eQTL module network involving sequence variations, expression of regulators and prior knowledge; and CONEXIC [16] integrates matched copy number (amplifications and deletions) and gene expression data from tumor samples to identify driving mutations and the processes they influence.

Linear regression models

Linear regression models have provided a robust alternative to module-based algorithms as observational models [17]. In a multiple linear regression, the objective is to find weights or regression coefficients for the independent variables such that the

linear combination explains the variance of the dependent variable as well as possible. The weights describe the relative importance of each independent variable.

Linear regression models have been applied to learn both *cis*-regulatory models [18], by modelling gene expression as a function of one or more known or potential transcription factor binding site variables based on the gene's regulatory region (e.g. motif counts, position weight matrix (PWM) scores, ChIP-chip quantities), or *trans*-regulatory models of gene regulation [19], by modelling target gene expression as a combination of TFs expression levels.

High dimensionality of microarray data can lead to models that are very complex and pose challenges in prediction and interpretation. For this reason, penalized regression methods have received much attention over the past few years, as a proper way to get sparse models. The use of penalties facilitate fitting models with predictors that run into thousands, including many irrelevant to the response, far exceed the sample size, or are highly correlated. Three regularized linear regression models often used in genomic studies (e.g. [15]) are ridge regression [20], least absolute shrinkage and selection operator (LASSO) [21], and elastic net [22]. The ridge regression uses a L_2 penalty and is best used when there are high correlations between predictors. However, it could be influenced by irrelevant variables since it uses all the predictors in hand. The LASSO uses the L_1 penalty and does both continuous shrinkage and automatic variable selection simultaneously. However, in the presence of multicollinearity, it has empirically been observed that the prediction performance of the LASSO is dominated by ridge regression [21]. Elastic net attempts to keep the advantages of both ridge and LASSO, overcoming their limitations by combining the L_1 and L_2 penalties. In addition, it has a grouping effect, i.e., if there is a set of variables among which the pairwise correlations are high, the elastic net groups the correlated variables together.

Reconstructing models from temporal data

Biological systems are predominantly dynamic [23] and analysis of temporal gene expression profiles can provide valuable insights about a system. For this reason,

systems' dynamics over a period of time have been investigated in computational modeling of cellular processes [24, 25].

Different methods have been extensively explored to infer causal relationships from time series gene expression data, such as ordinary differential equations [26], dynamic bayesian networks [27], Granger causality [28], Hidden Markov Models [29] and mutual information approaches [30]. However, these expression-based inference methods have had only limited success for several reasons: (1) the insufficient time resolution of the available samples often limit our ability to distinguish signal from noise and results in fast responses being missed; (2) it is hard to distinguish between regulators that actually regulate a gene (i.e., that have a direct causal effect) and regulators that are merely co-expressed with a gene; and (3) some methods lack a systematic way to determine a biologically relevant transcriptional time lag, which may be due to the combined effects of the translation, folding, nuclear translocation and turnover time-scales for the regulatory protein and for elongation of the target gene mRNA. Not modeling the time that it takes for the regulator gene to express its protein product and the transcription of the target gene to be affected by this regulator protein prevents from capturing expression dynamics and may result in low accuracy.

Perturbational models can uncover cellular wiring diagrams

Perturbational models associate targets to factors based on the effect of the factors' genetic manipulation on gene expression. Recent advances in genomics technology, such as mRNA profiling and the development of genome-scale lentiviral RNA interference (RNAi) for efficient perturbation, make it possible to reconstruct complex circuits. For example, Baym *et al.* [31] present a technique to infer the Rho-signaling network in *Drosophila* from microarray data on perturbation experiments.

Systematic perturbation experiments can be important for validating the regulatory interactions predicted by an observational (non-perturbational) model. Such experiments have been successfully employed in modeling regulatory networks in sea urchin [32] and yeast [33], calling for new computational strategies. In particular,

effective ways to distinguish the indirect effects of a perturbation from the direct effects, since many of the network interactions, identified by gene knockdown instead of by direct measurements of transcription factor-promoter binding, are likely to be indirect.

A starting point to distinguish direct from indirect effects has been proposed by Wagner [34–36]. This approach uses gene perturbation experiments to build a directed graph (*perturbation graph*), where an edge is introduced from gene A to gene B if perturbing A results in a significant expression change in B. It then uses graph-theoretic methods of transitive reduction [37] to reconstruct the minimal (most parsimonious) subgraph. However, this approach presents several limitations, such as radical pruning and the pre-requisite that the perturbation graph is acyclic. To overcome these issues, several other strategies have followed [38–40], but they are generally computationally expensive and fail to take into account the effect of indirect interactions while globally explaining the experimental data. These approaches are extensively discussed in Chapter 3.

Accurate computational methods to analyze perturbation graphs are essential in a time when RNAi screening is becoming part of the standard experimental repertoire. In the future, RNAi screens will only be the first step in the comprehensive analysis of biological phenomena. Iteratively integrating experimentation and computation is a promising approach to refine our understanding of the inner working of the cell.

Physical models examine direct binding

Physical models associate regulatory factors with the targets whose promoters they bind [41, 42]. Indeed, chromatin immunoprecipitation (ChIP) has revolutionized the study of transcriptional networks: maps of TF-DNA binding can be coupled with measurements of expression output to build models of regulatory circuits [8]. Even though binding information should be interpreted with caution, as the presence of a regulator at a promoter region binding does not necessarily mean functional regulation, measurement of TF binding before and after a stimulus can help distinguish between direct and indirect targets [33, 42, 43].

Measurements of *trans* inputs are still limited by the effort and cost in generating the needed reagents and data, such as the development of an epitope-tagged version of every TF and the large amounts of material required in large-scale screens. Automated methods for systematic mapping TFs, that increase the throughput and sensitivity, while reducing the labor and cost for ChIP experiments, can assist in further understanding mammalian regulatory circuits across a range of cell states, conditions and perturbations.

1.3 Heterogeneous models capture regulatory circuitry at several levels

Biological systems employ heterogeneous regulatory mechanisms that are frequently intertwined. Recent technological advances have made available several types of genomic and proteomic data, some already mentioned above. Among them are miRNA and gene expression, copy number aberrations (CNAs), single nucleotide polymorphisms (SNPs), and protein-protein/gene-gene interactions. Each of these data types provides a different, partly independent and complementary, view of the whole circuit. However, understanding functions of genes, proteins, and other components of the genome requires more information than provided by each of the datasets. For example, Huang *et al.* [44] derive a network of protein-protein and protein-DNA interactions that explains the functional context of genes and proteins detected in these assays and uncovers diverse pathways not obvious from the input.

Integrating data from different sources can bring many challenges. Genomic data arise in the form of vectors, graphs, or sequences, and it is important to carefully consider strategies that best capture the most information contained in each data. Moreover, data from different sources might have different quality and informativity. For example, probe design and experimental conditions are known to influence signal intensities and sensitivities for many high-throughput technologies.

Data integration in clinical setting

The need for integration of heterogeneous data measured on the same individuals arises in a wide range of clinical applications. In this regard, the best example is perhaps the challenge that cancer researchers and clinicians face in the diagnosis, treatment, and prognosis of this complex disease. Cancer is thought to be primarily caused by genetic alterations and, as such, genomic data like gene expression and CNAs can be used together to develop models that learn novel regulation functions [45]. The critical challenge is in differentiating between alterations that drive the cancer growth and other seemingly random alterations that accumulate through instability induced by tumorigenesis. For example, the CONEXIC algorithm [16] combines DNA copy number with gene expression levels to identify driver mutations and predict the processes that they alter. The model assumes that a driver mutation should co-vary with a gene module involved in tumorigenesis (i.e., it assumes that the modules expression is "modulated" by the driver), and that expression levels of the driver control the malignant phenotype rather than copy number (because other mechanisms may lead to similar dysregulated expression of the driver gene). This approach predicted two new tumor dependencies in melanoma and the processes that they alter. Other studies tracked the causes of abnormal gene expressions by correlating them with gene mutations, DNA methylations or microRNA expressions (e.g., [46, 47]). However, integrative studies are often restricted to pairwise comparisons between two types of data and lack a unifying framework to integrate multiple types of data in the same model.

Figure

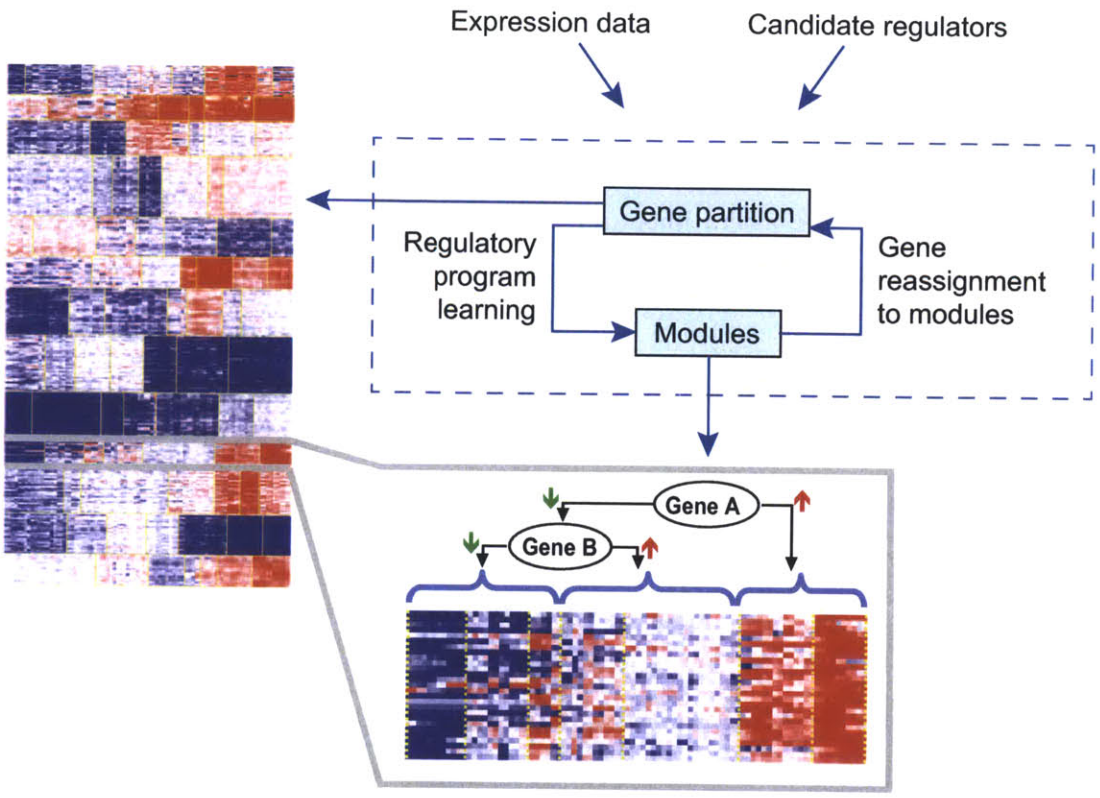

Figure 1-1: Overview of the module networks algorithm. The procedure takes as input a data set of gene expression profiles and a pre-compiled set of candidate regulator genes. The method itself (dashed box) is an iterative procedure that determines both the partition of genes to modules and the regulatory program (below the dashed box) for each module. Red/Blue - high/low expression. Figure adapted from [3].

Chapter 2

Unbiased Reconstruction of a Mammalian Transcriptional Network Mediating Pathogen Responses

Part of the work included in this chapter was first published as:

Ido Amit, Manuel Garber*, Nicolas Chevrier*, **Ana Paula Leite***, Yoni Donner*, Thomas Eisenhaure, Mitchell Guttman, Jennifer K. Grenier, Weibo Li, Or Zuk, Lisa A. Schubert, Brian Birditt, Tal Shay, Alon Goren, Xiaolan Zhang, Zachary Smith, Raquel Deering, Rebecca C. McDonald, Moran Cabili, Bradley E. Bernstein, John L. Rinn, Alex Meissner, David E. Root, Nir Hacohen, and Aviv Regev. Unbiased Reconstruction of a Mammalian Transcriptional Network Mediating Pathogen Responses. *Science*, 2009;**326**(5950):257-63. PMID: 19729616. **These authors contributed equally to this work.*

Experiments in this chapter were performed by Ido Amit and Nicolas Chevrier in the Regev and Hacohen laboratories. Yoni Donner developed GeneSelector and Manuel Garber colaborated in nCounter data analysis.

Chapter 2

Reconstruction of a Mammalian Transcriptional Network Mediating Pathogen Responses

2.1 Abstract

Deciphering the regulatory networks that control dynamic and specific gene expression responses in mammalian cells remains a major challenge. While models inferred from genomic data have identified candidate regulatory mechanisms, such models remain largely unvalidated. Here, we built on the framework of graphical models and Elastic-Net regression to learn an observational model of gene regulation from temporal expression profiles. We used it to identify candidate regulators that act on target genes in the transcriptional response to pathogens in primary Dendritic Cells (DCs). We then tested the regulatory function of 125 of the candidate regulators (consisting of transcription factors, chromatin modifiers, and RNA binding proteins) by systematic perturbions with shRNAs, followed by measurement of a gene expression signature during response to the same pathogen components. This approach accurately assigned 32 known regulators (e.g. NFB, IRFs, and STATs) to their target genes and discovered 68 additional functional regulators that were not previously

implicated in this response. This unique dataset provided a valuable resource to estimate the quality of the observational model. We observed that it identified many correct regulators but revealed numerous false positive relations, confirming that inferring regulatory activity from expression levels alone limits the model's ability to distinguish between causality and correlation. Generally, this study establishes a broadly-applicable, comprehensive and unbiased approach to identifying the wiring and function of a regulatory network controlling a major transcriptional response in primary mammalian cells.

2.2 Background

Regulatory networks controlling gene expression serve as decision-making circuits within cells. For example, when immune dendritic cells (DCs) are exposed to viruses, bacteria, or fungi, they respond with transcriptional programs that are specific to each pathogen [48] and are essential for establishing appropriate immunological outcomes [49]. These fast and dynamic responses are initiated through specific receptors, such as Toll-like receptors (TLRs), that distinguish broad pathogen classes and are propagated through well-characterized signaling cascades [49]. However, little is known about how the transcriptional network is wired to produce specific outputs.

Systematic unbiased reconstruction of regulatory networks in any mammalian cell remains a fundamental challenge. Two major strategies have been previously used to associate regulators with their putative targets on a genome scale [8]: *cis*-regulatory models rely on the presence of predicted binding sites for a factor in the promoters of target genes [1, 8, 50], whereas *trans*-regulatory models rely on the dependence between the activity of expression of a regulator to that of its targets across samples [1, 3, 8, 50]. In innate immunity, such *observational models* have led to the successful identification of a few key regulators Atf3 [51], Tgif [1], and CEBP/δ [52], but have failed to provide a comprehensive model that explains the complexity and specificity of the response. This is due to the limitations inherent in each approach: *cis*-regulatory models are biased to regulators with known binding sites and explain little of the

observed expression [8], whereas *trans*-models are under-determined to distinguish correlation from causation [3, 8].

A complementary strategy is to infer a *perturbational model*, by systematically perturbing every regulatory input and measuring its effect on the transcriptional output of target genes. This strategy has been successfully employed in yeast [4, 33, 43] and sea urchin [32], but not in mammals, due to the lack of efficient genetic tools and the prohibitive cost of large-scale profiling studies. Recent technological advances in gene perturbation and multiplex detection [8] can mitigate these limitations.

Here, we combined an observational model and perturbations to develop the first unbiased perturbational model of a regulatory network in a mammalian cell (Fig. 2-1). We first used transcriptional profiles and an observational model to select candidate regulators (transcription factors, chromatin modifiers, and RNA binding proteins) for a perturbation screen, working with the model system of primary DCs responding to pathogens. Next we used a lentiviral shRNA library [53] to silence 125 of the candidates in primary DCs. We then used an innovative and accurate multiplex detection technology to monitor a representative signature of target gene expression following each perturbation. The result was a perturbational model of cause-and-effect relationships in the regulatory circuit controlling this transcriptional response. The unique data generated in this work allowed us to estimate the quality of the observational model and will assist in the development of new computational approaches to infer regulatory models (Chapter 3). Furthermore, this systematic and unbiased approach not only re-discovered 32 known regulators of TLR-dependent gene regulation (e.g. NFKB, IRFs, AP1, STATs), but also identified another 68 regulators, which have not been previously associated with the innate immune response and could only be discovered by an unbiased approach.

2.3 Profiling of mRNA levels

To determine the output of pathogen-sensing regulatory networks, we measured genome-wide expression profiles in DCs exposed to PAM3CSK4 (PAM), a synthetic mimic of

bacterial lipopeptides; polyinosine-polycytidylic acid [poly(I:C)], a viral-like double-stranded RNA; lipopolysaccharide (LPS), a purified component from Gram-negative *Escherichia coli*; gardiquimod, a small-molecule agonist; and CpG, a synthetic single-stranded DNA. These compounds are known agonists of TLR2, TLR3, TLR4, TLR7, and TLR9, respectively. Poly(I:C) also activates the cytosolic viral RNA sensor MDA-5, and LPS can also act through co-receptors such as CD14; we therefore refer to the ligands rather than their receptors for clarity. On the basis of pilot experiments (see materials and methods), we measured mRNA expression at nine time points - 0.5, 1, 2, 4, 6, 8, 12, 16, and 24 hours - after stimulation with these pathogen components. These agonists and time points reflect the most dramatic gene expression changes in DCs.

2.4 Detection of response specific genes

We focused on regulator genes whose expression change during pathogen sensing, a reasonable assumption for many mammalian responses. We defined induced and inhibited probesets for each condition (TLR agonist) as probesets that display at least 1.7 fold up- or down- regulation in both duplicates of at least one time point, as compared to the control. This set consisted of 3635 genes (Fig. 2-2). Control values were defined as the median expression calculated over control non stimulated samples times 0, 1, 2 and 4 hours.

Clustering the expression profiles of induced genes (Fig. 2-3A) we observed that the transcriptional response can be classified into two major distinct 'programs' - a 'TLR2-like' program and a 'TLR3-like' program - and a shared response (24.5% shared by TLR2/3/4). The TLR4 response is largely the union of the TLR2 and TLR3 programs (Fig. 2-3). This is in line with the known signaling pathways controlled by the different sensors: TLR2 depends solely on the Myd88 pathway, TLR3 and MDA-5 depend mostly on the TRAM/TRIF and IPS-1 pathways, while TLR4 acts through both the MYD88 and TRIF pathways [54]. It is also consistent with previous reports of the induction of an anti-viral program by TLR3 and TLR4 [55]. The genes

responsive in each of these programs are consistent with the type of pathogen that elicits each response: the TLR2-like program is enriched for NFB and inflammatory responsive genes ($P< 6.1\times10^{-8}$), whereas the TLR3 program is enriched for viral- and interferon-responsive genes ($P< 8.3\times10^{-24}$). We thus term them the "inflammatory-like" and "antiviral-like" programs. The TLR7 and TLR9 responses are very similar to the TLR4 response (more than 82% shared with TLR4) suggesting that these pathways converge on many of the same responses as the MYD88- and TRIF- TRAM dependent pathways. A notable exception is the small number of genes that are specific to a single stimulus (e.g. $\sim$ 250 for TLR3). For example, IFNB1, which is crucial to elicit elimination of viral infections, is induced at high and sustained levels in response only to TLR3 stimulation, but transiently and at substantially lower levels in response to TLR4.

2.5 Reconstruction of an observational *trans*-model of gene regulation

We next asked which regulators could account for the observed transcriptional responses. The strategy consisted in identifying potential regulators based on changes in their own expression level (a reasonable assumption for many mammalian responses [9, 56], including pathogen-sensing [1, 48]) that are also predictive of changes in those of other (target) genes. To this end, we developed an extension of the Module Networks algorithm [3] to reconstruct an observational *trans*-model of gene regulation from transcriptional profiles collected along a time course (Fig. 2-1B, top). Our method assumes that co-regulated genes have a similar regulatory program and consists of two steps: (1) divide genes into modules using the Module Networks algorithm; (2) use a regularized linear regression model [15, 22] to learn regulatory programs with time delays for each module that attempt to explain the expression of each module as a function of the expression of specific regulators. This approach would capture multiple candidate regulators with a similar role, while eliminating weak spurious con-

tributors, and would learn the temporal lags between grouped regulators and target genes.

Learning a network of interacting modules

We started our approach by applying the Module Networks algorithm as originally developed [3], which assumes that the expression of the target genes in each module is governed by the same regulatory program. Module Networks uses an iterative learning procedure using the Expectation Maximization (EM) algorithm. Each iteration consists of two steps: an E-step and an M-step. In the M-step, the procedure is given a partition of the genes into modules and learns the best regulatory program (as a decision tree) for each module. In the E-step, given the inferred regulatory programs, it re-assigns each gene to the module that best predicts the genes behavior (it does not assign a regulator gene to a module in which it is also a regulatory input, directly or indirectly). The regulatory program is chosen from a pre-defined set of candidate regulators (typically including all known and putative regulatory factors in a genome).

Here, we initialized Module Networks from 10 to 250 modules (in increments of 10) and using a set of candidate regulators from a curated list of 3287 proximal regulators of mRNA levels, including 1885 transcription factors, 1069 RNA binding proteins and 333 chromatin factors. We then chose the model whose likelihood score was 70% of the best score, to avoid overfitting of the model to the data (at this point, a straight line with slope approximately 1 crossed the likelihood score plot). The chosen model consisted of 80 modules (Fig. 2-4A) of 3635 co-regulated genes.

Regulation as linear regression

The original Module Networks algorithm [3] describes the regulatory programs as regression trees (as explained on page 15). One potential limitation of this algorithm is that it allows only a single regulator in each split in the tree, so many correct regulators may not be selected (see Fig. 2-5 which shows Klf4 as the top regulator).

Once Module Networks selects one regulator that seems to split the samples well, other regulators that are probably almost as good are not going to be added to the regulator set: since they are correlated with the original regulator they will not add additional information beyond the original split. To this end, a regression model could overcome this limitation. We also reasoned that a regression tree is not an appropriate model for time course data and that it does not exploit its full power. In particular, Module Networks does not reflect any temporal relations in the data that would allow for time lags. Allowing such lags is based on the observation that when induced regulators affect a target gene's expression through their own differentially regulated mRNA level, the induction of the target gene's mRNA expression may occur with a time lag relative to the induction of the regulator [57]. Therefore, a regression model could also have the potential of learning lagged relationships between groups of regulators and target genes. In addition, a regression model seems appropriate for our dataset since all sample are related (same cell type and system, not many different ones where context really matters).

Thus, we decided to infer regulatory programs for the modules using Elastic Net (EN, [22]), an L_1/L_2-regularized linear regression procedure. EN is an algorithm based on a regularized least square procedure with a penalty which is the sum of an L_1 penalty (like LASSO [58]) and an L_2 penalty (like Ridge Regression [59]). The first term enforces the sparsity of the solution, doing automatic variable selection, whereas the second term prevents arbitrary choice of only one out of several highly correlated variables. As shown in [22], EN outperforms LASSO in terms of prediction accuracy and tends to select strongly correlated genes into the model together when applied to microarray data. Next, according to [22], we introduce the *naïve elastic net*, followed by a scaled version, the *elastic net*.

Elastic net regularized regression

Given a dataset with n observations (experiments) and p predictors (candidate regulator genes), let $\mathbf{y} = (y_1, \ldots, y_n)^T$ be the response and $\mathbf{X} = [\mathbf{x_1}] \ldots [\mathbf{x}_p]$ be the model matrix (candidate regulators' profiles), where $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})^T$, $j = 1, \ldots, p$ are

the predictors. With a location and scale transformation we can assume the response is centered and the predictors are standardized,

$$\sum_{i=1}^{n} y_i = 0, \quad \sum_{i=1}^{n} x_{ij} = 0, \quad \text{and} \quad \sum_{i=1}^{n} x_{ij}^2 = 1, \quad \text{for } j = 1, 2, \ldots, p.$$

For any fixed non-negative λ_1 and λ_2, the naïve EN criterion is defined as

$$L(\lambda_1, \lambda_2, \beta) = \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda_2 \sum_{j=1}^{p} \beta_j^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j|,$$

and the naïve EN estimator $\hat{\beta}$ is the minimizer of that equation, that is

$$\hat{\beta} = \arg\min_{\beta} \{L(\lambda_1, \lambda_2, \beta)\}.$$

However, Zou and Hastie [22] showed that the naïve EN does not perform satisfactorily unless it is very close to either ridge regression or the lasso. This is why it is called naïve. The naïve EN estimator consists of a two-stage procedure: for each fixed λ_2 it first finds the ridge regression coefficients, and then does the LASSO-type shrinkage along the LASSO coefficient solution paths. Thus, it appears to incur a double amount of shrinkage. A scaling transformation can improve the prediction performance of the naïve elastic net by correcting this double-shrinkage. Defining an artificial dataset $(\mathbf{X}^*, \mathbf{Y}^*)$ so that

$$\mathbf{X}^* = (1 + \lambda_2)^{-1/2} \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda_2}\mathbf{I_p} \end{pmatrix},$$

where $\mathbf{I_p}$ is the $p \times p$ identity matrix, and

$$\mathbf{Y}^* = \begin{pmatrix} \mathbf{Y} \\ \mathbf{0} \end{pmatrix},$$

where $\mathbf{0}$ is the $p \times 1$ 0-vector, Zou and Hastie [22] showed that the naïve EN solves a

LASSO-type problem [21] given by

$$\hat{\beta}\,(\text{elastic net}) = \sqrt{1+\lambda_2}\hat{\beta}^*.$$

This kind of scaling undoes the overshrinking effect when the L_1 and L_2 penalties are combined. To choose the coefficients we performed a two dimensional cross validation. The least-angle regression (LARS) extension which implements the EN algorithm, called the LARS-EN, outputs a sequence of variables corresponding to a given λ_2. λ_1 has a one to one correspondence with the number of iterations that the LARS-EN algorithm was run for. Therefore selecting the active model at a given iteration for a particular λ_2 would give the EN solution corresponding to particular value of (λ_1, λ_2). To estimate the model parameters, we chose different values of λ_2 (0, 0.01, 0.1, 1, 10, 100) and the other tuning parameter was chosen using standard cross validation procedure. The chosen λ_2 was the one giving the minimum cross-validation error. The MATLAB implementation of LARS-EN available in `http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3897` was used for this work.

So, for each module generated with Module Networks, we used the LARS-EN algorithm to regress the mean profile of the module's genes with a selected combination of candidate regulators (to avoid cyclicity, we eliminated the regulator genes from the modules in this step). Simultaneously, to maximize the utility of the temporal data, we allowed a 'lag' of up to three time points between the expression of the regulators and that of the target module. This reflects a possible maximum delay of two hours between regulators and target genes (Fig. 2-6).

The EN target function optimizes only for prediction error, which is a proxy for the goal of identifying causal variables. Since not all predictive variables are necessarily causal, we further decided to reduced the number of selected variables using non-parametric bootstrap [60]. We randomly sampled the module's genes with replacement to obtain 1000 bootstrap datasets, holding off 20% of each module's genes at a time. We applied the described EN regression to each bootstrap dataset to obtain a set of regression coefficients $\hat{\beta}^B, B = 1, \ldots, 1000$, where B indicates the index of the

bootstrap set. Those variables with a non-zero coefficient in $\hat{\beta}^B$, defined a sparse set of regression solutions for the bootstrap set B. In order to obtain statistically robust variables, we computed the *selection frequency*, γ_p, for each variable p, defined as:

$$\gamma_p = \sum_{B=1}^{1000} \delta\left(\hat{\beta}_p^B \neq 0\right)/1000,$$

where $\hat{\beta}_p^B$ is the coefficient of variable p in bootstrap model B, and δ is an indicator function. We obtained set $\mathbf{X}^\star$ according to γ_p:

$$\mathbf{X}^\star = \{X_p \mid \gamma_p \geq 0.5, \quad p: \text{ index of variable}\}.$$

Thus, the set $\mathbf{X}^\star$ includes variables that had non-zero coefficients for at least 50% of the bootstrap runs. The elastic net and bootstrap procedures resulted in the choice of 117 regulators for the 80 modules (Fig. 2-4B). These included known regulators, from the NFkB, Stat and IRF families and novel intriguing candidates such as the circadian regulator Timeless and the DNA methyltransferase Dnmt3a.

2.6 Selection of candidate regulators for perturbation and a response signature

To minimize the bias in our choice of regulators, we used two other complementary strategies to identify additional sets of candidate regulators. First, we added 5 constitutively expressed regulators whose *cis*-regulatory elements are enriched in the responsive genes. These included Irf3, Rela, Nfe2l2, Ets1, Creb3. Second, we compared the transcriptional profiles of several known regulators to those of their targets and found that several relations will likely be missed by the model either because of specific 'windows' of regulation or highly non-linear relations; to maximize our ability to understand such relations, we incorporated any other regulator that had at least a 2-fold change in expression (22 added regulators). Overall, this resulted in 144 candidate regulators for further analysis (Fig. 2-7). The expression of the candidate

regulators covers the full transcriptional response (27% 'TLR2-like', 46% 'TLR3-like', 30% in both responses, albeit with different kinetics).

Notably, the transcriptional profiles and timing of activation of the regulators under TLR4 activation are conserved between DCs and the functionally similar macrophages especially at early time points (Pearson correlation $r \sim 0.9$ at 1h, Fig. 2-8A) and between human macrophages and mouse DC ($r \sim 0.6$ at 2h, Fig. 2-8B). This conservation supports the relevance of the regulator's transcription to the network's function.

To facilitate large-scale perturbation and monitoring, we next determined a representative gene signature and a time point post-stimulation that captures the complexity of the different programs. We devised *GeneSelector*, an information-theoretic approach, for selecting genes that are highly informative about the stimulus. In this approach, the expression levels of all genes depend on the experimental stimulus, and we employ conditional entropy as a measure of the remaining uncertainty about the stimulus once the expression levels of the signature genes are known. We used a greedy procedure to incrementally select genes that maximally reduce the entropy given the previously selected genes. We applied this approach repeatedly to select multiple disjoint gene sets, until we reached a set of 80 genes (the set size was limited by the experimental detection method). We used the same approach to choose a single time point, six hours post activation, at which the conditional entropy was minimal and hence best distinguishes the stimuli (see materials and methods). Finally, we chose a single treatment (LPS), since it activates both the viral and inflammatory gene programs. By this principled selection of a 'reporter signature' and a single time point post-stimulation we can adequately represent the complexity of the transcriptional response. Notably, we added two types of internal control genes. First, we added 10 genes whose expression level is unchanged under any TLR stimulation, but whose (constant) basal levels vary from very low to high. Second, we added to the reporter genes the 37 candidate regulators that had a significant mRNA level at the 6 hours time point chosen for the screen. These added genes would allow us to assess knockdown efficacy and the sensitivity of our statistical methods. The final set thus

included 118 'reporter' genes and 10 controls.

2.7 Perturbation experiments and network reconstruction

To provide causal data and to rigorously test the model just described, we used RNAi libraries to test the roles of the selected candidate regulators in controlling gene expression. We generated a validated lentiviral shRNA library for 125 of the 144 candidate regulators (Fig. 2-9) and used it to systematically perturb each of the regulators in DCs. To carry out our perturbational study, we selected a single treatment, LPS, that activates the majority of both the "inflammatory-like" and "antiviral-like" programs. After stimulation of shRNA-perturbed DCs with LPS for 6 hours, we profiled the expression of a signature of 118 marker genes (see [6] for details on this selection) with the Nanostring nCounter system that provides a fast and cheap multiplex assay [61].

The changes in signature gene expression resulting from infection with each shRNA were used to construct a model that associated regulators to their targets. We expected increases in the transcript levels of reporter genes whose repressors are targeted by knockdown, and decreases in reporters whose activators are targeted. Our false discovery rate (FDR) model estimates the statistical significance of a change in transcripts in DCs infected with a given shRNA. We controlled for gene specific noise by comparing to changes in the expression of each gene after perturbation with the control shRNAs (Fig. 2-10A), and for shRNA-specific noise by comparing to changes in the expression of the control genes after a given shRNA perturbation (Fig. 2-10B). We estimated the sensitivity of our calls from the 37 regulators, which are also included as target reporters (Fig. 2-11).

On the basis of these results, we identified a densely overlapping network with 2322 significant regulatory connections, including 1728 activations and 594 repressions (Fig. 2-10C, red and blue, respectively, at 95% confidence). Of the 125 tested regulators,

we confidently identified 100 with at least four targets. Among those were 24 hub regulators that were predicted to regulate more than 25% of the 118 genes measured, as well as 76 specific regulators each affecting the expression of 4 to 25 genes. On average, $\approx$ 14 ($\pm$8; SD) regulators activated a target gene, and 5 ($\pm$5.8) regulators repressed it. Indirect effects may account for the large number of regulators we observed for each target, and we discuss how to address those in Chapter 3.

The perturbational model captured many known regulatory relations - for example, the NF-kB family of transcription factors (Rel, Rela, Relb, Nfkb1, Nfkb2, and Nfkbiz) regulating their known inflammatory gene targets. Our network provided evidence for the involvement of at least 68 additional regulators in the response to pathogens, of which 11 were hubs not previously associated with this system.

Focusing on the network architecture, we found multiple feedforward circuits in this response, where an upstream regulator controls a target gene both directly and indirectly through a secondary regulator [62]. The majority (76%, 4892 of 6444) of these feedforward circuits were found to be coherent [62], having the same direct and indirect effect on the regulated gene. The vast majority (80%) are type I loops [63] with all-positive regulation (e.g., Nfkbiz activates E2f5 and both activate IL-6). Such feedforward circuits respond to persistent rather than transient stimulation, protecting the system from responding to spurious signals, as was shown for one circuit in LPS-stimulated macrophages [52]. Our finding suggests that coherent feedforward loops, especially class I [62], are a general design principle in this system and may have a physiological impact on this response.

2.8 Comparison between the observational and the perturbation model

When we compared the results from the observational and the perturbational models, even though there were some true positive associations in the observational model, there were also substantial discrepancies. Fig. 2-12 illustrates the correspondence

between the models. For example, the perturbational model indicates (confidence value above 0.95) that Arid5a represses Cxcl5, but the pair was not captured in the observational model (Fig. 2-13A). More importantly, we found a substantial number of false positive relations in the observational model, mostly due to the fact that both the correct regulator and many others have indistinguishable expression patterns (Figs. 2-14 and 2-15). This phenomenon is observable regardless of the specific variation of the observational model, using different time lags (none, variable lags) and different number of treatments (all treatments or only LPS). Fig. 2-13B shows gene pair Bbx and Hbegf which was predicted in the observational model due to the high correlation, but where the regulatory relation was not confirmed by perturbation.

Furthermore, from the perturbational dataset, we observed that correlation is not a good indicator of the type of activity of a gene. For example, Stat2 positively correlates ($r \sim 1$) with Acpp and Isg20, but the perturbational model identified Stat2 as a repressor of Acpp and as an activator of Isg20. On the other hand, Cebpz negatively correlates ($r \sim -1$) with BC006779 and Tcf4, but Cebpz activates BC006779 and represses Tcf4. These observations indicate that expression data alone do not suffice to provide functional information about gene regulation. However, most of the regulators predicted by the observational model were functional.

2.9 Discussion

This study opens the way for the development of a next-generation of computational approaches to infer regulatory models from genomic data such as genome-wide expression profiles. While there are many computational approaches to derive observational models, it has been exceedingly difficult to estimate their quality [8]. The unique data generated in this work includes both a comprehensive set of expression profiles for learning a model (training data), and a large scale perturbational screen for estimate its quality (test data). The broad utility of our screen is enhanced by the largely unbiased approach in our choice of candidates for perturbation.

The draft model predicted many of correct regulators ('key players') as well as

validated regulatory interactions but at a cost of a relatively high false positive rate in interactions; this was mainly due to the fact that the method is based on a linear regression and thus many of the regressors that are selected are the ones that have a very high absolute correlation value with the dependent variable. One potential way of overcoming this limitation is the incorporation of *cis*-regulatory information [1, 50]. However, such models are biased since they can neither detect novel transcription factors whose binding sites are not known, nor the involvement of chromatin factors which do not bind specific sequence elements. Indeed, only 25 of our regulators have a known binding site matrix (at the appropriate protein family level). 12 of these sites are enriched in our TLR responsive genes (Materials and Methods) and all are associated with factors that are well known to be involved in this response. Sites for some of the 12 factors (e.g. Ets1, Sp1, Klf4, Creb, Egr1, E2f, Plagl1) were enriched based on the timing of expression but not the specific pathogen. Overall, while this analysis is consistent with our model, it did not expand the scope of regulators beyond those previously known, and it could not discover many of the key novel regulators in the perturbational model, since they do not have known sites.

A central goal of the study was to address the mechanistic basis for pathogen-specific responses. Consistent with previous studies [55], we distinguished two key programs, a PAM (TLR2)-like inflammatory response and a poly(I:C) (TLR3/MDA-5)-like antiviral response, which are together induced by LPS, a Gram-negative bacterial component and a TLR4 ligand. These programs reflect both qualitative and quantitative differences between the required functional responses, and are consistent with the cross-protection between certain bacteria and virus infections [55]. The broad effect of LPS allowed us to focus on a single stimulus and time point, but screens with other stimuli may identify additional unique regulators.

Our study has benefited from several features of the DC system, as well as from careful design. Most notably, the organization of the DC transcriptional response allowed us to identify regulators with a scope much broader than the single stimulus and time point used in our screen. Nevertheless, we except that additional functional regulators can be identified by testing additional stimuli (e.g. poly I:C for specific

anti-viral genes) or time points (e.g. early time points to study pulse-like responses).

Generally, this work establishes an unbiased, straightforward, and general framework for network reconstruction in mammalian cells, including several strategies to leverage shRNA for the study of gene regulation. This approach can be executed at substantial scale and reasonable cost, and is compatible with the challenge of deciphering the multiple regulatory systems that operate in mammals. It can be expanded to derive increasingly detailed models and to distinguish direct from indirect targets.

2.10 Materials and methods

Mouse dendritic cells

6-8 week old female C57BL/6J mice were obtained from the Jackson Laboratories. Bone marrow dendritic cells (BMDCs) were collected from femora and tibiae and plated on non-tissue culture treated plastic dishes in RPMI medium (Gibco, Carlsbad, CA, Invitrogen, Carlsbad, CA), supplemented with 10% FBS, L-glutamine, penicillin/streptomycin, MEM non-essential amino acids, HEPES, sodium pyruvate, β-mercaptoethanol, and GM-CSF (15 ng/mL; Peprotech, Rocky Hill, NJ). These cells were used directly for all RNAi experiments. For all other experiments, at day 5, floating CD11c+ cells were sorted on the autoMACS separator with the CD11c (N418) MicroBeads kit (Myltenyi Biotec, Auburn, CA). CD11c+ cells where replated at a concentration of 106 cells/ml and stimulated 16 hours post sorting.

TLR agonists experiments

All ligands were purchased from Invivogen (San Diego, CA) and used at the following concentrations: PAM3CSK4 (250 ng/ml), polyI:C (10 ug/ml), LPS (rough, ultra-pure E.coli K12 strain LPS, 100 ng/ml), gardiquimod (250 ng/ml), CpG DNA (1ug/ml). We first optimized a synchronized and robust activation of CD11c+ DCs to different ligands and found that non-charged plastics ('Petri dish') retain the DCs in a näive-like state on the basis of both cell phenotype and selected gene markers.

mRNA isolation

Total RNA was extracted with QIAzol reagent following the miRNeasy kit's procedure (Qiagen, Valencia, CA), and sample quality was tested on a 2100 Bioanalyzer (Agilent, Palo Alto, CA). RNA was reverse transcribed with the High Capacity cDNA Reverse Transcription kit (Applied Biosystems, Foster City, CA). For experiments with more than 12 samples, we harvested PolyA+ RNA in 96- or 384-well plates with the Turbocapture mRNA kit (Qiagen) and reverse transcribed with the Sensiscript RT kit (Qiagen).

Pilot experiments

To choose the parameters (culture conditions, ligands, and time points) for the full experiments we conducted several pilots. We first optimized a synchronized and robust activation of CD11c+ DCs to different TLR ligands and found that non-charged plastics ('Petri dish') retain the DCs in a näive-like state based on both cell phenotype and selected gene markers. We then used microarrays to profile mRNA levels at a few time points to identify the time windows when: (a) DCs become quiescent following CD11c+ positive selection; and (b) TLR ligands regulate DC gene expression and phenotype. Next, we determined the full set of time points based on qRT-PCR measurements of a small number of marker genes along a high-resolution time course. Based on these pilot studies, we profiled mRNA expression at 9 time points (0.5, 1, 2, 4, 6, 8, 12, 16, and 24 hours) following stimulation with Pam3CSK, polyIC, LPS, gardiquimod, and CpG.

Array hybridizations and processing

For oligonucleotide microarray hybridization, 1.5g RNA were labeled, fragmented and hybridized to an Affymetrix Mouse Genome 430A 2.0 Array. After scanning, the expression value for each gene was calculated with RMA (Robust Multi-Array) normalization [64]. The average intensity difference values were normalized across the sample set. Probe sets that were absent in all samples according to Affymetrix

flags were removed. All values lower than 50 were replaced by 50.

GeneSelector

To choose a set of genes that will capture as much as possible of the information on the expression of all genes, we used an information-theoretic approach. We modeled the expression levels X given the experimental condition C with a naïve Bayes model where the expression of gene i under condition c follows a normal distribution $X_i|C = c \sim N(\mu_{ic}, \sigma_i^2)$. In this model, the expression levels of all genes depend on the experimental condition C, and we selected genes that are highly informative about C.

Formally, for a set of genes Y we used the conditional entropy

$$H(C|Y) = -\sum_c p(C=c) \sum_y p(Y=y|C=c) \, log \, p(C=c|Y=y)$$

as a measure of the remaining uncertainty in C once the expression levels Y are known. We then used this measure and a greedy procedure to select multiple disjoint gene sets $Y_1, \ldots, Y_k$, such that for each set Y_i, $H(C|Y_i) < \eta$ (we set $\eta = 0.5$). In the greedy procedure, we select genes one at a time, and with each selected gene re-compute the entropy given the genes already selected in the current set. Once a set is complete (the remaining conditional entropy is below the threshold η), we add all the genes to the selected set, and repeat the procedure (excluding all the selected genes from consideration). We stop when the number of selected genes has reached a user-defined threshold, set by the number of genes feasible for the experimental assay.

To select a time point, we used the same approach. Here, we measured entropy under all time points for multiple randomly selected gene sets of several sizes and plotted the average entropy for each timepoint. We chose the time point with the minimal entropy.

Analysis of *cis*-regulatory elements

Each *cis*-regulatory element was represented by a Position Weight Matrix (PWM). We compiled a set of 1651 PWMs from the TRANSFAC matrix database v8.3 [65], JASPAR Version 2008 [66] and experimentally determined PWMs [67, 68]. Given a PWM, for each nucleotide position in the promoter of each mouse gene, we calculated an affinity score defined as the log-likelihood ratio (LOD score) for observing the sequence given the PWM versus a given random genomic background. We then found the best conserved motif instance over the entire promoter region. We automatically computed a PWM-specific cutoff, by using the information content of each motif, computed as the 2-IC quantile of the PWM LODs distribution. We considered a 'hit' in the promoter if the maximal LOD score was above this cutoff. Finally, we computed the enrichment of the motif in each of six clusters determined by the microarray experiments, using a two-sided Wilcoxon rank-sum test between the set and the background (see Table S13 in the *Science* manuscript website). To ensure that enrichment was not due to nucleotide bias within the promoter, we also shuffled the PWM and computed enrichment for the true PWM compared to the shuffled PWMs. A motif was considered enriched in a gene set if it passed P-value$<$0.01.

shRNA Perturbation experiments

We generated validated shRNAs that knockdown each of the regulators by at least 75%. For each regulator, we tested five shRNAs [53] by introducing each shRNA into in vitro cultured mouse bone marrow cells using a lentivirus expression system followed by puromycin selection. We tested the effect of each shRNA on the target transcript using qPCR and found an shRNA with a knockdown efficiency greater than 75% for 125 of our 144 regulators. Our lentivirus-infected bone marrow cell population was composed of 90% DCs as determined by FACS using the CD11c marker. Smaller populations of uncharacterized cells largely retain the expression profiles of DCs for most genes, including the regulators. Lentivirus infection four days prior to TLR4 stimulation does not significantly activate the DC population,

as reflected by the very low expression levels of major cytokines and co-stimulatory molecules, comparable to that of uninfected cells. Furthermore, the infected cells retain the ability to robustly respond to a pathogenic stimulus. Finally, knockdown of the individual regulators does not cause a significant change in differentiation or basal activation of the DC population prior to LPS treatment, as reflected by staining of each knockdown with CD11c, a positive marker for mature DC, and with CD86 a marker for 'activated' DC.

Reproducibility and reliability of perturbation experiments

The results of the perturbation experiments are highly reproducible and reliable as indicated by several controls of sample variability. First, biological duplicates following re-infection with eight of the shRNAs on different cells from the same mouse were highly correlated ($r > 0.995$ in all cases; a technical duplicate has $r > 0.99989$). Second, biological duplicates following re-infection with 18 of the shRNAs of a new batch of bone marrow cells from a different mouse were also highly correlated ($r > 0.9$). Third, infection with a lentiviral vector harboring a second shRNA construct for 18 of the tested regulators led to correlated effects ($r > 0.8$). Notably, these additional constructs resulted in a lesser, though still robust, knock down efficiency greater than 60%. Fourth, the mRNA levels measured by the nCounter assay were consistent with qPCR validation for four target genes (IL-12, PTGS2, IL6, CXCL10) tested under 20 different shRNAs in an independent experiment, demonstrating the robustness of these results. Finally, the mRNA levels measured by nCounter following infection with five shRNAs targeting regulators and two control non-targeting shRNAs were highly ($r > 0.88$ in all cases) correlated to those measured on Affymetrix following independent infections with the same constructs.

mRNA measurements with nCounter

Details on the nCounter system are presented in full in [61]. CodeSets were constructed to detect genes selected by the GeneSelector algorithm and additional controls as

described. 5*104 bone marrow cells were lysed in RLT buffer (Qiagen) supplemented with β-mercaptoethanol. 10% of the lysate was hybridized for 16 hours with the codeset and loaded into the nCounter prep station followed by quantification using the nCounter Digital Analyzer.

Normalization of nCounter data

We normalized the nCounter data in three steps. In the first step, we controlled for small variations in the efficiency of the automated sample processing. To this end, we followed the manufacturers instructions, and normalized measurements from all samples analyzed on a given run to the levels of a chosen sample (in all cases we used the first sample in the set). This was done using the positive spiked-in controls provided by the nCounter instrument.

In the next step, we relied on ten control genes (Ik, Ndufa7, Tomm7, Tbca, Ndufs5, Ywhaz, Mea1, Rbm6, Shfm1, and Gapdh) which were included as reporters and were identified from the microarray experiment as unperturbed upon stimulation by any of the pathogen components. We found that two of these genes (Gapdh and Rbm6) showed too much variation and removed them from all subsequent analysis. We used the remaining eight genes for a second round of normalization. For every sample, we computed the weighted average m_i of the mRNA counts of the seven transcripts and normalized the samples values by multiplying by the constant m_1/m_i. Finally, we obtained a normalized expression quantity that takes into account the intrinsic noise in our system. We used the 32 samples treated with control shRNAs (that do not target any gene in the mouse genome) to define a z-statistic z for each observation o_{ij} of transcript i in each shRNA experiment j: $z_{ij} = \frac{o_{ij}-m_j}{s_j}$ where m_j and s_j are, respectively, the mean and variance of the expression of transcript j in the control shRNA experiments.

Confidence estimates for differential expression in perturbation experiments

We used two permutation-based approaches to estimate our confidence in an observed z-score value for a transcript in an shRNA experiment. In the first approach, we defined a per-gene confidence score for each measurement, by using the variation in that gene's expression in the control shRNA experiments. We computed the confidence scores for each measurement (one gene in one experiment) at a time, by swapping the measured value of that gene with each of its measurements in the control experiments in turn, and re-calculating a new z-score. We then assessed the significance of the real z-score, given the distribution of the permuted scores as a null distribution. More formally, for each of the observed counts o_{ij} of the reporter gene i in sample j we generated r permuted values (where r is the number of control shRNA experiments) as follows. Let $c_{i1}, \ldots, c_{ir}$ be the r transcript counts for gene i in each of the r control experiments. The k permuted z-score is obtained by swapping o_{ij} with c_{ik} and computing a z-score as $z_k^{ij} = \frac{c_{ik} - m_{ij}^k}{s_{ij}^k}$, where $m_{ij}^k = \frac{c_{i1}+\ldots+c_{ik-1}+o_{ij}+c_{ik+1}+\ldots+c_{ir}}{r}$ and $s_{ij}^k = \frac{\sqrt{\left(c_{i1}-m_{ij}^k\right)^2+\ldots+\left(c_{ik-1}-m_{ij}^k\right)^2+\left(o_{ij}-m_{ij}^k\right)^2+\left(c_{ik+1}-m_{ij}^k\right)^2+\ldots+\left(c_{ir}-m_{ij}^k\right)^2}}{\sqrt{r}}$. We take the permuted scores z_k^{ij} as a null distribution and the FDR for a given z-score z_{ij} for gene i in experiment j is given as $FDR(z) = \frac{E_k\left(\#\left\{z_k^{ij} | z_k^{ij} > z; j=1,\ldots,n\right\}\right)}{\#\{z_{ij} > z; j=1,\ldots,n\}}$, where n is the number of shRNA experiments. The confidence for z is $conf(z) = 1 - FDR(z)$.

In the second approach, we devised a per-experiment confidence score for each measurement. We use a similar procedure to control the FDR on the z-statistic, based on variation in the expression of control genes in each experiment. Formally, let $z_{ij}, \ldots, z_{nj}$ be the z-scores for the j^{th} experiment (shRNA), and assume the first l transcripts are control transcripts whose expression does not change in response to any pathogen component ($l = 8$, see above). We defined $\tilde{z}_{ij} = \frac{z_{ij} - \tilde{m}_j}{\tilde{s}_j}$ where now $\tilde{m}_j$ and $\tilde{s}_j$ are, respectively, the mean and variance of the z-scores of the control transcripts $1, \ldots, l$ in the j^{th} shRNA experiment. We perform l permutations as described above, by swapping each observed z-scores with a control transcript score and computing $\tilde{z}$, then computing an FDR as above.

False Discovery Based (FDR) model

We used the signature measurement across 125 genetic perturbations to construct a regulatory model, associating regulators to their targets. We expect an increase in the transcript levels of a reporter gene whose repressor is targeted by knockdown, and a decrease in a reporter whose activator is targeted. We devised a statistical procedure to estimate, at a desired FDR, which transcripts are significantly decreased or increased in DCs infected with a given shRNA, as compared to their expected level in the absence of the shRNA perturbation. Our procedure controls for two potential sources of noise. First, we assess gene-specific noise based on changes in the expression of each reporter gene following infection with 32 control shRNAs (Fig. 2-10A). Second, we assess shRNA-specific noise based on changes in expression of 8 control genes following a given shRNA perturbation (Fig. 2-10B). (These control genes do not change following any TLR stimulation in wild type DCs). Together, these methods allow us to devise two confidence measures to estimate the significance of change in each reporter gene under each shRNA perturbation.

We estimated the sensitivity of our calls based on our ability to accurately identify the knockdown of the 37 regulators that are also included as target reporters and should be knocked down by specific shRNAs (two additional regulators, Fos and Klf10, are included as targets but are themselves down regulated in the native response and hence cannot be scored). At 95% confidence, the gene-specific FDR accurately scored the knockdown target as significantly repressed for each of these 37 genes. The shRNA-specific FDR accurately scored 80% of the 37 regulators/targets as significantly repressed. This suggests that our second FDR is likely an over-conservative estimate of the confidence.

2.11 Figures

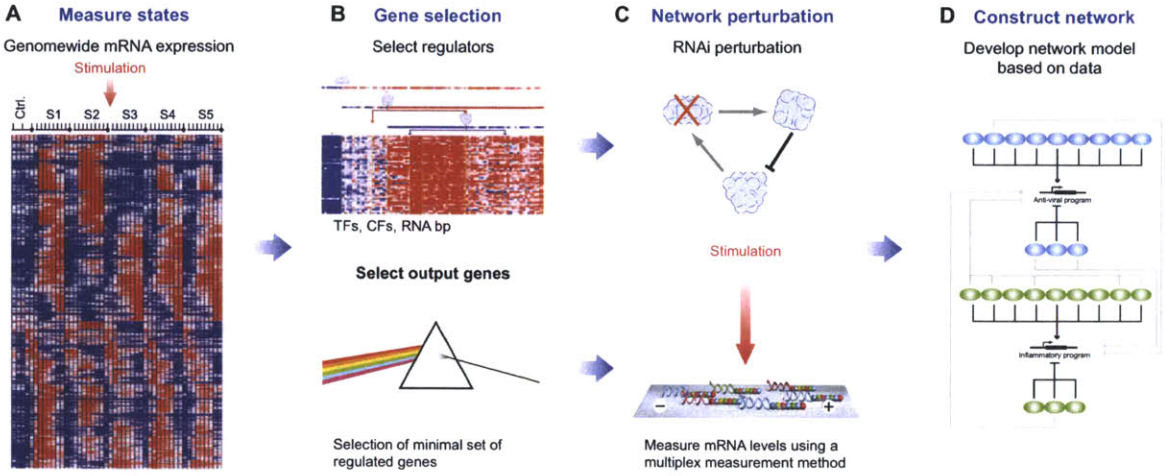

Figure 2-1: A systematic strategy for network reconstruction. The strategy consists of four steps (left to right). (**A**) State measurement. We use genome wide expression profiles under different stimuli (S1-S5), at different time points (tick marks). Rows genes, columns experiments, red induced, blue repressed, white unchanged. (**B**) Gene selection. We identify candidate regulators that are transcriptionally regulated and predictive of the expression of gene modules (top). We select a signature of target genes that maximally represents the full expression profile (bottom). (**C**) Network perturbation. We generate a functionally validated shRNA library for all potential regulators and use it to knockdown each regulator (top). Following stimulation of genetically perturbed cells (red arrow), we measure the expression of the signature genes using the nCounter multiplex mRNA detection system (bottom) (**D**) Network reconstruction. We combine genome-wide expression profiles and perturbed multiplex measurements to reconstruct a regulatory network associating regulators with individual targets and overall responses.

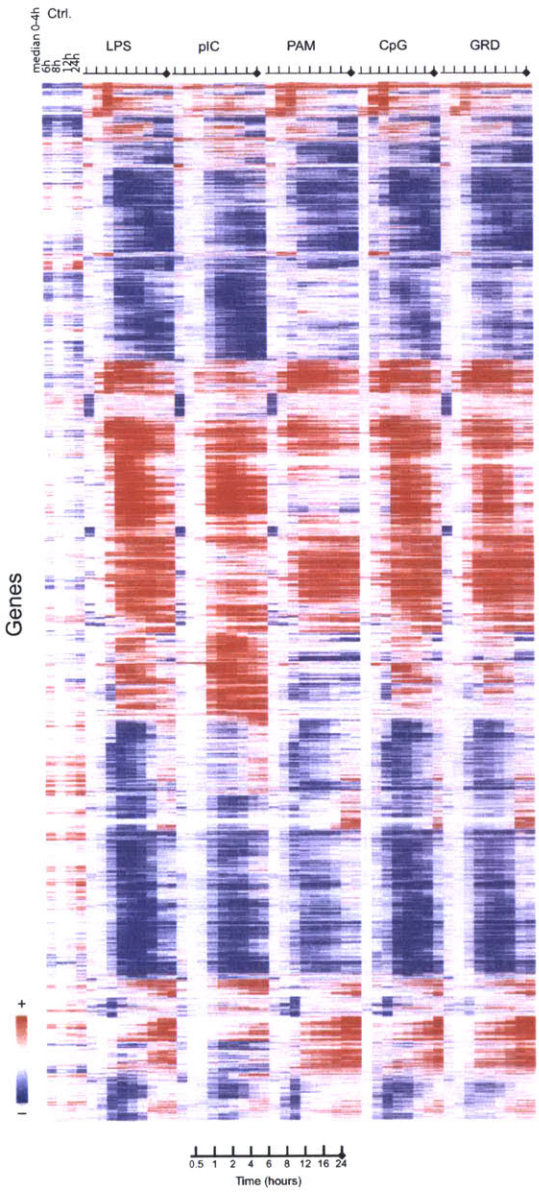

Figure 2-2: Expression profiles of the 3635 genes whose expression was at least 1.7 fold up- or down- regulated in both duplicates of at least one time point as compared to the control, in CD11c+ DCs stimulated with the indicated pathogen component across a time course of 0, 0.5, 1, 2, 4, 6, 8, 12, 16, or 24 hours (tick marks). Replicates were collapsed and genes hierarchically clustered (rows, genes; columns, experiments; red, induced from baseline; blue, repressed from baseline; white, unchanged from baseline).

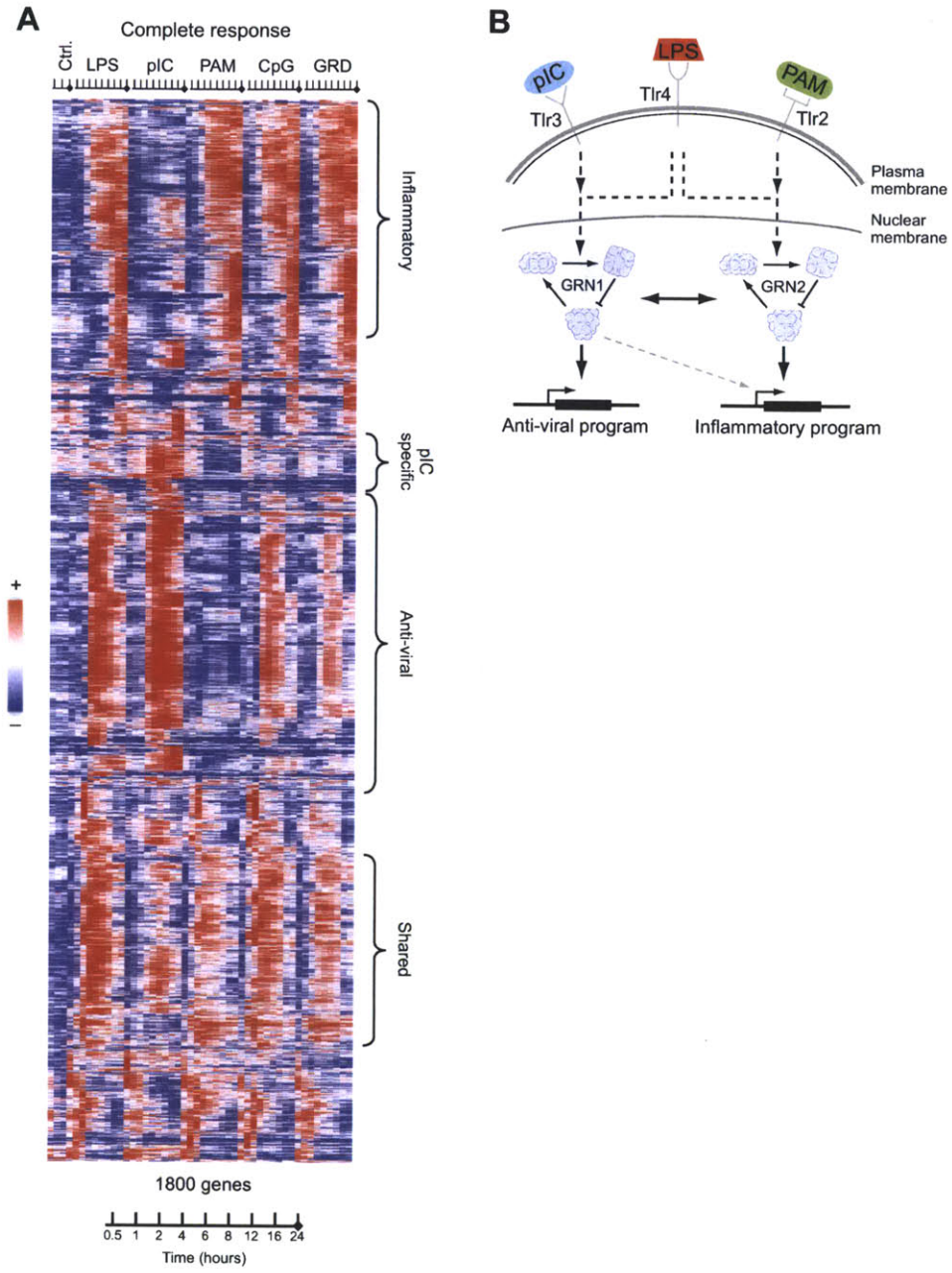

Figure 2-3: Gene expression response to pathogen components. (**A**) mRNA profiles of the 1800 genes whose expression was induced by a factor of at least 1.7 from baseline level in both duplicates of at least one time point in CD11c+ DCs stimulated with the indicated pathogen component across a time course of 0, 0.5, 1, 2, 4, 6, 8, 12, 16, or 24 hours (tick marks; pIC, poly(I:C); GRD, gardiquimod). Replicates were collapsed and genes hierarchically clustered (rows, genes; columns, experiments; red, induced from baseline; blue, repressed from baseline; white, unchanged from baseline). (**B**) Model illustrating the differential gene regulatory networks controlling the antiviral ["poly(I:C)-like"] and inflammatory ("PAM-like") programs.

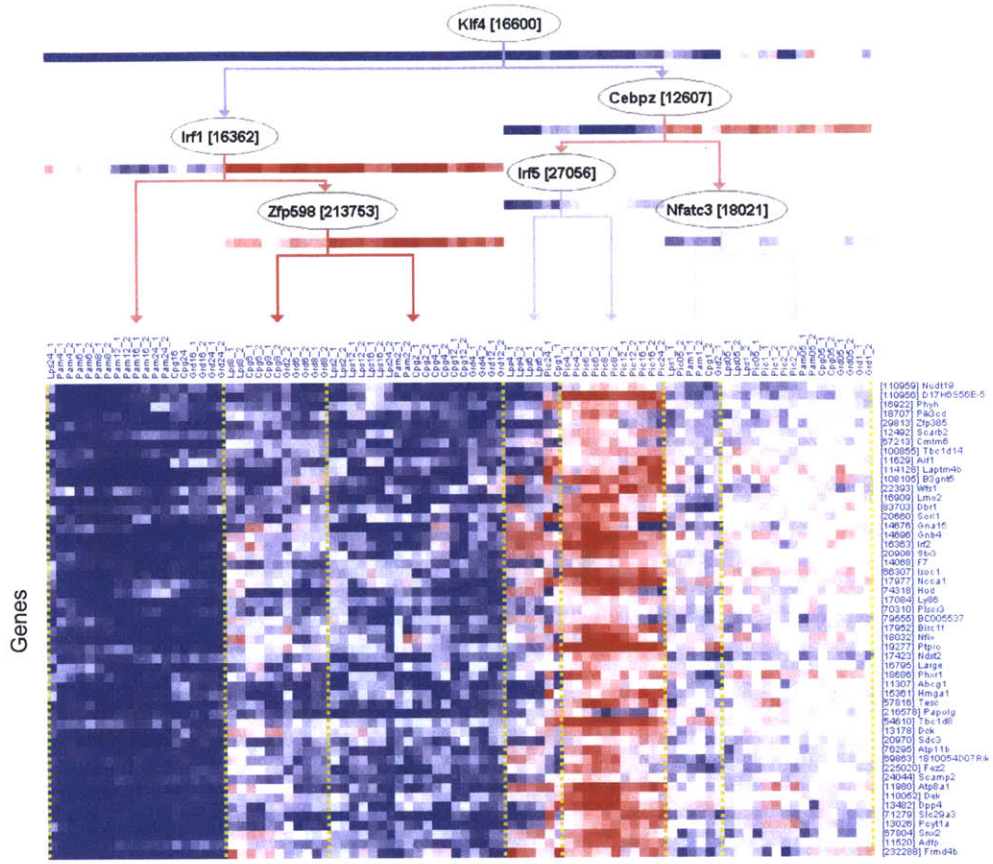

Figure 2-5: Example module. Rows are genes, columns are experiments, and the tree represents the regulatory program for the module.

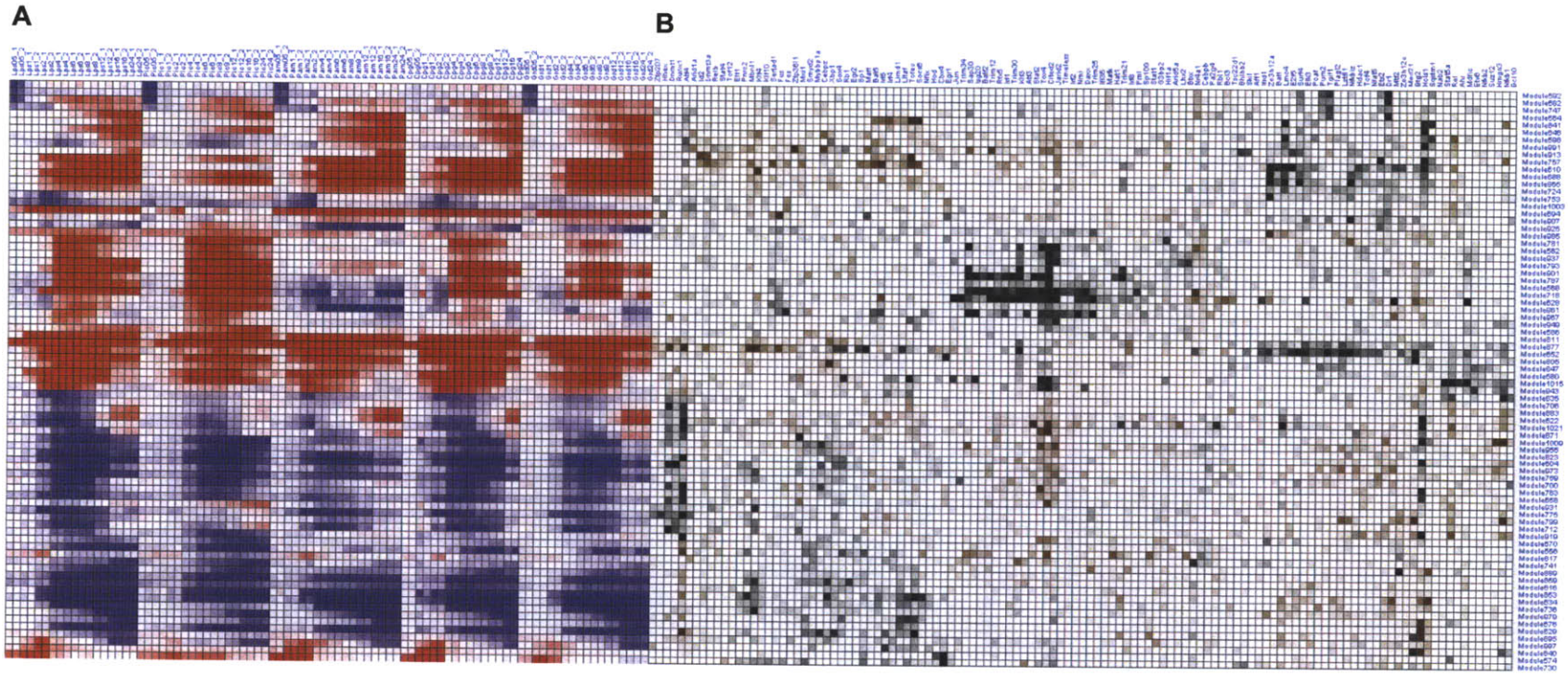

Figure 2-4: A module networks-LARSEN model of candidate regulators and target modules. (**A**) Mean expression of each of 80 modules (rows) across the five time courses. Red - induced; blue - repressed; white - unchanged; all values are relative to the t=0 baseline. (**B**) The regulators (columns, hierarchically clustered) associated with each of the modules. Black/brown - positive/negative coefficients from the regularized regression (the stronger the color intensity the higher the regression coefficient). Rows are hierarchically clustered according to the mean expression data in (A).

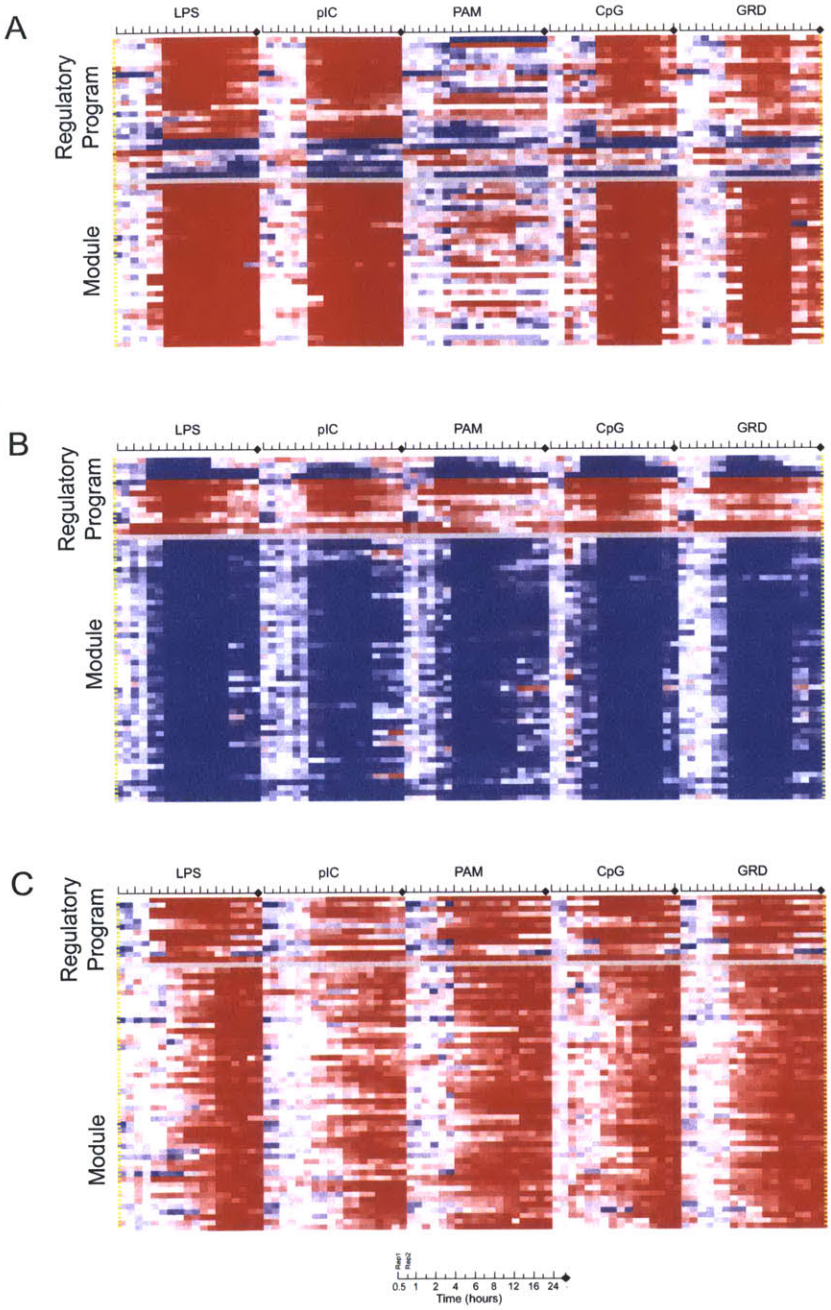

Figure 2-6: Three modules and the regulatory programs learned with the LARS-EN approach across the five time courses of 0, 0.5, 1, 2, 4, 6, 8, 12, 16, and 24 hours (tick marks show both replicates for each time point) for the five stimulated pathogens. Rows are genes, columns are experiments, the regulatory programs are above the grey row for each module. (A) Target module with no time delay. (B) Target module with 0.5h time delay. (C) Target module with 2h (maximum) delay.

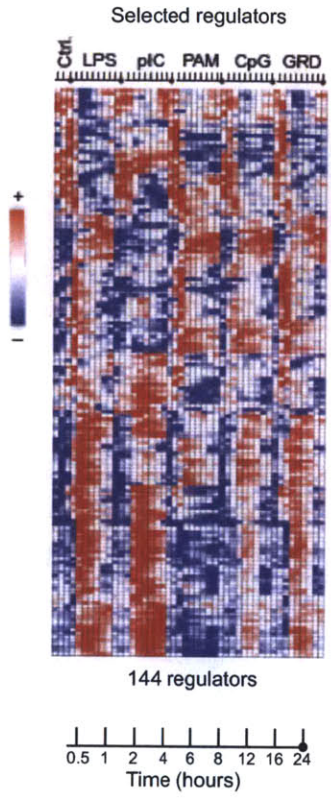

Figure 2-7: Gene expression profiles of selected regulators. CD11c+ dendritic cells were stimulated with the indicated pathogen component across a time course of 0, 0.5, 1, 2, 4, 6, 8, 12, 16, 24 hours (tick marks), followed by measurement of whole genome mRNA profiles. Shown are the expression profiles for the 144 TFs, RNA binding proteins and chromatin modifiers selected for perturbation. Replicates are collapsed and genes are hierarchically clustered.

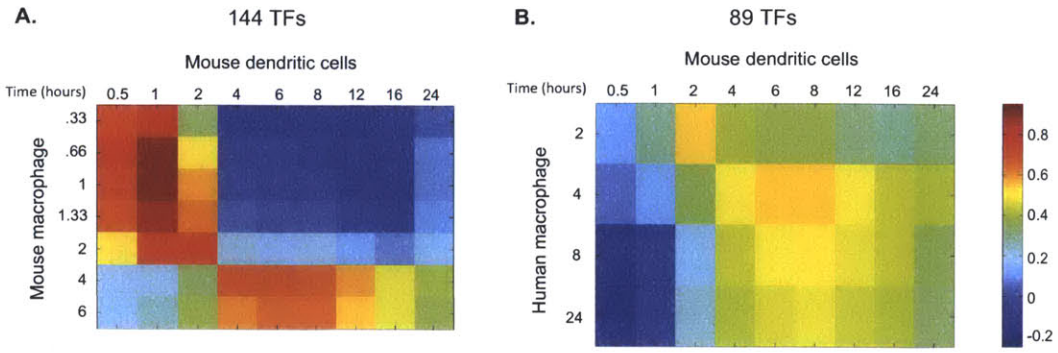

Figure 2-8: Correlation between mouse DC profiles to mouse macrophages and human DCs. **A**) Pearson correlation coefficients between the expression profiles in this study (columns) and in LPS-stimulated mouse macrophages (15) (rows), when the profiles are restricted only to the 144 regulators as in Fig. 2-7. **B**) Correlation between the expression profiles of human macrophages and mouse DCs both stimulated by LPS, on the basis of only 89 regulator genes. The 89 regulators are all those out of the 144 candidates (as in Fig. 2-7) that could be mapped between the two organisms and arrays.

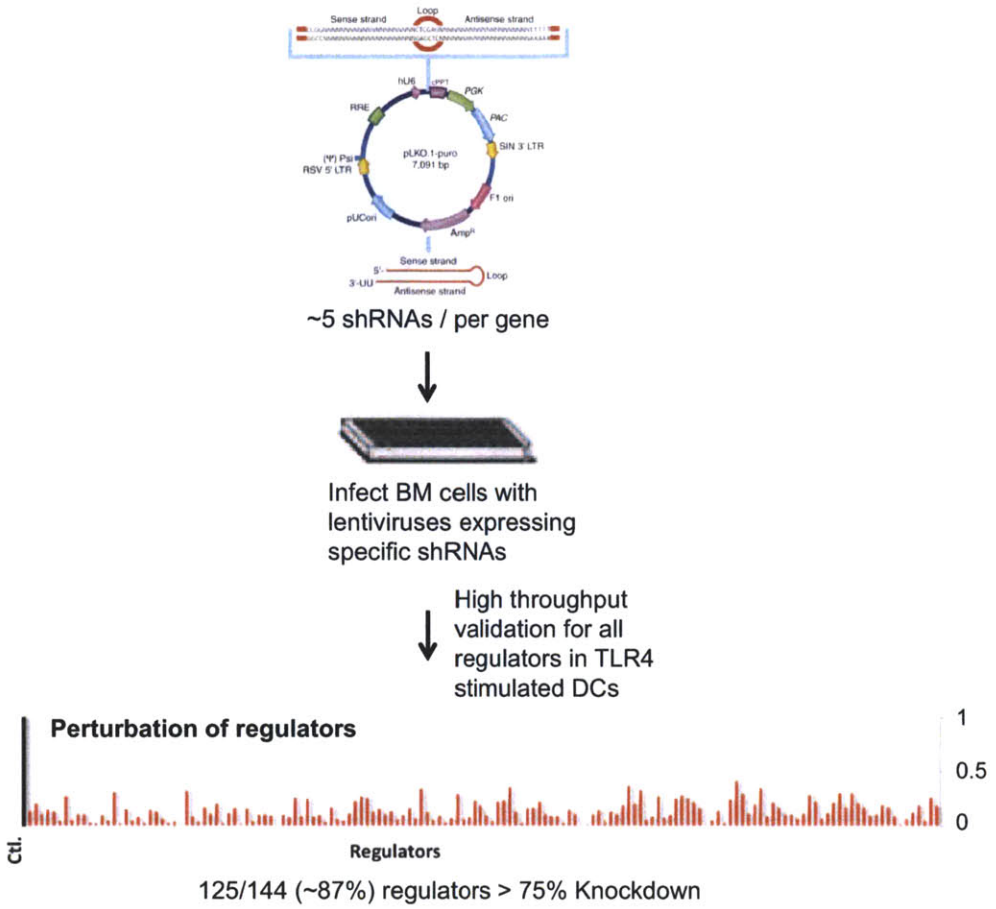

Figure 2-9: Validation of shRNA knockdown efficiency. Five lentiviruses expressing independent shRNAs targeting each of 144 induced regulators were generated (top). Bone marrow cells were infected with control or regulator shRNA viruses and stimulated for 2 or 6 hours with LPS (middle). PolyA+ RNA was prepared and reverse transcribed with random nonamers followed by quantitative real-time PCR with primers specific to each regulator gene. Relative knockdown efficiency for each regulator was calculated relative to its level in a set of experiments with control shRNAs ('Ctl') (bottom). 125 of 144 candidate regulators had an shRNA which caused greater than 75% knockdown.

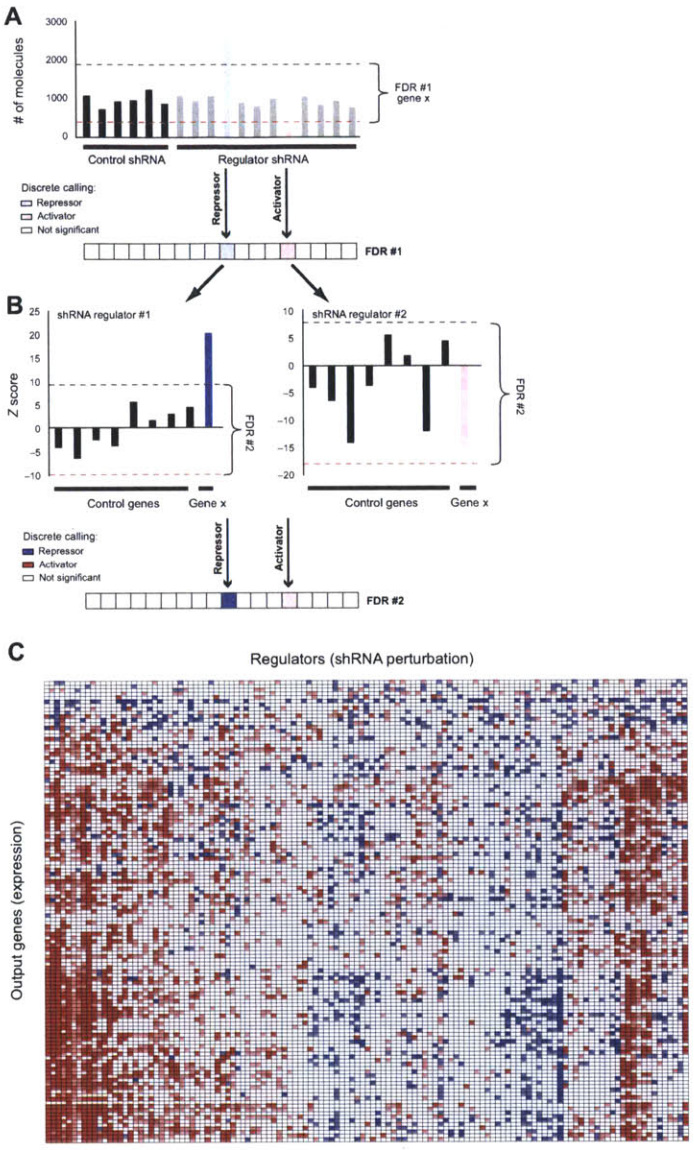

Figure 2-10: Gene regulatory programs controlling the response to pathogen components. (**A** and **B**) A strategy to minimize the false discovery rate (FDR) calls of significant changes in an output target gene resulting from knockdown of a regulator gene. (A) The first FDR procedure (top) compares the expression of the gene after a perturbation with a regulator shRNA (right) to its expression upon perturbation with 32 nontargeting shRNAs (left). The dashed lines identify the gene-specific FDR-based thresholds for induction (blue line) and repression (red line). A discrete vector of significant calls (bottom) is derived from the raw data (blue, regulator represses the target gene; red, regulator induces the target gene). (B) A second FDR procedure (top) compares the expression of the target gene to that of eight control (target) genes upon perturbation with the same shRNA. In the example shown, the genes induction (left) was significant relative to the variation in expression among the control target genes, resulting in a high score (bottom, dark blue), but its repression did not significantly differ from the controls, resulting in a lower score (bottom, weaker red). (**C**) A heat map showing all the significant relations between the perturbed regulators (columns) and the measured targets (rows), colored as in (B). Darker colors - high-confidence calls passing both a gene-noise model (FDR#1) and an shRNA-noise model (FDR#2) at 95% confidence. Light colors - calls passing only the gene-noise model at 95% confidence.

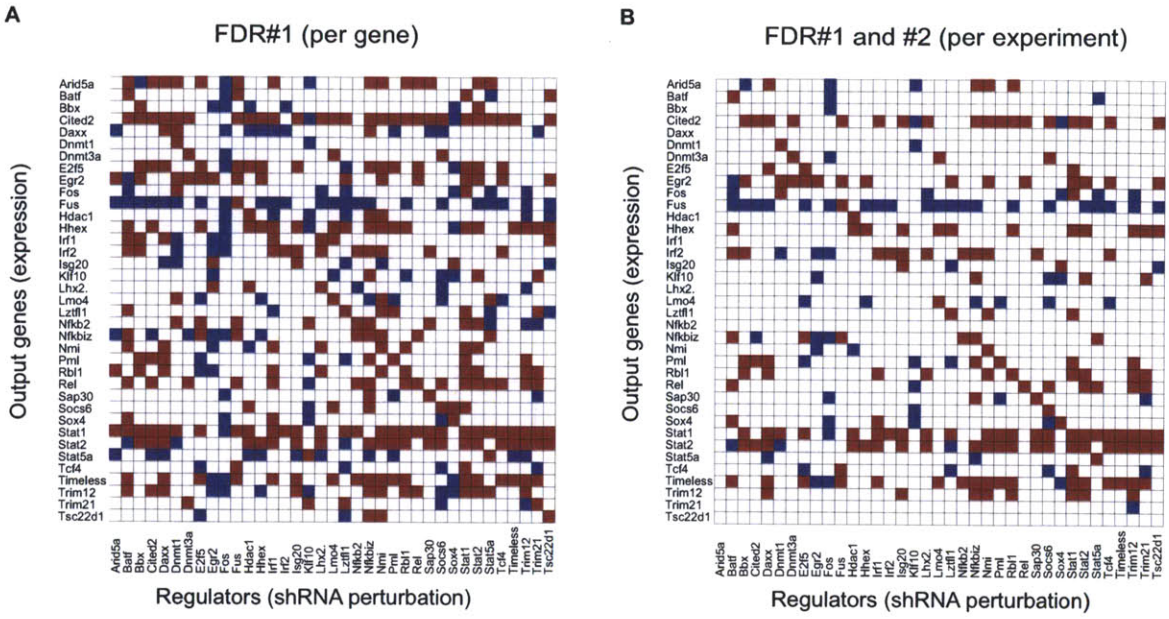

Figure 2-11: FDR values for 39 regulators that were also included in the 118 signature set. Shown are the FDR values for induction (red) and repression (blue) for each of the 39 regulators that were also measured as target genes (rows), in the 39 experiments that target these regulators with shRNA (columns). The rows and columns are sorted in the same order. (**A**) FDR#1 (gene-specific) correctly scores all knockdowns, except for two genes (Fos, Klf10) that are repressed by the response in wild-type cells and hence cannot be assessed. (**B**) FDR#1 followed by FDR#2 (shRNA-specific) correctly scored 80% of knockdowns (i.e., 20% of shRNAs targets were not called as significant due to effects on expression of control target genes).

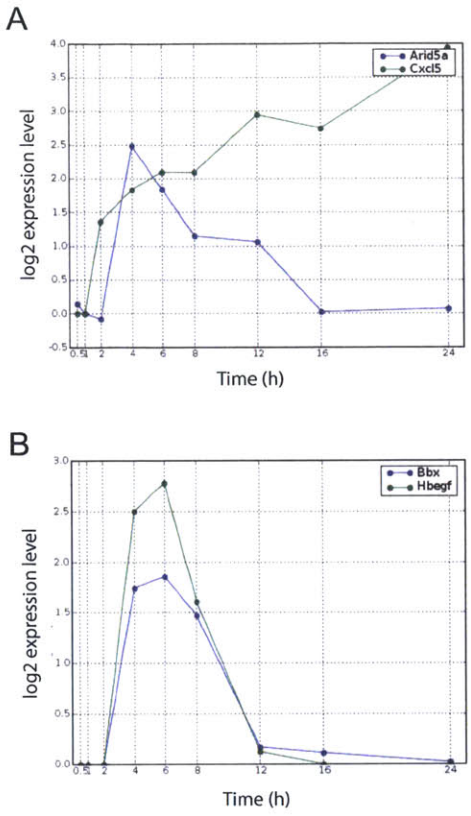

Figure 2-13: (A) Expression profiles of two genes whose interaction was identified in the perturbational model, with confidence value above 0.95, but that was not predicted in the observational model. (B) Expression profiles of two genes whose interaction was predicted by the observational model but not found in the perturbational model.

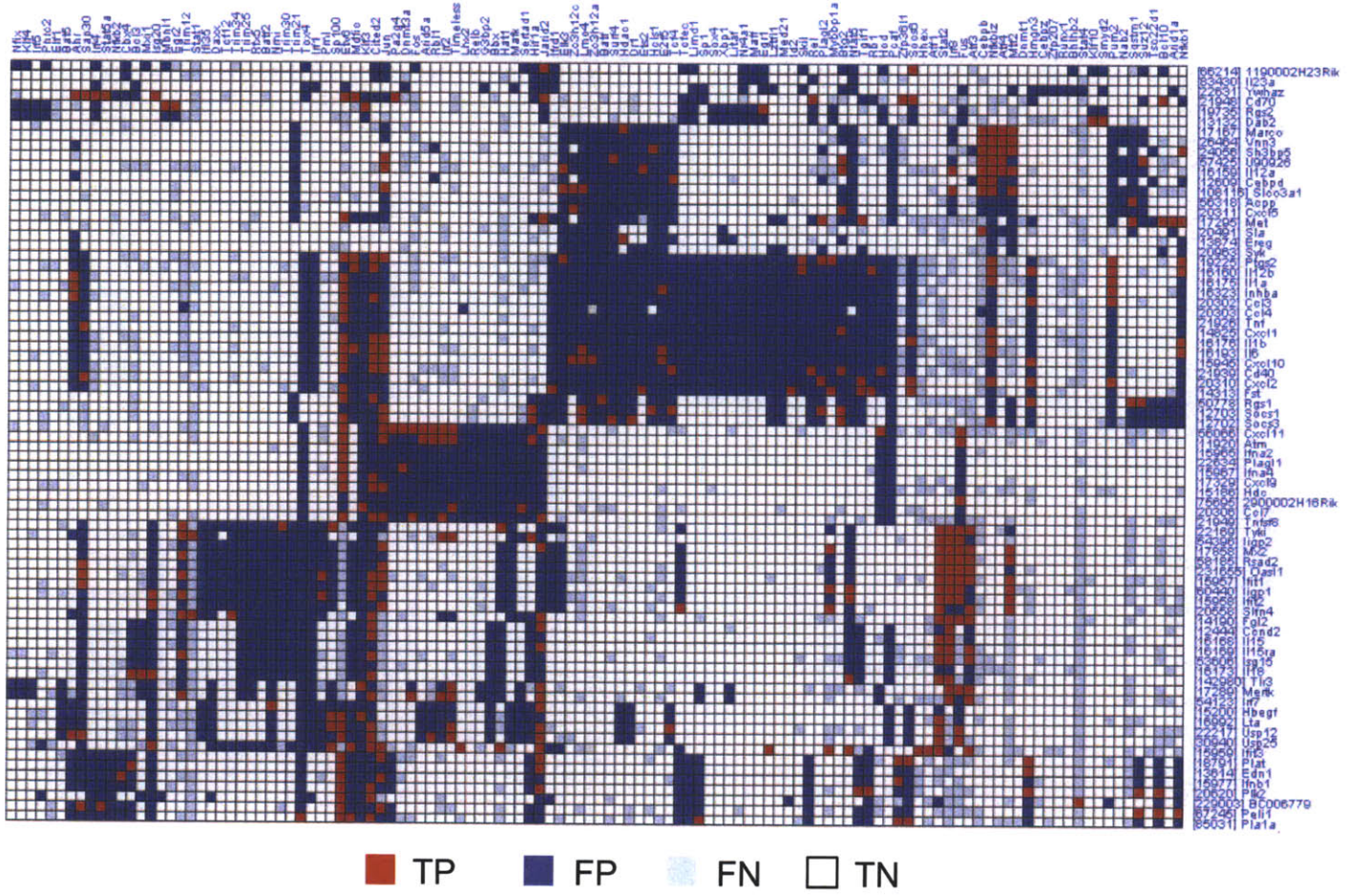

Figure 2-12: True-false positive and negative associations in the observational model.

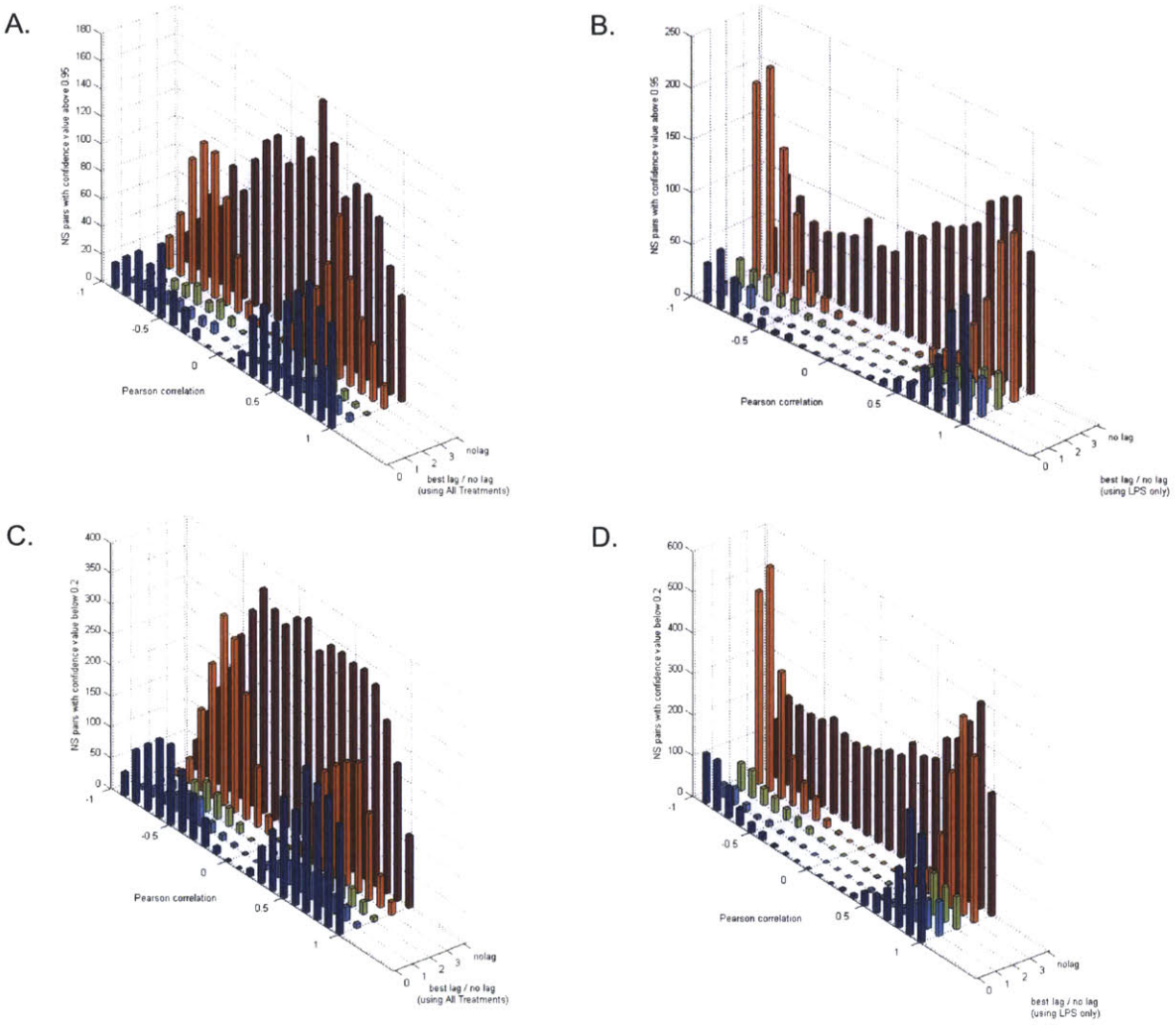

Figure 2-14: Correlation between expression of regulator and that of its target and non-target genes. Shown are distributions of Pearson correlation coefficients (dark red) and the best correlation coefficient when allowing a lag of up to 4 time points (3) (blue - no lag, light blue - 0.5 h lag, green - 1h lag, orange - 2 hour lag). The coefficients are calculated in the following way. **A.** between all confident (FDR#1 and FDR#2 > 0.95) regulator-target pairs, with expression data from all treatments; **B.** between all confident (FDR#1 and FDR#2 > 0.95) regulator-target pairs, with expression data from the LPS time course only; **C.** between all non-regulator-target pairs (FDR#1 and FDR#2 < 0.2), with expression data from all treatments; **D.** between all non- regulator-target pairs (FDR#1 and FDR#2 < 0.2), with expression data from the LPS time course only. The distributions for correct regulator-target pairs and false pairs are highly similar, indicating the high number of false positives in *trans*-models.

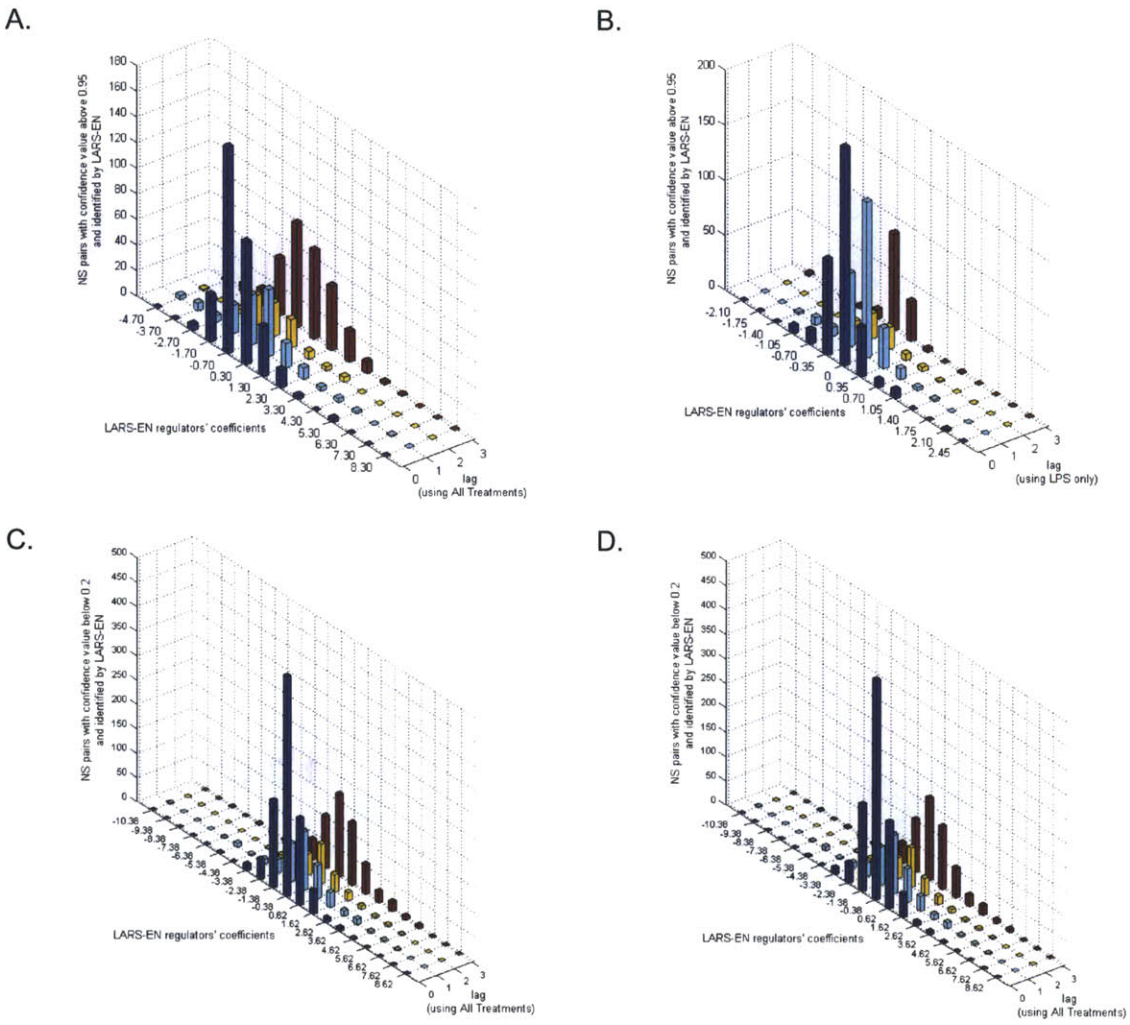

Figure 2-15: Regularized regression coefficients between expression of regulator and that of its target and non-target genes. Shown are distributions of the coefficients from the L2-regularized regressions chosen when allowing a lag of up to 4 time points (blue - no lag, light blue - 0.5 h lag, yellow - 1h lag, red - 2 hour lag). The coefficients are calculated in the following way. **A.** between all confident (FDR#1 and FDR#2 > 0.95) regulator-target pairs, with expression data from all treatments; **B.** between all confident (FDR#1 and FDR#2 > 0.95) regulator-target pairs, with expression data from the LPS time course only; **C.** between all non-regulator-target pairs (FDR#1 and FDR#2 < 0.2), with expression data from all treatments; **D.** between all non-regulator-target pairs (FDR#1 and FDR#2 < 0.2), with expression data from the LPS time course only. The distributions for correct regulator-target pairs and false pairs are highly similar, indicating the high number of false positives in *trans*-models.

Chapter 3

Systematic identification of topologically essential interactions in regulatory networks

Maxim N. Artyomov*, Ana Paula Leite*, Fadi Towfic, Aviv Regev

*These authors contributed equally to this work.

The work included in this chapter was adapted from the manuscript submitted for publication.

Chapter 3

Systematic identification of topologically essential interactions in regulatory networks

3.1 Abstract

Screens monitoring the effects of deletion, knock-down or overexpression of regulatory genes on the expression of their target genes are critical for deciphering the organization of complex regulatory networks. However, since perturbation assays cannot distinguish direct from indirect effects, the derived networks are significantly more complex than the true underlying one. Previous approaches to identify a minimal network topology consistent with the results of a perturbation screen only presented approximate methods with major limitations and are often applicable only to simple network topologies.

We present Exigo, an approach to systematically find a family of core networks for an input network of any topology with an arbitrary number of activating and inhibiting interactions. Using a novel matrix representation of the network topology we reduce the problem of identifying the core underlying networks to counting self-avoiding random walks on the original network. This systematic approach allows

us to globally analyze the network's topology to determine the functional effect of modifications such as edge removal.

Exigo outperforms previous approaches on simulated data, successfully uncovers the core network structure when applied to real networks derived from perturbation studies in mammals, and improves the performance of network inference methods, thus providing a valuable tool for accurate global analysis of gene regulatory networks. Exigo is available for download at `http://www.broadinstitute.org/regev/exigo`.

3.2 Background

Systematic perturbation of genes by genetic manipulation or RNAi is a major tool in functional genomics [69, 70]. When coupled to a readout measuring mRNA levels, perturbation screens allow us to decipher the functional relationship between a regulator and its targets, thus providing new insights in understanding the complexity of gene regulatory circuits [6, 33, 69, 71].

However, genetic perturbations alone cannot distinguish between direct molecular effects of a transcription factor on its targets and indirect effects through additional layers in the circuit. These indirect effects result in 'extra' edges in the network, rendering it more complex than the 'real' biological ones. Pruning such indirect edges is important for the interpretation of biological screens as well as for the development of effective network inference networks. In particular, the DREAM4 challenge [72] indicates that the best-performing inference methods were those that leveraged information from perturbation screens through various topology analysis approaches [40, 73].

Previous studies [34, 35, 74] have formulated the problem of the identification of indirect interactions as finding the sparsest (most parsimonious) network consistent with the experimental observation, where the experimental observation is represented by a signed interaction graph reflecting all the experimentally-identified interactions (many likely indirect). In principle, multiple networks may be consistent with a given experimental interaction graph (Fig. 3-1). We term all of these as experimentally

equivalent networks, to indicate that one cannot distinguish between them based on a single-gene perturbation experiment alone.

The identification of all the experimentally equivalent networks consistent with the result of a perturbation experiment is a significant theoretical and computational problem. Published studies [36, 38, 40, 74–76] presented only a few methods for finding the sparsest network, each suffering from inherent limitations. Pioneering work [34] showed how a graph-theoretical method of transitive reduction could be used to find the most parsimonious genetic network for acyclic networks. In this transitive reduction approach, edges whose effect can be recapitulated by alternative interaction paths are considered unnecessary and are iteratively removed. This approach has been extended to handle some higher-order topological effects [35] or to use associated experimental data, such as confidence measures [38, 40]. However, these procedures are limited due to either the nature of approximations made or their extreme computational intensiveness. On the one hand, most approaches fail to accurately account for global effects caused by edge removal, when a particular edge contributes to an indirect interaction between a distant pair of nodes. In an alternative approach, the TRANSWESD [40] algorithm accounts for this limitation, by the enumeration of all possible random walks on the network. While this guarantees that for every experimentally observed interaction there is a path in the core network that recapitulates its effect, it is computationally very intensive [40] and, thus, cannot scale for realistically large biological networks. Finally, all transitive reduction methods ignore the non-uniqueness of the most parsimonious graph: sequential edge removal only allows the identification of a single graph. Thus, to the best of our knowledge there is currently no systematic procedure to determine if two or more networks have the same outcome in a genetic perturbation experiment, and hence no formal way to assign networks to equivalence classes with respect to a set of experimental observations ('observability classes').

Here, we present Exigo, a comprehensive approach to find multiple core networks consistent with experimental observations. Exigo uses an adjacency matrix representation of interacting networks and introduces novel matrix transformations to uncover

the relation between different networks that can produce the same experimental observations. Rather than enumerating all possible paths on the network, Exigo leverages the fact that a signed accessibility graph results from a cumulative effect of all possible self-avoiding walks on the network. It then computes these efficiently to generate a reference matrix that encodes a representation of all possible connections for the entire class of experimentally equivalent networks. Using the reference matrix, Exigo directly checks if removal of any one or more matrix entries (i.e., one or more network edges) would change the observability class of the modified network.

We apply Exigo to analyze the network structure of both simulated and biologically derived regulatory networks, which contain activations, inhibitions and cycles, and demonstrate that the method is applicable to networks of any topological complexity. Furthermore, we incorporate our topological analysis module into a state-of-the-art network inference procedure [73] and show that performance is substantially improved based on DREAM4 benchmarks, surpassing the previous top performer.

3.3 Results

Exigo: a method to identify core networks consistent with experimental observations

A given network is consistent with that defined by a perturbation experiment if and only if every interaction that is topologically possible in one network is also possible in the other, either directly (through one edge) or indirectly (through a multi-edge directed path with the same composite effect of inhibition or activation). We term all the networks that are consistent with one perturbation experiment as belonging to one observability class.

Exigo identifies core networks consistent with an experimental observation by following four steps (Fig. 3-2 A). First, given an original (experimental) network, Exigo identifies all topologically possible interactions within the network, by computing a reference network (matrix), which encapsulates all the topologically possible interac-

tions - direct or indirect - between the nodes of the original graph. Specifically, the reference network has an edge between two nodes if and only if there is a path (direct or indirect) between these nodes in the original network. The reference network thus extends the notion of an accessibility graph as defined originally in [34]: it is defined for networks with both activating and inhibiting interactions and with arbitrary complex topological elements, such as feedback loops. In particular, networks belong to the same observability class only if they generate (converge to) the same reference matrix. Second, Exigo attempts to remove single entries in the experimental network, one at a time, to identify all the edges that can be individually removed, while still being able to generate (converge to) the same reference matrix. However, this does not mean that these individual non-essential edges may all be removed together. Specifically, some subsets of edges are degenerate, they can each be individually removed but they cannot be simultaneously removed (e.g., edges AC and BC in the network (I) in Fig. 3-2 A). Third, Exigo identifies a subset of individual non-essential edges contained within such degeneracies. Finally, Exigo simultaneously removes all the individual non-essential edges that are not contained within the degeneracies in order to obtain a reduced network that belongs to the same observability class. In addition, the remaining degeneracies are marked and reported.

Notably, for sufficiently complex networks, some of the edges that were considered as essential in the initial network become non-essential in the reduced network, due to the simplification of the global network structure. Thus, Exigo iteratively repeats steps 2-4 (i.e., removing the edges and testing convergence of the modified network to the original reference matrix derived from the experimental data, see Materials and methods) until all the individually non-essential edges that are not contained within degeneracies are removed and the reduced network cannot be further simplified.

Computation of the reference matrix

To construct the reference matrix (Fig. 3-2 B), we first represent the experimental network by its signed adjacency matrix, M, with rows corresponding to regulator nodes and columns to target nodes. Each entry $M^{(i,j)}$ is either -1 (inhibition), $+1$

(activation), or 0 (no interaction) depending on the functional interaction between regulator i and target node j. Due to the nature of perturbation experiments, where the regulator is removed genetically or by shRNAs, diagonal entries are not well defined, but as we show below should be set to zero.

Next, we use matrix multiplication to identify all paths ('walks') from a given regulator node i to a target node j in the experimental matrix. We note that the n^{th} power M^n of matrix M has all such paths of length n, with the sign of the entry corresponding to the effect of i on j through such paths. For example (Fig. 3-2 B), after squaring matrix M_1, entry $M_2^{(2,4)}$ will be -2, consistent with two paths of length 2 in the experimental network, namely BCD and BAD, each mediating an inhibiting interaction.

When considering biological networks, one excludes as unrealistic all paths that pass the same node more than once when connecting node i to node j, such that only self-avoiding walks are considered [77]. To achieve this, we replace the diagonal elements by zeros after each round of multiplication (Fig. 3-2 B). Since this violates the commutative property of multiplication, in order to proceed to the next order we always multiply the original matrix from the right as,

$$M_1 \times M_1 \overset{\text{0s on diagonal}}{\longrightarrow} M_2; M_2 \times M_1 \overset{\text{0s on diagonal}}{\longrightarrow} M_3; etc. \tag{3.1}$$

where M_1 is the matrix representing the original network and M_n describes all self-avoiding paths of length n within M_1. Naturally, n_{max} should not exceed the number of nodes in M_1 (the largest self-avoiding random walk would involve all nodes).

"Weighing and thresholding" procedure

Notably, until this point, the procedure outlined above is virtually identical to those previously described for self-avoiding walks [78]. Such procedures then follow with direct enumeration of self-avoiding walks. This is required to distinguish between self-avoiding walks and loops that end at previously visited nodes, but is computationally prohibitive, exponentially in the rank of the matrix.

We address this challenge by a "weighing and thresholding" procedure, where each matrix M_i is weighted by successively smaller weights. This procedure eliminates the contribution from loops to the reference matrix. We rely on the fact that for any interaction between two nodes, the shortest path that connects them is a self-avoiding path with no loops and, thus, it will determine the sign of the interaction, as long as shorter paths are weighted higher than longer paths. Thus, once the sign function is applied to a *weighted* sum of matrices M_i, only contributions from self-avoiding walks will remain.

The weighing factors w_i, which allow us to exclude loops from consideration, play a fundamentally different role than that of the various weighted measures previously introduced in studies of network topologies. For example, communicability betweenness [79] reflects the balance between information propagation along the short and long paths and serves as an effective parameter to tune the importance of the interaction length (also, this concept is applied to nodes). In Exigo the weights w_i do not depend on specific biological knowledge on the strength or confidence of individual interactions.

It is possible for two or more different indirect paths from one node to another to deliver contradictory signals (e.g. inhibitory vs. activating arms of an incoherent feed forward loop). Since their combined effect cannot be unequivocally identified on topological grounds alone, we assume that in cases of paths of different length, the input through the shortest path determines the overall effect on the node. To reflect this, we choose weights w_i as exponentially decreasing positive numbers, $w_1 = 1 >> w_2 >> w_3 >> \ldots >> 0$, thus ensuring that the shortest path always overpowers any combined contradictory effect of longer paths and hence determines the sign of the entry in M_{sum}. Similarly, for conflicting paths of the same length we assume that they cancel each other, since topologically both paths are of the same "strength". In general, competing paths of any length can be compared through the use of confidence measures for individual interactions and appropriate rules for calculating the confidence of multistep path.

Thus, the reference matrix is obtained through:

$$M_{sum} = \sum_{i=1}^{n_{max}} w_i M_i \tag{3.2}$$

where higher weights are used for shorter paths and n_{max} is found iteratively by following the convergence of the reference matrix. Upon convergence, Exigo discretizes all the values in M_{sum}, setting all positive values to $+1$, all negative values to -1 and keeping 0 values.

Complexity

The reference matrix is computed in polynomial time (in the number of nodes, or matrix rank), since the longest self-avoiding walk cannot be longer than number of nodes n in the network. The procedure's running time is $O(2n^4)$ (see Methods for details).

The reference matrix, $M_{reference}$, is akin to the *path matrix* of the original matrix, M_1, except that it is signed to represent negative or positive interactions. To compute the path matrix of a graph is equivalent to creating a data structure that contains reachability information about a graph. This similarity is explained as follows:

Let $G = G(V, E)$ be a directed graph with n vertices $v_1, v_2, \ldots, v_n$. The *path matrix* or *reachibility matrix* of G is the n-square matrix $P = (p_{ij})$ defined as follows:

$$p_{ij} = \begin{cases} 1, & \text{if there is a path from } v_i \text{ to } v_j \\ 0, & \text{otherwise} \end{cases}$$

Suppose now that there is a path from vertex v_i to vertex v_j in a graph G with n vertices. Then there must be a simple path from v_i to v_j when $v_i \neq v_j$, with length $n-1$ or less, or there must be a cycle from v_i to v_j when $v_i = v_j$, with length n or less. This means that there is a nonzero ij entry in the matrix

$$B_n = A + A^2 + A^3 + \cdots + A^n$$

where A is the adjacency matrix of G. To generate P, we replace the nonzero entries in B_n with 1.

In Exigo, we added a weighting scheme to this procedure as explained above to efficiently compute collective, global effects of all self-avoiding walks in the network, while bypassing the need for exact enumeration of self-avoiding walks of every length (the exponential complexity of the enumeration problem would be computationally prohibitive [80]). Exigo's reference matrix computation is a major improvement over the exponential cost of exhaustive enumeration. To confirm the practical accuracy of our procedure, we compared our procedure for reference matrix construction to the exact procedure based on "direct enumeration" of self-avoiding walks on a compendium of 10,000 random networks. Specifically, we have constructed random networks of various sizes ($n < 100$) by assigning randomly +1,-1 or 0 to every entry of adjacency matrix. Then, for every randomly constructed network we constructed all possible self-avoiding directed paths by enumerating all the possibilities. Based on the obtained compendium of these walk, we have constructed a reference matrix for every network, assigning +1 to indirect interactions when two nodes were connected activating shortest self-avoiding path; -1, when shortest self-avoiding path was repressing and 0 when there were no indirect paths connecting two nodes. We then compared the reference matrix constructed in such way to the one constructed through Exigo's reference matrix procedure. These two reference matrices were identical for all $10,000$ randomly constructed networks.

In 1960, S. Warshall described an algorithm which is more efficient than calculating the powers of the adjacency matrix [81]. It has a worst case complexity of $O(n^3)$ where n is the number of vertices of the graph. Warshall's algorithm consists in defining n-square Boolean matrices $P_0, P_1, \ldots, P_n$ as follows. Let $P_k[i,j]$ denote the ij entry of the matrix P_k, so that

$$P_k[i,j] = \begin{cases} 1, & \text{if there is a simple path from } v_i \text{ to } v_j \text{ which does not use any other} \\ & \text{vertices except possibly } v_1, v_2, \ldots, v_k \\ 0, & \text{otherwise} \end{cases}$$

That is,

$P_0[i,j] = 1$ if there is an edge from v_i to v_j

$P_1[i,j] = 1$ if there is a simple path from v_i to v_j which does not use any other vertices except possibly v_1

$P_2[i,j] = 1$ if there is a simple path from v_i to v_j which does not use any other vertices except possibly v_1 and v_2

And so on.

The first matrix is the adjacency matrix of G (i.e., $P_0 = A$), and since G has only n vertices, the last matrix is the path matrix of G (i.e., $P_n = P$). Warshall noted that $P_k[i,j] = 1$ can occur only if one of the following cases occurs:

- There is a simple path from v_i to v_j which does not use any other vertices except possibly $v_1, v_2, \ldots, v_{k-1}$; hence

$$P_{k-1}[i,j] = 1$$

- There is a simple path from v_i to v_k and a simple path from v_k to v_j where each simple path does not use any other vertices except possibly $v_1, v_2, \ldots, v_{k-1}$; hence

$$P_{k-1}[i,k] = 1 \quad \text{and} \quad P_{k-1}[k,j] = 1$$

Thus, the elements of P_k can be obtained by

$$P_k[i,j] = P_{k-1}[i,j] \vee (P_{k-1}[i,k] \wedge P_{k-1}[k,j])$$

where the logical operations $\vee$ (AND) and $\wedge$ (OR) are explained in Tables 3.1 and 3.2 below. In other words, each entry in the matrix P_k can be obtained by looking at only three entries in the matrix P_{k-1}.

While Warshall's algorithm is appealing, if we apply it to a graph that has both positive ($+1$) and negative (-1) entries, it fails in taking into account the existence of conflicting paths of the same length. In Exigo, we assume that they cancel each other, since topologically both paths would have the same "strength". This cumulative effect

$\wedge$	0	1
0	0	0
1	0	1

Table 3.1: AND

$\vee$	0	1
0	0	1
1	1	1

Table 3.2: OR

of all paths is taken into account by Exigo but not by Warshall's algorithm (Fig. 3-3).

Identification of individually essential and non-essential interactions

To identify all the edges in the original network (M_1) that are individually non-essential, we construct all networks M_1' that can be derived by removal of a single edge from M_1 (there are as many M_1' networks as there are edges in the original network M_1). We next test whether each M_1' converges to the original reference matrix: for each M_1' we find the reference matrix $M'_{reference}$ and compare it to the reference matrix $M_{reference}$ obtained from the original network M_1. As long as $M'_{reference} = M_{reference}$, the removed edge is deemed individually non-essential, since it can be removed without affecting the experimental observations. For edges identified as individually essential ($M'_{reference} \neq M_{reference}$), we can explicitly determine which direct or indirect interactions are lost upon removal of this edge, by comparing $M'_{reference}$ to $M_{reference}$.

Importantly, identification of all the individual non-essential interactions does not yet directly determine the core network consistent with the experimental results. Indeed, simultaneous removal of all the individual non-essential edges may result in a network inconsistent with the original one, due to higher-order network structures. For example, consider the network in Fig. 3-2 A. It has three non-essential individual edges (AC, BC and AD, network (III), Fig. 3-2A). However, two of these edges, AC and BC, have the same effect, and are thus degenerate (networks (IV) and (V), Fig. 3-2A): either one can be removed as long as the other is intact, but if both (or all three edges) are removed, the resulting network is no longer consistent with the data. When such degeneracies exist, there will be multiple core networks consistent

with a single experimental network. Thus, we next determine which edges can be simultaneously removed and which are part of different degeneracies.

Identification of non-essential edges embedded in degenerate topologies

A direct and exhaustive way to identify all degenerate interactions is by systematic removal of all possible pairs of interactions, all possible triplets, etc., comparing each time to the reference matrix. This brute force approach is akin to a power-series expansion classically employed in statistical physics. Unfortunately, this approach is computationally expensive. As a practical solution, we devised an alternative procedure that performs well in computational time for the relatively sparse biological regulatory networks.

Specifically, we use a stepwise, iterative procedure to identify the edges that cannot be removed simultaneously, by sequentially removing multiple edges and testing at each point the convergence to the original reference matrix (materials and methods). Once an edge removal breaks convergence, we retain that edge in the network. We then cyclically permute the order in which edges are removed and repeat the same procedure. In this way, we identify all the degeneracies that arise due to edge redundancy within strongly connected components. In the most general case, however, this approach does not guarantee the identification of all degenerate entries in more complex topologies, but rather only a subset of degenerate entries representing one subset of possible core networks. At this point, upon removal of a fraction of non-essential edges, the remaining network consists only of essential and non-essential-but-degenerate edges.

Breaking the degeneracies to identify a minimal core topology consistent with experimental data

In practice, it is often important to also find a single, specific, parsimonious graph that is most consistent with the experimental data. In fact, one can leverage experimental

confidence measures assigned to each interaction (edge) to solve this problem. In this case, the goal is to break the degeneracies such that only the most confident interactions are kept among the ones involved in the degenerate subgroup.

To this end, we devised a procedure that guarantees that interactions of lower confidence are removed first among the degenerate edges. First, we apply Exigo to identify all the non-essential entries. Given experimental confidence measures for each interaction, we list all topologically non-essential entries in order of increasing confidence. We then attempt to sequentially remove non-essential interactions each time testing for convergence to the original reference matrix. If convergence is maintained, we remove the interaction, otherwise we keep it. The sequential removal ensures that the most confident interactions within degenerate subgroups will be preferentially kept in the core network.

Exigo finds multiple non-essential interactions in a mammalian regulatory network

We applied Exigo to an experimentally measured regulatory network that controls the transcriptional response of mouse primary dendritic cells (DCs) to lipopolysaccharide (LPS) [6]. This network connects 144 regulators (each perturbed by shRNA knockdown) to 128 targets whose expression is measured under each perturbation. The confidence in each interaction is estimated by an FDR-based model [6].

First, we considered the sub-network composed of the 39 genes that have were both perturbed as regulators and measured as targets. This subnetwork comprised 253 interactions: 168 activatory and 85 inhibitory. Exigo identified 44 (16%) individually non-essential interactions in this network, 24 of which were contained in degeneracies (Fig. 3-4A). It removed the 20 non-essential non-degenerate edges and then used the remaining interactions' confidence measures to sequentially remove the least confident degenerate edges, keeping only the ones that would violate the equivalence to the experimental network. Overall, it removed 37 of the 44 interactions individually non-essential interactions without breaking the convergence to the original reference

matrix (Fig. 3-5). Thus, the resulting minimal core network that is most consistent with experimental observations contains 216 edges between 39 nodes.

To characterize the effect of edge removal on the network's topology, we studied the local connectivity patterns in the experimental (original) and pruned (core) networks. We analyzed the number of feed-forward loops of each of 8 possible canonical classes [63] in each of the two networks (Fig. 3-4 B, 3-6). Despite the fact that only a small fraction of edges ($\sim$ 10%) are non-essential, the number of feed-forward loops decreased by over 30% (from 464 to 310). Importantly, the relative proportions of different classes of loops remained the same (Figs. 3-4 B, 3-6).

Furthermore, we tested how the choice of non-essential interations identified by Exigo was related to TF-DNA binding data. To this end, we compared Exigo's result to a High-Throughput Chromatin ImmunoPrecipitation (HT-ChIP) dataset (unpublished data) that builds genome-wide binding maps for 29 transcriptions factors following stimulation of primary innate imune DCs stimulated with the pathogen component LPS. We focused in 8 of these transcription factors (Egr2, Fos, Irf1, Irf2, Nfkb2, Rel, Stat1, Stat2) for the comparison, since they are the ones also present in the perturbation network analysed with Exigo. We found out that 8 out of 9 non-essential interactions have no binding and that 5 out of 7 non-essential degenerate interactions have no binding. However, we also identified 49 out of 73 essential interactions with no binding; this may be due to the fact that intermediate genes are responsible for the causal effect observed in these 49 essential interactions in the perturbation screen.

A relatively small number of redundant edges between regulators might lead to large redundancies in the effects that the regulators exert on other (non-regulator) target genes. To study this, we next considered the complete experimental network (144 perturbed regulators, 128 targets) [6]. The adjacency matrix has size 233×233 (144 regulators + 128 targets - 39 both regulator and target = 233) and contains 1,774 non-zero entries. Exigo identified 725 individual non-essential interactions, 134 of which were contained in degeneracies. Using the confidence values associated with each interaction [6], Exigo further parsed the degenerate entries to retain only 1,112

topologically essential edges, thus removing 38% of the interactions found in the initial screen as topologically non-essential and hence possibly indirect.

Exigo can be used for the analysis of networks of any complex topology

Experimentally-determined networks, even small ones, typically include multiple, often entangled, feedback loops, but all previously published methods are limited in their ability to handle such structures. To illustrate this, we consider a small synthetic network of 10 nodes and 19 interactions (Fig. 3-7). Exigo uncovered the structure of the minimal solution, identifying five non-essential edges, two of which are in a degeneracy. Breaking this degeneracy leads to 2 possible minimal networks, each with 15 edges. In contrast, despite the small size of the network, none of the available methods - SOS Pruning [38], NET-SYNTHESIS [39] and TRANSWESD [40] correctly identified any of these topologies.

In fact, some of the edges determined as non-essential and removed by some of these alternative approaches cannot be reconstructed (as an indirect path) from the resulting core network, indicating a significant failure. Notably, only TRANSWESD (relaying on exhaustive enumeration) ensures that any removed interaction can still be recapitulated by an indirect path. However, not only TRANSWESD is computationally highly intensive (e.g., more than to 5 hours on an Intel Core 2 Quad CPU Q6700; 2.67 GHz for a 100-node network), it does not account for the length of these indirect paths. This leads to the counterintuitive situation where the shortest (indirect) path between two nodes A and B in the core network has an effect opposite to that of the removed direct edge between these nodes (e.g., upon removing an activating edge $A \rightarrow B$, the shortest indirect path between A and B is inhibitory). Conversely, in Exigo all of the experimentally observed interactions are always represented by shortest paths of the same sign.

To further evaluate the performance of our approach, we analyzed a compendium of biologically-motivated synthetic networks. First, we generated 60 'ground truth'

networks by sampling in each case a 100-node sub-network from the *S. cerevisiae* real regulatory network, using GeneNetWeaver [82] with settings corresponding to the DREAM4 challenge (Materials and methods). The sampled genetic networks had 285 edges on average (±8). Next, we used GeneNetWeaver [82] to generate a corresponding experimental perturbation network for each 'ground truth network' by simulating a set of deletion experiments on each of the nodes in each network. On average, the perturbation networks contained 275 edges (±35), only 125 (±14) of them true positives. This fraction is consistent with previously published results [40].

On average, Exigo identified 100 (±20 STDV) non-essential edges with 30 (±10 STDV) of them contained in different degeneracies. These non-essential edges consist primarily of false positive edges (Table 3.3 and Fig. 3-8). Thus, the reduced (pruned) networks contained on average only 67% of the edges in the perturbation network (183 ± 23 edges), while still maintaining the vast majority (91%, 114 ± 11 edges) of the 125 true positive edges (see Table 3.3 for precision-recall and F-score computations). Exigo has the highest F-score of all methods. Biologically, a good reconstruction algorithm should infer as many correct edges as possible, in addition to the criteria that most of the inferred edges should be correct, and the F-score represents a compromise between these two objectives. Notably, it is expected that due to true biological redundancy, not only false positives but also true positives will be found topologically non-essential, albeit in different proportions.

In contrast, when SOS Pruning [38], NET-SYNTHESIS [39] and TRANSWESD [40] were applied to the same compendium, the original experimental perturbation results could no longer be reconstructed from the resulting pruned networks (on average 50% of interactions for SOS Pruning, 17% for NET-SYNTHESIS and 12% for TRANSWESD were not reconstructable from pruned networks), thus indicating that too many edges, or incorrect edges, have been removed (Fig. 3-8 and Table 3.3).

Exigo-enhanced network inference outperforms state-of-the-art methods

Recent studies [72] have shown that information from perturbation screens can readily improve the performance of network inference methods. In fact, methods that incorporate topological analysis of perturbation data have performed best ([40, 73] - first and third place, respectively) in the recent DREAM4 competition [72] for network inference methods.

To test whether Exigo's topological analysis can improve such network inference methods, we combined the confidence values from Pinna *et al.*'s [73], state-of-the-art network inference algorithm with Exigo (Materials and methods). Briefly, we use the same input confidence matrix as Pinna *et al.* and apply Exigo to a thresholded confidence matrix. Then, the entries found to be topologically essential are upweighted by its confidence value (see Materials and methods for details) and the confidence matrix is normalized. We tested the performance of the hybrid method on five DREAM4 benchmark networks, and found that it substantially improved performance compared to previous state-of-the-art network inference methods: 75.413 points against 71.589 points from Pinna *et al.* (for AUROC and AUPR values of DREAM4 benchmark networks see Table 3.4). This improvement was observed for all values of threshold parameter required by Pinna *et al.* [73] method, indicating that Exigo's topological analysis contributes novel valuable information.

Notably, while Exigo's approach can enhance network inference methods, it is not designed for this purpose. Indeed, there is a fundamental difference between identifying the minimal topology most consistent with perturbation data (as Exigo aims to do) and the more general problem of network inference. In particular, Exigo's goal is to identify the set of essential edges that must be preserved across all possible topologies. It thus provides a valuable additional input that can be used by various inference algorithms, and improve the quality of networks that they reconstruct.

3.4 Discussion

We present Exigo, an approach to systematically identify interactions that are topologically unnecessary, find a core network that is consistent with a given perturbation experiment and characterize the relative importance of topologically essential edges. Exigo relies on comparing a reference matrix derived from the original network and with those derived from networks where one edge is missing. Thus, it explicitly lists the indirect interactions that depend on the removed interaction.

Exigo is a global approach applicable to full networks that, unlike most other published methods [36, 38–40], accounts for the indirect effects that the removal of an edge between one pair of nodes can have on the interaction between another pair of nodes. For example, existing approaches that rely on transitive reduction [34] would consider 109 of 253 of the edges in the mammalian network above as candidates for removal, since removing the edge from node i to j is compensated by indirect interactions between i and j. Yet, only 41 edges are found to be non-essential if we consider the global effect of edge removal - in the remaining cases, there always exists a seemingly unrelated pair of nodes k and l whose interaction is destroyed by removal of edge from i to j. This 'global entanglement' illustrates that a naïve approach to transitive edge removal is unrealistic. Furthermore, when constructing the most parsimonious network in the above example, only 17 edges out of 41 can be simultaneously removed, while 24 edges are contained in different degeneracies and there are multiple different solutions that can be constructed by removing degenerate edges in different combinations. Conversely, exhaustive enumeration of all possibilities, as in the TRANSWESD algorithm [40], is computationally intensive, and does not scale well for realistic biological networks. Thus, Exigo strikes an effective balance between global analysis and computational limitations, and substantially improves over existing approaches. In particular, its complexity is polynomial, it relies exclusively on matrix-based operations (that can be efficiently implemented in e.g., MATLAB, Fig. 3-9) and it avoids any direct paths enumeration techniques. Exigo's current computational time is 1-2 minutes (on an Intel Core 2 Quad CPU; 2.4 GHz) for networks

of 100-300 nodes, a realistic scale for experimental perturbation screens. Analysis of a 1000-node network takes on average 155 minutes (on the same platform as above). Finally, as perturbation screens of thousands of genes (followed by signature profiling, as in [83]) are now becoming possible, Exigo's approach is fully parallelizable, as analysis of each edge can be done independently. It thus can be scaled in principle to genome-size networks. By comparison, a full version of TRANSWESD algorithm takes several hours for networks of 100-nodes, while their approximate version takes less than a minute, much like Exigo's exact version.

Furthermore, while we allow indirect interactions of any length to contribute to the reference matrix, in some circumstances one might prefer considering only paths shorter than a certain length (for instance, if longer indirect interactions are biologically unrealistic). In this case, the procedure for finding the reference matrix can be modified by restricting the number of terms that contribute to the reference matrix in Eq. 3.2. Finally, Exigo can be extended beyond purely topological considerations by explicitly considering the confidence weights associated with each interaction. Specifically, given a "composition" rule for determining the confidence of a multi-step path from the confidences associated with its constituent edges, the convergence operation can be modified for the space of matrices that have signed confidence values as individual entries. In this case, the reference matrix would be obtained by thresholding the resulting matrix that contains confidence measures for all direct and indirect interactions.

Topological analysis can substantially enhance network inference algorithms [72]. To demonstrate Exigo's utility for this purpose, we have integrated it as a 'topological analysis module' within a state-of-the-art network inference algorithm used in the DREAM4 challenge [73]. The substantial enhancement in performance gained by the addition of Exigo illustrates its utility. Future efforts can further enhance such integration, for example by introducing novel confidence-based metrics for identifying the most parsimonious graphs or through comparative analysis of different possible parsimonious networks.

When applying Exigo, it is important to keep some limitations in mind. The core

networks identified by Exigo are not necessarily the most parsimonious networks: a group of individually essential edges may still be removable together without perturbing the experimental observability class of the whole network, even though each edge is individually essential. In its present form, Exigo does not probe such removal of multiple essential edges, due to computational complexity. However, the same converging transformation can be used to evaluate if such removal leads to a network of a different observability class. In this sense, Exigo provides a systematic approach to critical for the identification of parsimonious networks, analogous to power-series based approaches for studying critical phenomena in statistical physics [84].

In conclusion, we present a novel systematic approach for analysis and interpretation of large scale gene regulatory networks derived from gene perturbation screens. Our approach is robust, flexible and computationally efficient, and presents a substantial and demonstrable improvement to previous methods. Thus it can assist in the interpretation and follow-up of biological experiments and provides a systematic framework for topological analysis and comparison of large gene regulatory networks. While exact construction of the most parsimonious network for a given experimental observation remains an unsolved mathematical problem, our approach provides a direct and systematic path for studying the effects of higher order topological effects (such as degenerate structures) on the structure of the core network.

3.5 Materials and methods

Exigo

There are four major steps in Exigo: for a given input network, (1) Exigo computes the reference matrix by applying a convergence transformation. This reference matrix describes all possible (direct or indirect) interactions with their appropriate signs. (2) Exigo finds all individual non-essential interactions by comparing the reference matrix of the input network and those of networks where individual interactions have been removed. (3) Exigo identifies edges contained in various degeneracies and breaks the

degenerate structure of the solution by leveraging experimental information on the confidence of each interaction. Exigo overall procedure is the following:

Input: A matrix M, with regulator genes in rows and target genes in columns.
Output: A core matrix $coreM$, with the identification of individual non-essential entries and degenerate entries.

1. Find the reference matrix $M_{reference}$ for M using equations 3.1 and 3.2
2. Identify all individual non-essential edges

   for each edge (i, j)
   - create M' from M by setting $M(i, j)$ to zero
   - compute $M'_{reference}$, the reference matrix for M'
   - compare $M'_{reference}$ to $M_{reference}$

     if $M'_{reference} = M_{reference}$
     edge (i, j) is non-essential
     else
     edge (i, j) is essential
     end if

   end for
3. Identify non-essential edges embedded in degenerate topologies

   3.1. order all non-essential individual edges in an arbitrary but fixed order

   3.2. start with the network that has the first edge removed and remove the next edge in the ordered list

   3.3. compute the reference matrix of the network with two removed edges, $M'_{reference}$, and compare it to $M_{reference}$

   if $M'_{reference} = M_{reference}$
   - the two edges are non-essential even if they are removed simultaneously and are not identified as degenerate edges
   - remove the three first edges (the former two and an additional one) in the ordered list and repeat the convergence test of step 3.3

   else
   - the second edge in the ordered list is degenerate
   - keep the degenerate edge intact while proceeding to the next edge and removing it

   end if

   3.4. repeat steps 3.2 to 3.3 (removing second, third, etc., edge at each iteration) until going through the list of all individual non-essential edges

   3.5. change the list order by placing the last degenerate entry of the list at the beginning and keeping the rest of the sequence unchanged

3.6. consider this modified list as your new ordered list and repeat steps 3.2 to 3.5

4. Break degeneracies by using confidence values

   4.1. Arrange all degenerate edges in the order of increasing confidence

   4.2. Make matrix with the least confident edge removed

   4.3. Remove next edge in the ordered list

   4.4. Check if reference matrix of modified network is equal to the original one

   if equal
   go to the step 4.3
   else
   recover last removed edge and go to step 4.3
   end if

Identification of non-essential edges embedded in degenerate topologies

We use a stepwise, iterative procedure to identify the edges that cannot be removed simultaneously: **(1)** We order all the non-essential individual edges in an arbitrary but fixed order. **(2)** We start with the network that has the first edge removed (note that removal of only the first edge is guaranteed not to change the observability class of the network because each edge has already been determined to be individually non-essential). We then remove the next edge in the ordered list, use the convergence transformation to compute the reference matrix of the network with two removed edges, and compare it to the reference matrix of the original network. **(3)** If the perturbed network converges to the original reference matrix, the two edges are non-essential even if they are removed simultaneously and are not identified as degenerate edges, at least at this point, and we proceed to remove the three first edges (the former two and an additional one) in our ordered list and repeat the convergence test of step (2). **(4)** If, however, removal of the edges breaks convergence to original reference matrix, we identify the second edge in the ordered list as degenerate and keep this edge intact while proceeding to the next edge and removing it. **(5)** We then iterate steps (2)-(4) until we go through the complete list of all individually

non-essential edges. **(6)** Once we have reached all edges in the list, we change the list order by placing the last degenerate entry of the list at the beginning and keeping the rest of the sequence unchanged; we then repeat procedures (2)-(5) considering this modified list as our ordered list from step (1). Overall, we perform this procedure no more than n times, where n is the number of edges in the network, which is also the number of possible cyclic permutations of initial arbitrary ordered list. Fig. 3-10 shows an example of this procedure for the network (I) shown on Fig. 3-2 A.

Complexity Analysis

Reference Matrix computation Complexity. The algorithm continuously multiplies the adjacency matrix (adjusting the diagonal to 0 at every step to avoid self-walks) until either the matrix converges (i.e., $Mref_{t-1} = Mref_t$) or matrix has been multiplied by itself $2n$ times (where n is the number of rows/columns in the adjacency matrix). Thus, given that matrix multiplication takes $O(n^3)$ time, and this operation is repeated $2n$ times in the worst case, the complexity of the reference matrix computation is $O(2n \times n^3) = O(2n^4)$.

Core-finding algorithm Complexity. The algorithm first removes all edges in the network one by one to determine edges whose removal results in a change in the reference matrix (such edges are dubbed "essential" edges). The complexity of the algorithm at this step is $O(e2n^4)$ where e is the number of edges and n is the number of nodes in the adjacency matrix. Next, the algorithm goes through each of the found non-essential edges and finds the sets of degenerate edges (i.e., cannot remove all edges at the same time and preserve the reference matrix). The complexity of this step is $O(e_{nonessential}2n^4)$. Next, the algorithm finds the most parsimonious set of edges to preserve. Briefly, this step involves looping through the set of non-degenerate edges, removing edges one by one, then restarting until each non-degenerate edge has been removed at least once with respect to all other non- degenerate edges in the set. The complexity of this step is $O(e^2_{nonessential}2n^4)$. Thus, the total complexity of this algorithm is $O(e2n^4 + e_{nonessential}2n^4 + e^2_{nonessential}2n^4)$.

Degeneracy-breaking algorithm Complexity. The algorithm first removes all

edges in the network one by one to determine edges whose removal results in a change in the reference matrix (such edges are dubbed "essential" edges). The complexity of the algorithm at this step is $O(e2n^4)$ where e is the number of edges and n is the number of nodes in the adjacency matrix. Next, the algorithm steps through all non-essential edges in a sorted manner (according to each edges confidence value) and removes (without replacement) non-essential edges that do not change the reference matrix in a degenerate manner. The complexity of this step is $O(e_{nonessential}2n^4)$ where $e_{nonessential}$ is the number of non-essential edges found using the previous step. Thus, the overall complexity of the algorithm is $O(e2n^4 + e_{nonessential}2n^4)$.

Exigo implementation and test datasets

We provide a website (`http://www.broadinstitute.org/regev/exigo/`) with executable scripts that take networks specified in matrix format (perturbed regulators in rows) and outputs network analysis that includes the list of individually non-essential edges, list of edges affected by individually essential edges and the list of degenerate entries on the website. Additionally, on this website one can find documentation with a practical application of Exigo on all the networks used in our work: DC networks (sizes 39×39 and 155×128), small (10 nodes) synthetic network and sixty large (100 nodes) synthetic networks.

Chimeric network inference method

To combine Exigo within the inference method of Pinna *et al.* [73], we started with the same confidence matrix as Pinna *et al.* (the Z-score normalized raw perturbation matrix WZR) and the same threshold ($t = 2$), as the one used in the DREAM challenge (see Table S3 for threshold variations). We set all entries in WZR to 0, where $|WZR| < t$ and took the sign of the resulting matrix, M, such that all entries greater than 0 were set to 1 and all entries less than 0 were set to -1 in M. We applied the degeneracy breaking scheme of Exigo to M to obtain a matrix E. Finally, we set all non-zero entries in E to their respective confidence values from the WZR,

and normalized this confidence matrix by the maximum absolute value, according to the recommendation for DREAM evaluation. The final confidence output from the degeneracy breaking matrix E and Pinna *et al.*'s downRank matrix DR was the arithmetic mean of the entries in both matrices ($ChimericConfidence_{ij} = (DR_{ij} + E_{ij})/2$).

Software and datasets used

GeneNetWeaver. We used GeneNetWeaver [82] to sample 60 100-node synthetic networks and simulate mRNA levels following knock-out of each of the 100 transcription factors in each of those networks. We have produced knock-out time-series and recorded mRNA concentration at 400 s timepoint, assigning a confidence level to each genetic interaction using z-scores, akin to those used when handling real perturbation data [4, 6]: we computed z-scores for each interaction across all the knock-outs and, thus, found corresponding p-values to observe the specific mRNA level upon gene knock-down. To construct the perturbation graph we recorded only interactions with a p-value less than 0.05.

SOS Pruning, NET-SYNTHESIS and TRANSWESD. Based on the code available in the supplement of [38], we applied the SOS (save our signs) pruning procedure to several synthetic networks. We also downloaded NET-SYNTHESIS from `http://www.cs.uic.edu/~dasgupta/network-synthesis/` and applied the algorithm of transitive reduction to perturbation graphs. For TRANSWESD, we used version 9.9 of CellNetAnalyzer downloaded from `http://www.mpi-magdeburg.mpg.de/projects/cna/cna.html`.

Mammalian networks. The interaction matrix for the dendritic cell network was constructed based on Fig. S14B of [6]. The expanded interactions matrix and confidence values were taken from Tables S6 and S7 of [6].

3.6 Figures and tables

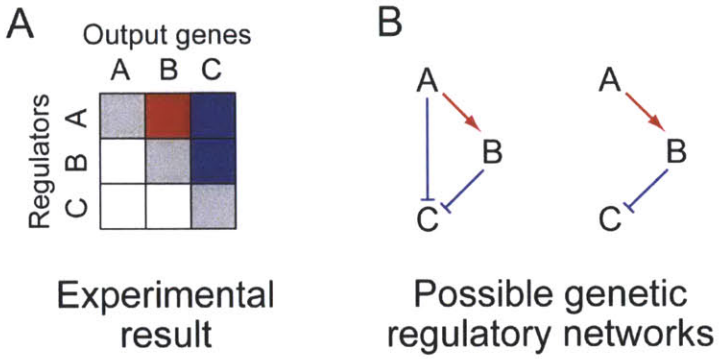

Figure 3-1: Experimental perturbation networks. Shown are a typical output of a gene perturbation screen **(A)** and two alternative networks **(B)**, either of which can produce the same experimental perturbation result. Red and blue square, respectively: perturbation of the regulator decreases or increases the expression of the output gene. Red arrow activation; blue blunt edge repression.

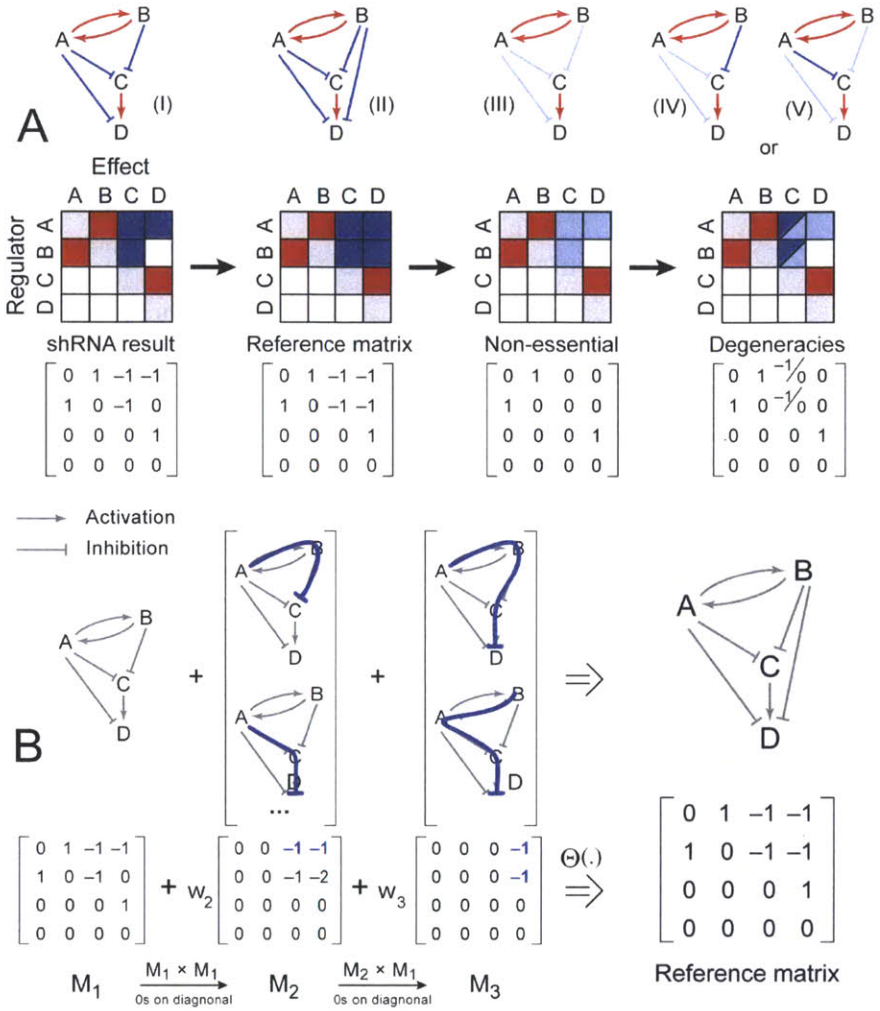

Figure 3-2: Exigo. (A) The sequence of steps in Exigo: given an experimental perturbation network (I), Exigo derives a reference matrix (II) which includes all possible (direct or indirect) interactions. It then identifies all individual non-essential interactions (III), followed by determining which edges are contained in degeneracies (IV). In each panel, shown are a network view (top), and the corresponding matrix in either color code (middle) or numerical (bottom) form. Red (blue) arrow/cell: activating/repressing interaction. Solid colors: essential interaction; faded color: inessential/removed interaction. In bottom matrix: 1 activation, -1 inhibition, 0 no interaction. **(B)** Deriving the reference matrix. Shown is the process of deriving a reference matrix by applying a convergence transformation to the network (I) from panel A. The original network I is shown on the top left, along with its corresponding adjacency matrix (left, bottom), M_1. The reference matrix (right, grey shading) is obtained as a weighted and thresholded sum of matrices representing self-avoiding random walks of different length (M_i). Each matrix can be constructed iteratively from M_1 by using the rules shown on the figure and described in detail in the text. Bold blue paths on the networks above M_2 and M_3 show indirect interactions of length 2 and 3, respectively that are computed in the transformation and contribute to the reference matrix. $\Theta(\cdot)$ designates a thresholding function theta that is applied to each element of the matrix obtained by summation of all indirect-effect matrices M_i in order to obtain the reference matrix.

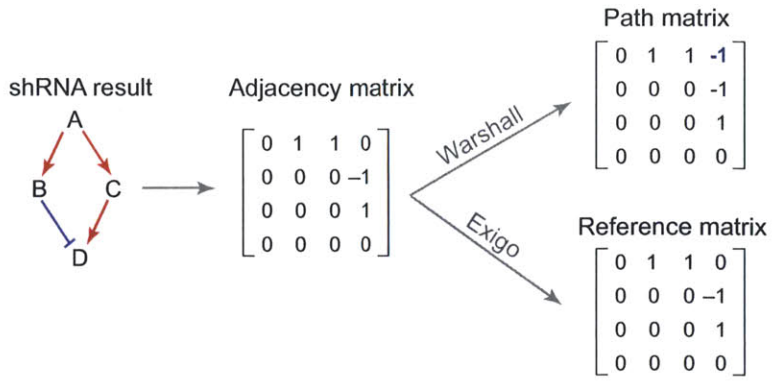

Figure 3-3: Comparison between the path matrix and the reference matrix generated by Warshall's and Exigo algorithms, respectively. Red arrow - activation; blue blunt edge - repression.

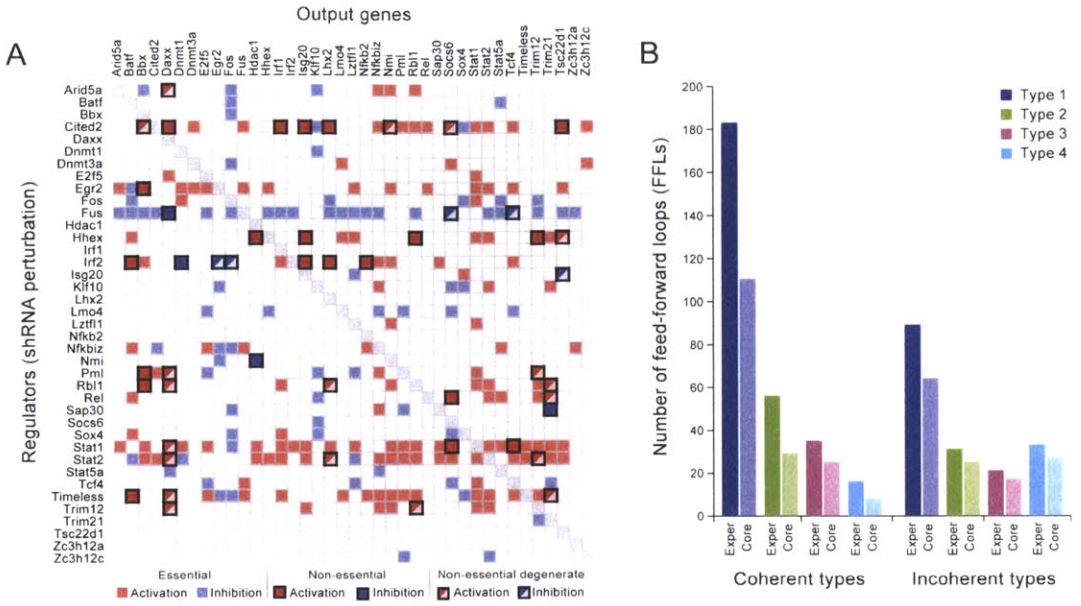

Figure 3-4: Application of Exigo to a mammalian network. (A) Results of applying Exigo to the regulatory network controlling activation of dendritic cells in response to LPS. Shown is a 39×39 experimental perturbation network (rows: perturbed regulators, columns: expression targets; red: activation; blue: inhibition; white: no interaction). Bold black borders encapsulate interactions that have been found individually non-essential (degenerate or not), and hatched entries indicate the ones that are contained in degeneracies. **(B)** Distribution of coherent and incoherent feedforward loops identified in the input experimental network (exper, dark left bar) and in the core network after confidence-based degeneracy breaking (core, light right bar). Note that the distribution remains virtually the same while the absolute number of loops decreases substantially.

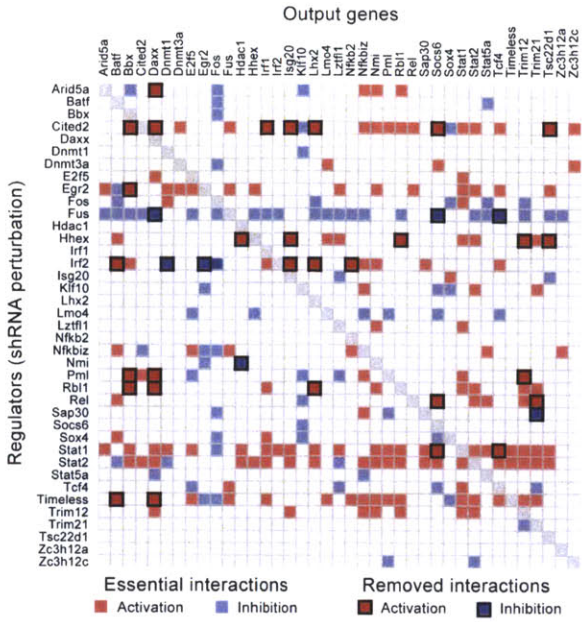

Figure 3-5: The Exigo procedure consists of two major steps: (1) identification of all individually non-essential interactions and finding degenerate edges among individually non-essential interactions; and (2) finding a minimal core network by breaking degeneracies based on the experimentally measured confidence values. Here we show results of applying Exigo to the regulatory network controlling activation of dendritic cells in response to LPS after breaking degenerate entries by the use of interaction confidence values (Note that Fig. 3-4 of the main text shows the network before degeneracy breaking, i.e., after step 1 only). Shown is a 39×39 experimental perturbation network (rows: perturbed regulators, columns: expression targets; red: activation; blue: inhibition; white: no interaction). Bold black borders encapsulate interactions that have been removed to produce the final core network.

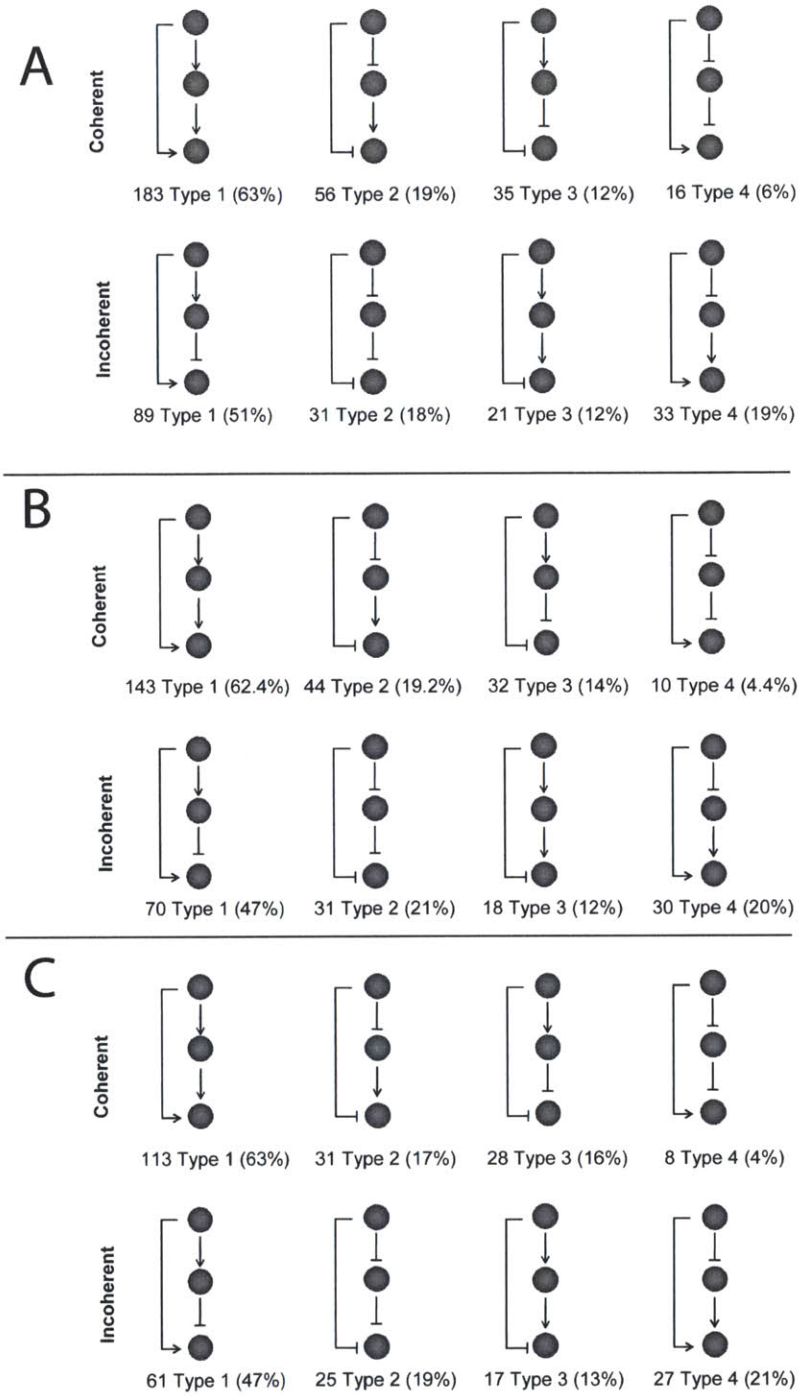

Figure 3-6: Number of coherent and incoherent feedforward loops of different types identified in the input network (A) and in the core network before (B) and after (C) confidence-based degeneracy breaking. In each case, shown are the eight possible classes of loops, with an illustrative example (arrows, activation; blunt arrows, inhibition) on top, and their number and fraction at the bottom.

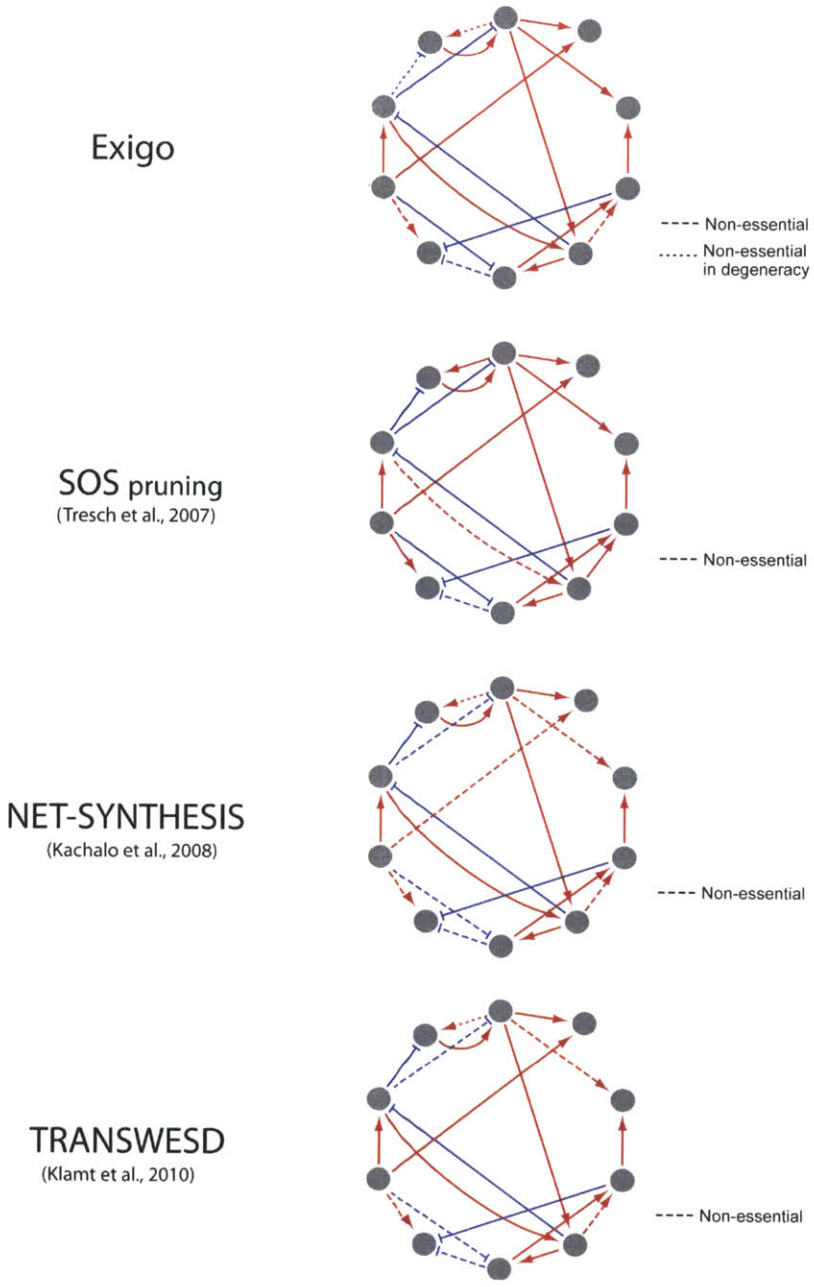

Figure 3-7: Performance of different network analysis methods (Exigo (top), SOS Pruning [38] (middle top), NET-SYNTHESIS [39] (middle bottom) and TRANSWESD [40] (bottom)) on the same synthetic network of 10 edges. Red arrows: activating interactions, blue blunt arrows: repressing interactions. Shaded arrows: interactions that were found non-essential by the indicated method. Both SOS Pruning and NET-SYNTHESIS incorrectly remove edges that are essential for observing the experimental screening results. TRANSWESD removes interactions such that interactions reconstructed based on the core network actually have opposite signs to the original ones: for example, upon removal of inhibiting interaction $E \dashv G$, the shortest path indirect interaction is actually an activating one. Similarly, when removing $E \rightarrow F$, the reconstructed interaction is actually inhibiting.

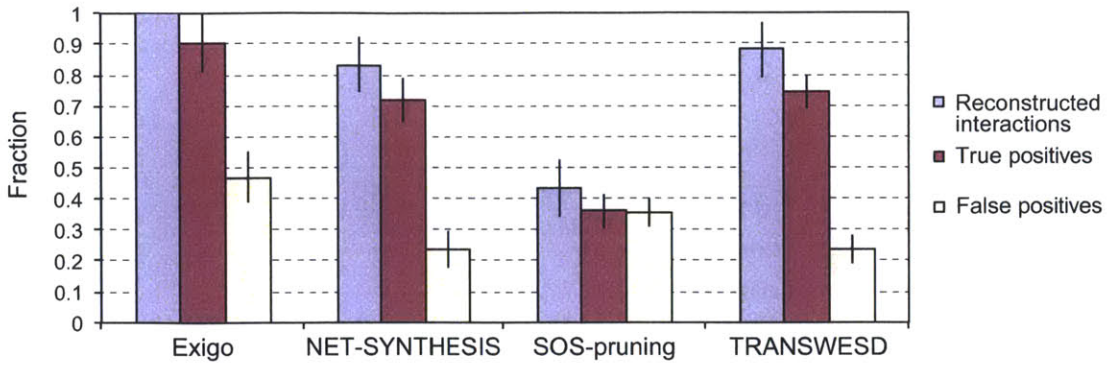

Figure 3-8: Comparison of the performance of different methods on simulated experimental networks derived from a compendium of 60 synthetic 'ground truth' networks (100 nodes each). For each method, shown are the fraction of experimental edges explained by the core networks produced (blue), the fraction of true positive edges (based on the ground truth network) that are retained in the core networks (dark red) and the fraction of false positive interactions in the core networks (yellow). By definition, Exigo always explains all of the available experimental observations. It also outperforms in the fraction of true positives retained. Note that some experimental edges are not explained by core networks produced by SOS Pruning and NET-SYNTHESIS. This means that neither direct nor indirect interactions on the core networks produced by SOS Pruning or NET- SYNTHESIS can recapitulate the effect of some of the experimental edges indicating that inconsistency have been introduced during the parsing process. Note that Exigo core network is always consistent with experimental network (see also Table 3.3 and Fig. 3-7).

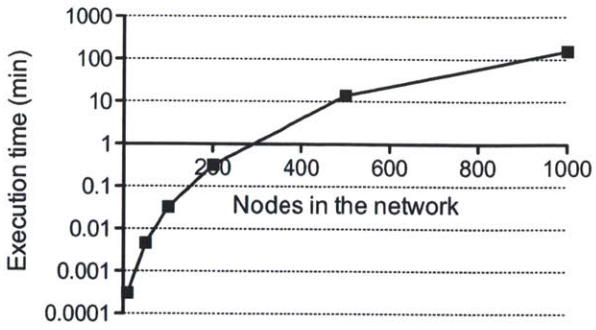

Figure 3-9: Timing results (Intel Core 2 Quad CPU; 2.4 GHz) for applying MATLAB coded degeneracy-breaking version of Exigo to networks of various size.

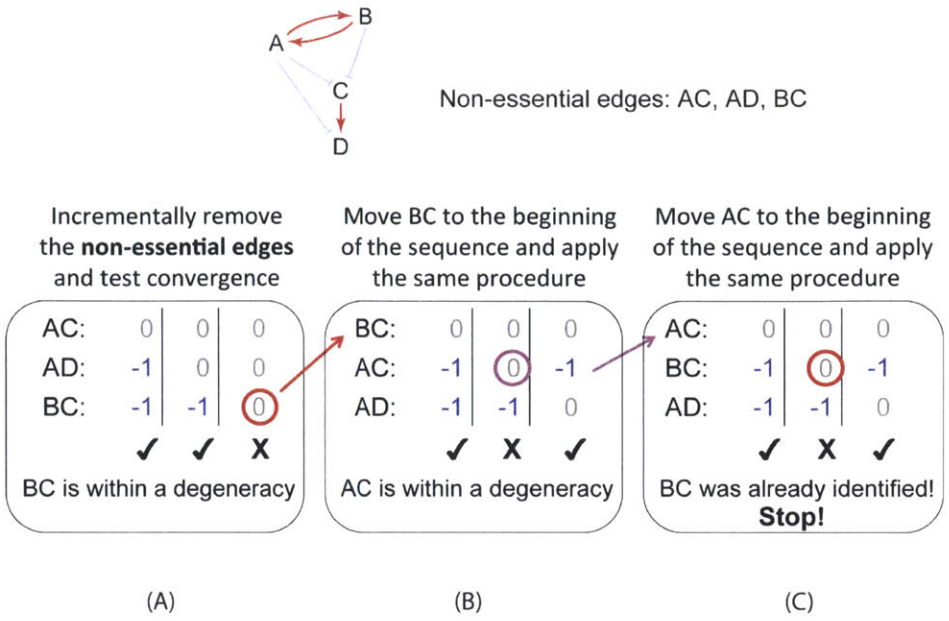

Figure 3-10: Illustration of the mechanism to identify degenerate edges for the network described in Fig. 2 of the main text (Materials and methods). First (A), the non-essential individual edges (AC, AD, BC) are ordered arbitrarily and then this order is fixed. The first edge is then removed, which is indicated by *0* in panel (A). Removal of the first edge (AC) does not change the observability class of the network which is indicated by the check- mark below the column on the panel (A). Next, we remove the two first edges in the ordered list (AC and AD), as indicated by 0 in the second column in panel (A). We find that removal of the first two edges does not alter the observability class and thus put a check-mark below this column and proceed to remove also the third non-essential edge (thus, AC, AD and BC are removed). In this situation we find that observability class changes indicating that BC, while being individually non-essential, is contained within a degeneracy. We next (B) modify the order of edges to put BC atop of the list and repeat the procedure described above to find that AC is contained within a degeneracy and modify the list order to the one shown on panel (C). Finally (C), we find edge BC as the degenerate one, which has already been identified and, thus, we stop the procedure.

		Average	Std. Deviation	Median
True network	Total edges	285.25	7.56	288.00
Perturbation (experimental) network	Total edges (TP same sign + TP opp signs)	274.63 (125.15+2.45)	35.02 (13.97+1.37)	276 (126+2)
	Precision	0.468	0.045	0.464
	Recall	0.448	0.055	0.444
	F-score	0.456	0.042	0.448
NET-SYNTHESIS (Kachalo *et al.*, 2008)	Total edges (TP same sign + TP opp signs)	124.65 (89.55+0.7)	10.87 (8.77+0.91)	124 (90+0.5)
	Unexplained Experimental Interactions	46.00	45.98	46.12
	Precision	0.725	0.054	0.715
	Recall	0.317	0.033	0.317
	F-score	0.440	0.039	0.440
SOS pruning (Tresch *et al.*, 2007)	Total edges (TP same sign + TP opp signs)	97.92 (44.15+0.85)	12.97 (6.59+0.82)	98 (43+1)
	Unexplained Experimental Interactions	156.98	21.49	157.00
	Precision	0.462	0.058	0.457
	Recall	0.158	0.026	0.153
	F-score	0.235	0.033	0.229
TRANSWESD (Klamt *et al.*, 2010)	Total edges (TP same sign + TP opp signs)	130.58 (93.67+0.75)	10.09 (9.09+0.93)	129 (94.5+1)
	Unexplained Experimental Interactions	33.77	22.67	28.00
	Precision	0.724	0.059	0.722
	Recall	0.331	0.035	0.335
	F-score	0.454	0.042	0.461
EXIGO Core Network (before degeneracy breaking)	Total edges (TP same sign + TP opp signs)	201.63 (115.95+1.73)	29.20 (12.41+1.30)	206.5 (117+1)
	Unexplained Experimental Interactions	**0.00**	0.00	0.00
	Precision	0.590	0.065	0.583
	Recall	0.413	0.050	0.418
	F-score	**0.483**	0.042	0.479
EXIGO Core Network (after degeneracy breaking)	Total edges (TP same sign + TP opp signs)	183.92 (113.93+1.28)	25.73 (11.91+1.14)	186.5 (114+1)
	Unexplained Experimental Interactions	**0.00**	0.00	0.00
	Precision	0.632	0.066	0.627
	Recall	0.405	0.048	0.409
	F-score	**0.491**	0.043	0.485

Table 3.3: Statistics for 60 synthetic 100-node networks used in this study. True networks row corresponds to the networks sampled from yeast regulatory networks by GeneNetWeaver; perturbation network row corresponds to the experimental networks constructed based on the kinetic simulations ran by GeneNetWeaver (page 89); subsequent rows correspond to the networks that result from application of Exigo and other methods to the perturbation networks. Additional row in the table describes the number of experimental edges that cannot be reconstructed from core networks produced by each method. A non-zero number of non-reconstructable experimental interactions means that this number of interactions is reconstructed incorrectly from core networks obtained by the pruning methods (SOS Pruning, NET-SYNTHESIS and TRANSWESD) indicating that inconsistency have been introduced during the parsing process. Specifically, in case of TRANSWESD, it means that incorrectly reconstructed interactions have opposite signs to the real ones. This is due to the failure of TRANSWESD to account for shorter indirect interactions of opposite signs when identifying the core network (a simple example of such situation is provided on Fig. 3-7). SOS Pruning and NET-SYNTHESIS fail to reconstruct the experimental network for the same reason as TRANSWESD and also simply due to excessive removal of essential interactions (again, a simple example of such situation is provided in Fig. 3-7). Exigo core network is always consistent with the experimental network.

	Pinna *et al.* (DREAM4) + Exigo	Pinna *et al.* (DREAM4 best)	Team 296 (second place)	Team 515 (third place)
Score	**75.413**	**71.589**	**71.297**	**64.715**
AUPR				
Network 1	0.63684	0.53607	0.512	0.49
Network 2	0.37775	0.37711	0.396	0.327
Network 3	0.40014	0.38983	0.38	0.326
Network 4	0.3851	0.34943	0.372	0.4
Network 5	0.21966	0.21332	0.178	0.159
AUROC				
Network 1	0.91481	0.91363	0.908	0.87
Network 2	0.80147	0.80149	0.797	0.773
Network 3	0.8331	0.83304	0.829	0.844
Network 4	0.84228	0.84163	0.844	0.827
Network 5	0.75936	0.75916	0.763	0.758

Table 3.4: Comparison of network inference algorithms on benchmark networks of the DREAM4 challenge. The 'chimeric algorithm' that includes the topological analysis module of Exigo outperforms other state-of-the-art network inference methods. Total score, AUPR and AUROC are computed using DREAM4 evaluation scripts.

	Downrank ($t = 2$) + Exigo	DownRank (DREAM4 Settings)	DownRank ($t = 2.5$)	DownRank ($t = 2.5$) + Exigo
Score	75.413	71.589	79.279	81.191

Table 3.5: Performance of chimeric network inference method shown for two threshold choices: $t = 2$ corresponds to the threshold used by Pinna *et al.* in the DREAM4 competition and $t = 2.5$ corresponds to the optimal threshold for DREAM4 benchmark networks. In both cases information added by topological analysis with Exigo improves performance.

Chapter 4

Inferring transcriptional and microRNA-mediated regulatory programs in Diffuse Large B-cell Lymphoma

Ana Paula Leite, Stefano Monti, Kunihiko Takeyama, Bjoern Chapay, Jun Lu, Todd R. Golub, Margaret Shipp, Aviv Regev

Experiments in this chapter were performed by Kunihiko Takeyama, Bjoern Chapay, and Jun Lu in the Golub and Shipp laboratories.

Chapter 4

Inferring transcriptional and microRNA-mediated regulatory programs in Diffuse Large B-cell Lymphoma

4.1 Abstract

Diffuse Large B-Cell Lymphoma (DLBCL) is the most common lymphoid malignancy in adulthood. Although some genetic abnormalities have been related to the pathogenesis of this disease, the full identity of dysregulated cellular pathways is still unknown. A system-level dissection of regulatory mechanisms using heterogeneous datasets can help identify, among other things, new previously uncharacterized regulatory elements and the molecular pathways they target and whose disruption might lead to tumorigenesis. However, integrating different data sources brings many challenges, since they may have different quality and informativity. We used Module Networks to identify modules of co-regulated genes and integrate datasets from multiple high-throughput assays: gene expression micro-arrays, DNA copy-number SNP arrays, and microRNA arrays. Our analysis associates miR152 and CD63 with survival,

links MYCBP and COPS5 with a module of genes enriched for oxidative phosphorylation and mitochondrial functions, and CREBL2 with a glycolysis module. This analysis expands the knowledge about causal and combinatorial relationships that characterize molecular signatures in DLBCL and, generally, provides a systematic approach for the integration and analysis of different types of datasets.

4.2 Background

Understanding the molecular mechanisms that drive tumorigenesis is a fundamental question in biomedical research. Genomics research in the past decade has shown that cancers harbor unique transcriptional signatures that distinguish them from other cancer types and sub-types and from normal tissue, and that have tremendous diagnostic and prognostic value [85]. More recently, large-scale projects, such as The Cancer Genome Atlas (TCGA), are also characterizing the cancer genome for large-scale genetic aberrations and coding mutations [47]. However, in most cases, we lack a genome-scale understanding of the mechanisms that 'translate' these genetic aberrations to transcriptional changes, and the role that transcription factors, miRNAs and other regulatory genes play in this process. For this reason, there has been an increased interest in approaches that provide a systems-level view of the components and the properties of the system. To this end, integrating data from different sources has become an important part of research in genomics [86].

In the past few years, some algorithms have been developed to integrate paired gene expression profiles and somatic copy number alterations (CNAs) information on the same patients (reviewed in [87]). CNAs, which are differences in the number of copies of multi-kilobase segments of the genome, are a major source of genetic diversity, with several variants now conclusively linked to human disease. While we were working on this project, Akavia *et al.* [16] published CONEXIC, a Bayesian algorithm that identifies candidate driver genes in cancer and links them to gene expression signatures they govern by integrating copy number and gene expression. CONEXIC is inspired by the Module Networks algorithm [3] and uses a score-guided

search to identify the combination of modulators that best explains the behavior of a gene expression module across tumor samples and searches for those with the highest score within the amplified or deleted regions. The algorithm was applied to data from melanoma cell lines, where it correctly identified known drivers and connected them to their known targets. Integrative analysis of CNAs and gene expression has also been used to identify tumor subtypes or patient groups that have different characteristics including patient survival, and response or resistance to therapy [88, 89]. In addition, other computational approaches have been proposed that identify miRNA regulatory modules by integrating expression profiles of miRNAs and mRNAs (e.g. [90, 91]). To our knowledge, very few studies have integrated CNA, miRNA and gene expression profiles simultaneously. An example of such integrative analysis is proposed in [92], where non-negative matrix factorization-based clustering [93] of mRNA expression data was used to identify molecular subgroups of medulloblastoma; DNA copy number, miRNA profiles, and clinical outcomes were then analyzed for each. Here, we use these heterogeneous datasets to learn an integrative module network that advances the molecular characterization of Diffuse Large B-cell Lymphoma (DLBCL).

DLBCL is clinically, morphologically and genetically a heterogeneous group of malignant proliferations of large lymphoid B cells, and accounts for 30 to 40% of newly diagnosed lymphomas [94]. Based on expression profiling studies, primary DLBCLs have been categorized into several tumor subtypes [95], including B-cell receptor/proliferation (BCR), oxidative phosphorylation (OxPhos) and host response (HR) [96]. The first subset, BCR, is enriched for genes involved in cell-cycle regulation, DNA repair, cell division and B-cell receptor signaling, and is associated with BCL6 gene rearrangements. The OxPhos signature includes genes involved in oxidative phosphorylation, suppression of apoptosis, and mitochondrial and proteasome function, and it is more frequently associated with the translocation t(14;18), affecting the antiapoptotic gene BCL2. Finally, the HR signature is enriched for genes involved in T-cell receptor and cytokine signaling, natural killer cell activation, dendritic cell maturation and chemotaxis. Studies of the diversity and clinical impact of the BCR and HR signatures have made the transition from transcriptional to

functional analysis, but little is still known for the OxPhos subset.

Our integrative analysis provides an unbiased way for identifying drivers that best account for the behaviour of sets of genes. We use Module Networks to infer modules of expression data and then learn regulatory programs for each module, consisting of regulator genes, miRNAs and CNAs (Fig. 4-1). The method identified 100 modules associated with 231 regulator genes (40 of which are in amplified or deleted regions), 108 miRNAs and 39 CNAs. Novel observations include the identification of CD63 and miR152 as biomarkers for predicting survival, a possible role of MYCBP and COPS5 in oxidative phosphorylation and mitochondrial functions, and the association of CREBL2 with glycolysis. Our model raises concrete testable hypotheses which can be tested by perturbation experiments, helping to advance the characterization of DLBCL and its subtypes.

4.3 Learning of a Module Network for DLBCL

We employed the Module Networks algorithm [3] to learn a network model for DLBCL that aims to explain changes in gene expression in tumors by underlying molecular and genetic changes. This model associates modules of co-expressed genes with regulatory programs that use a combination of regulator' profiles (genetic aberrations, regulator genes and microRNA expression) to predict these expression changes. It raises concrete testable hypotheses which can be tested by perturbation experiments.

Module Networks is a probabilistic graphical model [11] that automatically infers modules of co-expressed genes and their shared regulatory programs. A regulatory program uses the expression level of a set of regulators to predict the condition-dependent mean expression of the genes in a module. The algorithm uses an iterative learning procedure using the Expectation Maximization (EM) algorithm. Each iteration consists of two steps: an E-step and an M-step. In the M-step the procedure is given a partition of the genes into modules and learns the best regulatory program (as a regression tree) for each module. In the E-step, given the inferred regulatory programs, it re-assigns each gene tothe module that best predicts the gene's behavior

(it does not assign a regulator gene to a module in which it is also a regulatory input, directly or indirectly). The target function is the Bayesian score, derived from the posterior probability of the model (see [97] for a detailed description of the algorithm). The regulatory program is chosen from a pre-defined set of candidate regulators.

In this study we used three different types of datasets to build a comprehensive map for DLBCL. From the publicly available microarray-based transcription dataset of 176 primary DLBCLs generated using Affymetrix U133A and U133B platforms [96], we used 110 of these samples for which paired mRNA and miRNA profiles were available. After robust multichip average (RMA) processing, using Affymetrix' absolute call data, genes with "present" calls in fewer than 80% of arrays were discarded (eliminating 17,215 [77%] of 22,283 probe sets) so reducing genes whose expression were largely in the noise range. We further eliminated probe sets with no Entrez gene identifier and averaged probes mapping to the same gene. This resulted in a final dataset with 3,716 genes. Regarding the miRNA data, which was generated with Luminex, we used a list of 155 miRNAs. The chromosomal copy number alterations were collected from the HD SNP arrays of sixty-two of the 176 primary DLBCL genomic DNAs, as described in [98].

We initially considered three strategies to integrate these heterogeneous datasets: (1) search for one Module Network model whose regulatory programs were chosen from all types of features simultaneously; (2) search for separate Module Network models for each feature type; (3) search for one Module Network for one feature type and then learn extra regulatory programs for each module using the other feature types. However, the first strategy did not work well, since the measurements were obtained by different technologies and there was a consistent bias towards choosing regulators from the same data type of the genes in the modules. We found the second strategy very unstable for an integrative framework analysis, as the three module networks learned (one for each data type) generated different modules of genes. We decided to pursue the third strategy, which overcomes these limitations.

First, our approach consisted in applying the Module Networks procedure on the mRNA dataset, which consisted of 110 samples and 3,716 genes. This list included

397 genes with a regulatory potential (transcription factors and signaling molecules) that were used as input candidate regulators for Module Networks. We initialized the algorithm systematically from 10 to 250 modules (in increments of 10), and chose the model whose Bayesian score was 85% of the best score (Fig. 4-2). The chosen model consisted of 100 modules (Fig. 4-3). These modules have a variable number of genes, with a median size of 35 genes. Twenty-five percent of the clusters have less than 22 genes and 75% have less than 50 genes. The global view of the modules is presented in Fig. 4-4 and was generated by calculating the average expression over all genes in each module. Furthermore, for each of these 100 modules, we learned another two regulatory programs: one using all 155 miRNAs as candidate regulators and another one using 45 CNAs discrete profiles. These profiles, determined by GISTIC [99] according to a defined amplitude threshold (Fig. 4-5), are discrete, so that a '0' indicates that the copy number of the sample was not amplified or deleted, a '1' indicates that the sample had low-level copy number aberrations, and a '2' indicates that the sample had high-level copy number aberrations.

4.4 Modules are enriched for specific processes and functions

In order to identify the potential functional role of assigned regulators, we inspected the gene set enrichment of the target modules (using the Genomica software). It is expected that modules are likely enriched with genes whose expression is biologically affected by the regulators. Thus, the processes represented by genes in a module may suggest how a regulator gene, miRNA or CNA change the cellular physiology and contribute to the oncogenic phenotype. We collected functional sets of genes from GO [100] and MSigDB (C2-CGP: curated chemical and genetic perturbations) [101] and predicted miRNA target genes sets from MSigDB (C3.MIR) [101] and TargetScan [102]. TarBase [103] provided a manually curated collection of experimentally tested miRNA targets in human. In addition, we used the hematopoietic gene expression

differentiation map (DMAP) [12] to provide us sets of genes that were significantly over- or under-expressed across lineages in hematopoiesis (and hence relevant to DL-BCL), and gene sets associated with human ES cell identity collected from microarray studies [104], since human ES signatures are often related to poor prognosis in tumors.

A total of 78 modules have at least one functional category overrepresented at the 0.05 significance level (FDR corrected P-values). In total, 550 different categories are overrepresented, some of which are shown in Fig. 4-6. For example, modules 782, 800, 848 and 1087 are mostly enriched for oxidative phosphorylation and mitochondria related genes (P-values 1.96×10^{-11} and 8.21×10^{-7}, respectively), while modules 794 and 1177 are statistically enriched for cell cycle and DNA replication genes (P-values 1.23×10^{-5} and 4.62×10^{-8}, respectively). Modules 1129 and 1207 are enriched for immune response genes (P-values 4.19×10^{-12} and 1.77×10^{-7}, respectively), module 1123 for inflammatory response genes (P-value 3.37×10^{-5}), and modules 963 and 999 for extracellular matrix genes (P-value 1.96×10^{-11} and 8.21×10^{-7}, respectively). Several modules are statistically enriched for other functional categories like ribosome (modules 695, 1041, and 1053, with P-value 2.37×10^{-6}, 1.13×10^{-20} and 6.44×10^{-50}, respectively), splicing (module 1117, with P-value 2.09×10^{-6}), endoplasmatic reticulum (module 926, with P-value 2.04×10^{-4}) and glycolysis (module 969, P-value 4.19×10^{-5}).

4.5 Regulatory programs overview

Of the set of candidate regulators, 231 out of 397 were found to regulate at least one module in the inferred network, 40 of which are in amplified or deleted regions. Furthermore, the model associated 108 miRNAs with the modules and 39 CNAs. In particular, 57 regulator genes, 35 miRNAs and 23 CNAs were chosen as top regulators (Fig. 4-7). Among these sets, COPS5, miR-30e-50 and deletion peak in region 2q33.3 were the most frequent choices (15, 13 and 33 times, respectively).

4.6 Potential regulators with a role in DLBCL

To show the approach's ability to reproduce diverse features of regulatory programs, we discuss some of the key modules identified.

Inflammatory response module

The inflammatory response module (1123, fig. 4-8) is inducted by CEBP-β, a gene important in the regulation of genes involved in immune and inflammatory responses; in particular, it has been shown to bind to the IL-1 response element in the IL-6 gene, as well as to regulatory regions of several acute-phase and cytokine genes. CEBP-β is a crucial regulator of hematopoiesis, it belongs to the family of basic leucine zipper (bZIP) transcription factors, and its main function has been implicated to be in control of myeloid differentiation [105]. On the other hand, the inferred miRNA regulatory program specifies miR-223 as the module's top (activating) regulator. miR-223 is a hematopoietic specific microRNA with crucial functions in myeloid lineage development and it has been shown to target CEBP-β mRNA 3' UTR [106]. The choice of these top regulators is consistent with the module's enrichment for genes induced in the granulocyte/monocyte progenitor (GMP) and common myeloid progenitor (CMP) clades (P-values 4.96×10^{-9} and 1.33×10^{-7}, respectively), as well as in monocyte and granulocyte lineages (P-values 1.21×10^{-11} and 2.19×10^{-4}, respectively).

Cell Cycle and DNA replication modules

One of the fundamental traits of cancer cells is their ability to sustain chronic proliferation. While normal tissues control the production and release of growth-promoting signals that instruct entry into and progression through the cell growth-and-division cycle, these signals are deregulated in cancer cells. In DLBCL, the"BCR/proliferation" cluster identified in [96] had abundant expression of cell-cycle regulatory genes. Similarly, module 1177 from our model, is enriched for cell cycle and DNA replication genes (P-values 4.62×10^{-8} and 5.58×10^{-7}). In agreement with this finding, the module is induced by RNASEH2A (ribonuclease H2, subunit A), a gene that en-

codes a protein that participates in DNA replication, and by PBK (a.k.a. TOPK), a mitotic kinase that enhances Cdk1/cyclin B1-dependent phosphorylation of PRC1 and promotes cytokinesis. The miRNA regulatory program identifies miR-20a and miR-106b in the first and third levels of the decision tree, respectively. Ivanovska *et al.* [107] have shown that miR-106b family members contribute to tumor cell proliferation in part by regulating cell cycle progression and by modulating checkpoint functions. Furthermore, miR-20a and miR-106b (together with miR-17) are known to act in concert to modulate E2F activity on cell cycle arrest during neuronal lineage differentiation of unrestricted somatic stem cells from human cord blood (USSC) [108]. The second level regulator of this module predicted by the model is miR-26b that, even though not implicated in the previous study, has been shown to cooperate with their host genes to block the G1/S-phase transition by synergistically activating the pRb protein [109]. These findings suggest that miR-20a, miR-26b and miR-106b may act in concern to regulate cell cycle in DLBCL.

Also associated with these biological functions is module 794. It is primarily regulated by RACGAP1, a gene known to be up-regulated in B cell lymphoma tumors expressing an activated form of MYC [110], and by miR-30e-5p, whose induction represses the module genes' activity.

Extracellular matrix module

Extracellular matrix (ECM) components have been implicated in tumor growth, progression, and metastasis in lymphoid malignancies [111]. Our model associated NBL1 and miR-152 with modules enriched for ECM-related genes, cell adhesion and angiogenesis. NBL1 is a transcription factor that may function as an inhibitor or repressor in cell growth and/or maintenance, and plays a role in the negative regulation of the cell cycle.

Although strategies aiming at inducing modifications in the ECM components would be hard to conceive, given the high redundancy of the cellular dynamics leading to ECM regulation, ECM-related cues (such as miRNAs) should be taken into account as potential cancer-related markers for prognosis [111].

Oxidative phosphorylation and glycolysis modules

While an oxidative phosphorylation subtype has been previously identified in DLBCL, the regulatory mechanisms associated with this cancer subtype are still unknown. Our model associates module 782, enriched for oxidative phosphorylation and mitochondrial functions, with regulation by the factors MYCBP and COPS5 (Fig. 4-9). These two factors were previously associated with survival and oxidative metabolism in breast cancer [112, 113], suggesting a possibly general mechanism. In fact, in [112], it is found that the mitochondrial signature in breast cancer is induced by overexpression of MYC or MYC plus COPS5, but not COPS5 alone. MYC is a proto-oncogene that encodes a transcription factor involved in apoptosis, proliferation and the overall regulation of hematopoietic homeostasis [114]. It has been identified in several studies as highly prognostic in DLBCL [115–119] and its over-expression is often associated with malignant transformations. The MYCBP gene encodes a protein that binds to the N-terminal region of MYC and stimulates the activation of E box-dependent transcription by MYC. A second level regulator in module 782 is RAP1B, a small GTPase, and the module itself includes gene RAP1A. Members of the RAS-like small GTP-binding protein superfamily are known to regulate multiple cellular processes including cell adhesion and growth and differentiation, and one study has actually reported the role of this gene in the amelioration of high glucoseinduced injury and normalization of mitochondrial functions [120]. While COPS5 (a.k.a. CSN5) was selected as top regulator of module 782, COPS4, another subunit of the COP9 (constitutive photomorphogenic) signalosome (CSN), is a member of the module. The CSN complex is composed of eight subunits and is a highly conserved protein complex that is known to regulate processes such as cell cycle progression and kinase signaling.

Some cancer cells preferentially metabolize glucose through aerobic glycolysis, a phenomenon known as the Warburg effect [121]. Module 969 associates CREBL2 (cAMP responsive element binding protein-like 2) with target genes enriched for glycolysis (e.g. ALDOA, GPI and PKM2) and for up-regulated marker genes that dis-

tinguish DLBCL from follicular lymphoma (FL) samples, namely HSPCB, ALDOA, PKM2 and HMGA1. Our finding associates high expression of CREBL2 with the repression of genes with a glycolytic function. In addition, CREB3L2 shows in the third level of the regulation program. These findings suggest that genes encoding proteins of the CREB-family may be involved in regulating glycolysis. Interestingly, miRNAs can mediate fine-tuning of the cancer-associated glycolytic pathways either directly or at the level of oncogenes [122]. Our model associates let-7c, a miRNA known to be important in cell growth, with the glycolysis module, regulated by CREBL2.

4.7 Regulators associated with survival and "consensus cluster"

Our integrative analysis allows to systematically search for genes associated with survival. Given that the regulatory program is described by a decision tree, we could look at each first split (root) and assay the probability of survival for patients on each side of it. These two survival curves are compared using a log-rank test of the null hypothesis of a common survival curve. This strategy for finding association with an external phenotype can provide powerful results without the need of testing every gene assayed. In total, six modules show a first split in the regulatory program that is statistically significant with survival (FDR ≤ 0.25, Table 4.1).

Worth noting, is the association of poor prognosis in DLBCL patients with high expression of CD63 (Fig. 4-10). The protein encoded by this gene mediates signal transduction events that play a role in the regulation of cell development, activation, growth and motility. It is a cell surface glycoprotein that is known to complex with integrins and that may function as a blood platelet activation marker. Interestingly, CD63 has already been shown to be a biomarker for predicting the prognosis in earlier stage of adenocarcinomas of lung [123]. Module 1075, regulated by CD63 is enriched for immune response genes. Interestingly, is also enriched for up-regulated genes in B-CLL (B-cell chronic leukemia) patients expressing high levels of CD38 and ZAP70,

which are associated with poor survival.

In addition, module 716 strongly associates BCL6 expression with a better prognosis, a finding that has been reported in the literature [124]. BCL6 is the most commonly involved oncogene in DLBCL and its depletion or blockade in primary DLBCL samples or human DLBCL cell lines causes cell death, suggesting that these tumors are often addicted to this oncoprotein and need its continuous function for survival [125, 126].

Also worth noting is module 1201, that associates high expression of ZMYND11 (also known as BS69) with poor survival. This nuclear protein has been shown to be a transcriptional repressor through interacting with other proteins like c-Myb or N-CoR [127, 128].

Also, several modules associate miRNAs with disease prognosis (table 4.1). For example, high expression of miR-152 (a miRNA associated with modules enriched for ECM genes) predicts poor prognosis in both modules 963 and 999 ($p< 0.0032$, FDR=0.07).

Furthermore, to relate our modules to previous expression-based classifications of DLBCL, we studied whether any of the regulatory programs' first split correlated with a consensus cluster (OxPhos, BCR, HR), a sub-grouping of tumors identified in [96]. We found that module 1207, enriched for immune system genes, is the most related to HR (P-value 1.07×10^{-5}, two-tailed Fisher's exact test with FDR correction). Also, modules enriched for oxidative phosphorylation genes (782, 800, 848, 1087) are highly related to the OxPhos consensus cluster (P-values 3.33×10^{-7}, 8.73×10^{-7}, 3.33×10^{-7}, 2.22×10^{-5}, respectively); however, the most statistically significant split refers to module 1165 (P-value 2.09×10^{-8}) which has a member of RAS oncogene family, RAP2C, as top regulator.

These results show that our approach, by consistently isolating a small group of relevant predictor genes from high-dimensional microarray datasets, efficiently reduces the amount of tests required to assess survival prognosis if this handful of predictor variables were not known.

4.8 Association between copy number aberrations and modules

In order to understand the functional importance of the genetic alterations in DLBCL, we looked for regulatory programs with CNAs as candidate drivers. In total, 39 of the 45 CNAs were chosen, with 23 assigned as top regulators. The only copy number alteration that was selected as a top regulator of a module and that is also associated with survival is a deletion peak in region 7q34 (p=0.0014, FDR=0.13). It is the top regulator of the inflammatory response module (1123, fig. 4-11), already referred above (fig. 4-8)). This deletion peak contains the genes PRSS1, PRSS2 and PRSS3P2, trypsinogen genes that are localized to the T cell receptor beta locus on chromosome 7.

Among the altered regions, there are in total seven gene deserts. For example, the most frequent aberration in our model was deletion peak in region 2q33.3, a gene desert that would be interesting to further investigate. Generally, these regions appear to be depleted of conserved elements, which is not surprising since they are in gene deserts. A total of $\sim$ 9600bp are in the conserved regions ($\sim$ 1.1% of the regions), where in random genomic regions we would expect to get $\sim$ 4.5%. Interestingly, one of the hits is in deletion peak in chromosomal region 2p16.3 and overlaps with a large intergenic noncoding RNA.

The CNAs did not split the gene expression in the modules as clearly (see e.g. 4-11B) as when we used mRNA or miRNA profiles for the regulatory programs (an ideal signature would be, for example, the upregulation of target genes when a region is amplified). This problem may be due to the distinct nature of the two datasets (discrete vs. real-valued measurements). Therefore we considered an alternative approach to study driver alterations. We looked for genes in the regulatory programs that were located in amplified or deleted regions, since the same gene may be targeted genetically (amplified/deleted) or epigenetically (up/down regulated). We found that 40 of the 231 regulator genes that were associated with at least with one module by Module Networks are in amplified or deleted regions. In particular, the following

genes and were chosen as top regulators in regulation programs: BCL6 (module 716), CD63 (modules 860 and 1075), CREBL2 (modules 920 and 969), MINK1 (modules 740, 842 and 1017), DKFZP564K0822 (module 1111) and G10 (module 1267).

Overall, the integration of CNAs with gene expression profiles can give insights of how the malignant phenotype relates to genetic aberrations from which it is likely to have originated.

4.9 Design of experimental validation of biological findings

We decided to focus on the glycolysis and the oxphos/mitochondria modules for further validation given the known role of the modules' regulators. In order to test whether the predicted regulators for these modules have a functional role, we suggest a few experiments. First, we looked for cell lines that were high or low expressed in these contexts. From the Broad-Novartis Cancer Cell Line Encyclopedia [129] we collected two gene sets, with the genes in modules 782 and 969, comprising 16 cell lines representative of the histology subtype DLBCL. We performed a Kolmogorov-Smirnov (K-S) test on the expression of the probes in these two gene sets relative to background of all probes on array. We then sorted the cell lines by the P-values of the test and scored the genes by correlation to the K-S P-value vector. Then the genes were sorted by the correlation values. From these results we suggest to use cell lines SUDHL5, NUDHL1, SUDHL4, Pfeiffer, A4FUK to knockdown genes COPS5, COPS4, MYCBP, MYC, CREBL2, CREB3L2, RAP1A, RAP1B, and measure growth and gene expression. These experiments will test whether cell growth and signature genes' expression are altered after these knockdowns in these cell lines. These experiments are ongoing.

4.10 Conclusion

In the past few years, the importance of data integration has increased, due to the rapidly increasing amount of genomic and other high throughput data. However, data from different sources are subject to different noise levels due to difference in technologies, platforms and other systematic or random factors affecting the experiments. Thus, data might have different qualities and a naïve combination of data is not appropriate in such cases. In addition, the several formats and dimensions in which data are produced can make simple merging not applicable and in some cases impossible.

Here, we propose a conceptual framework for genomic data integration. In particular, we reconstruct an integrative model of genetic aberrations, miRNA and regulator genes that may explain expression levels in DLBCL. We first use the Module Networks procedure to learn a model that associates modules of co-expressed genes with 'regulatory programs' that use a combination of transcription factors and signaling molecules. We further extend this approach to integrate other datasets, namely miRNA expression and DNA copy-number information. In particular, the effect of miRNAs on cell pathology and physiology is likely to be complex given the fact that their activity is exerted in a one-to-many fashion (each miRNA can control translation of tens or even hundreds of different coding genes) and that a single gene can be controlled by more than one miRNA.

Among other findings, our analysis associates miR152 and CD63 with survival, links MYCBP and COPS5 with a module of genes enriched for oxidative phosphorylation and mitochondrial functions, and CREBL2 with a glycolysis module. The model raises concrete testable hypotheses which can be tested by perturbation experiments: for example, using RNA interference to interrogate whether decreased expression of individual regulators alters the expression of target genes.

Overall, our analysis demonstrates the discovery potential of systems-level approaches and represents an essential component of a rational strategy for identifying drug mechanisms and developing new diagnostic tests and therapeutic approaches.

4.11 Figures and tables

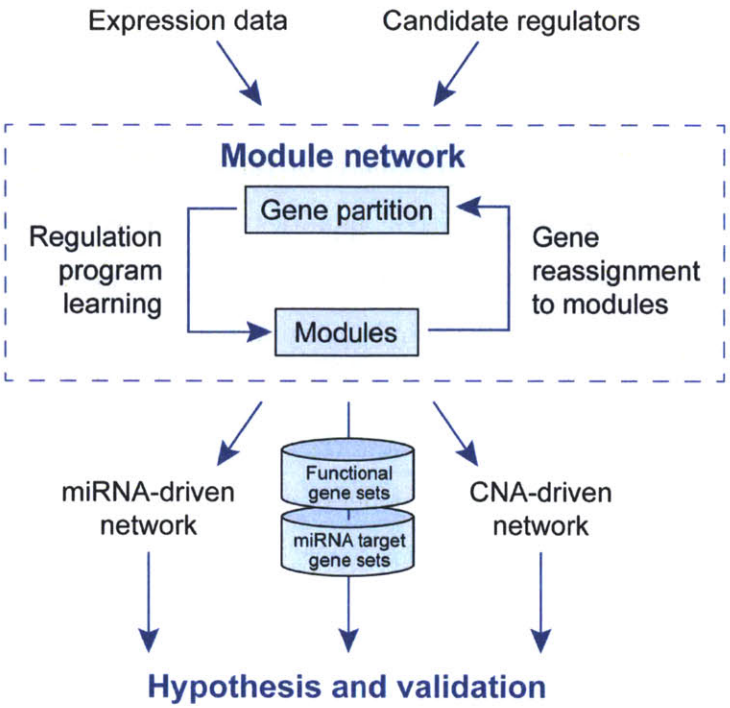

Figure 4-1: Analysis flow. Unsupervised reconstruction of transcriptional modules was performed with the Module Networks algorithm using transcription factors and signaling molecules as candidate regulators. The modules were utilized to study enrichment and to search for regulation programs with miRNAs and copy number alteration profiles. The results generate hypotheses about potential DLBCL drivers and genes associated with prognosis.

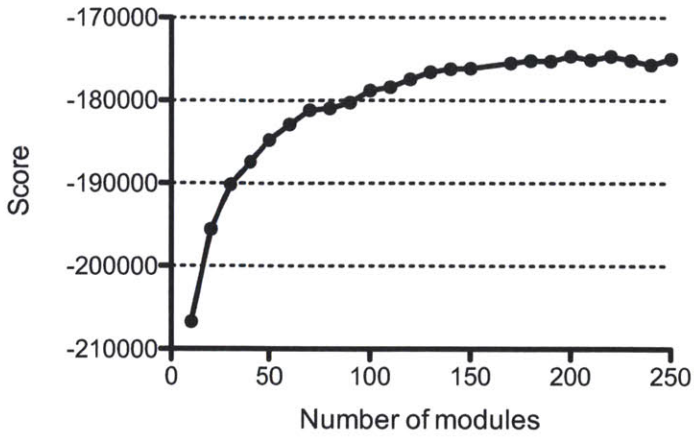

Figure 4-2: The average Bayesian Score per gene, as a function of module number, inferred with the Module Networks algorithm. We chose 100 modules, the model whose Bayesian score was 85% of the best score.

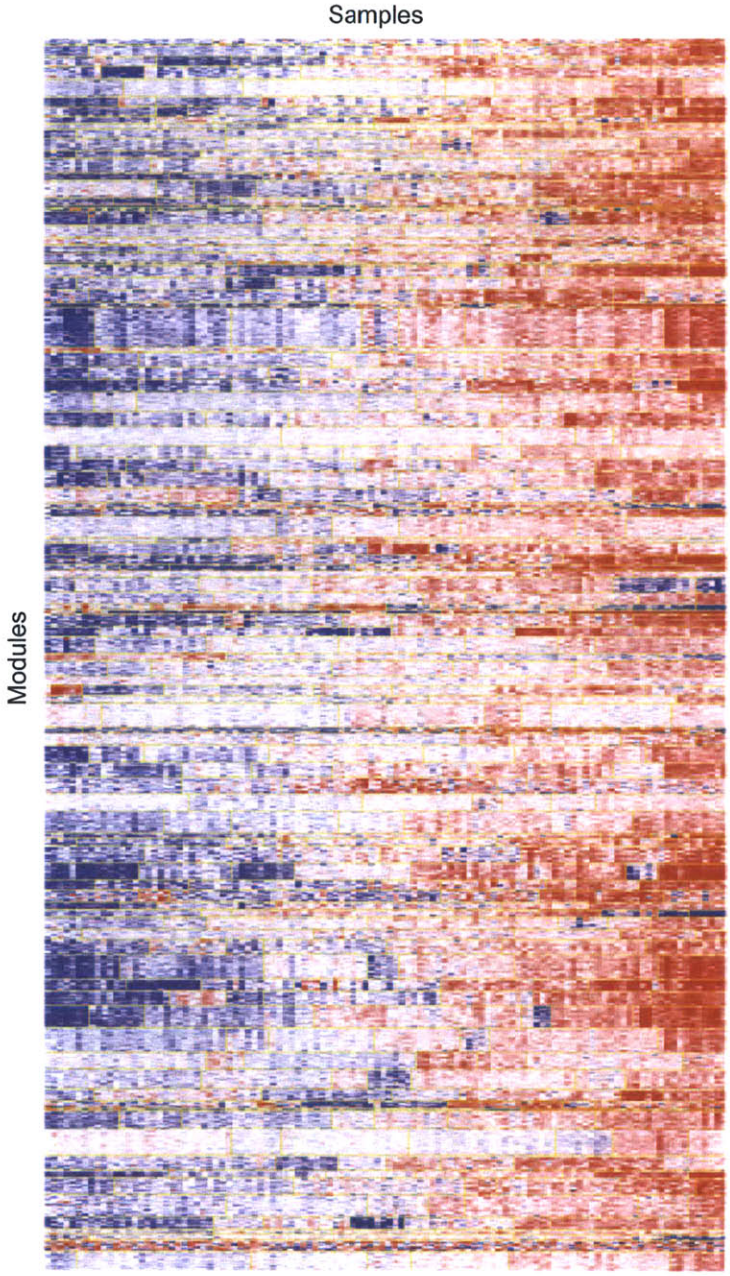

Figure 4-3: Birdseye view of the 100 modules. The modules are shown as horizontal strips (there are 100 such strips and thus 100 modules), and for each module, its samples are shown sorted by the regulatory program with each split in the tree shown by separate blocks separated by yellow lines.

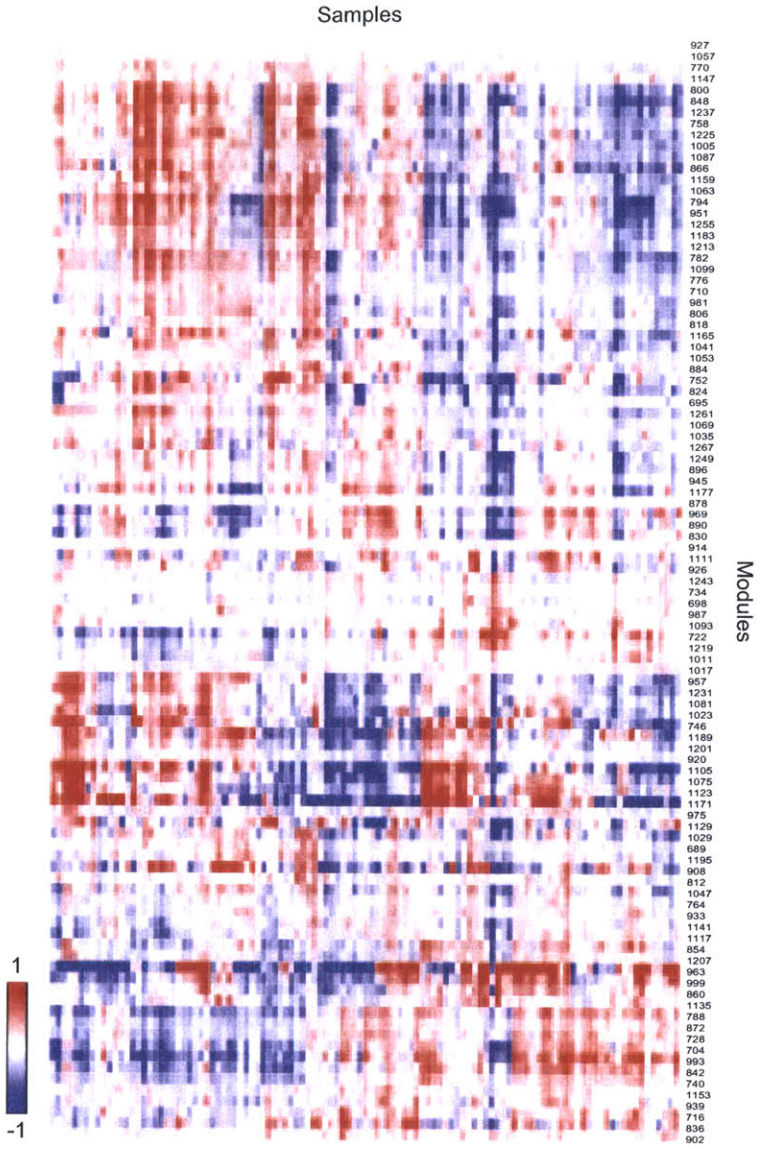

Figure 4-4: Global view of the 100 modules (rows) and 110 samples (columns) generated with the Module Networks algorithm and hierarchically clustered; the heatmap presents the average expression of the genes in each module in each sample.

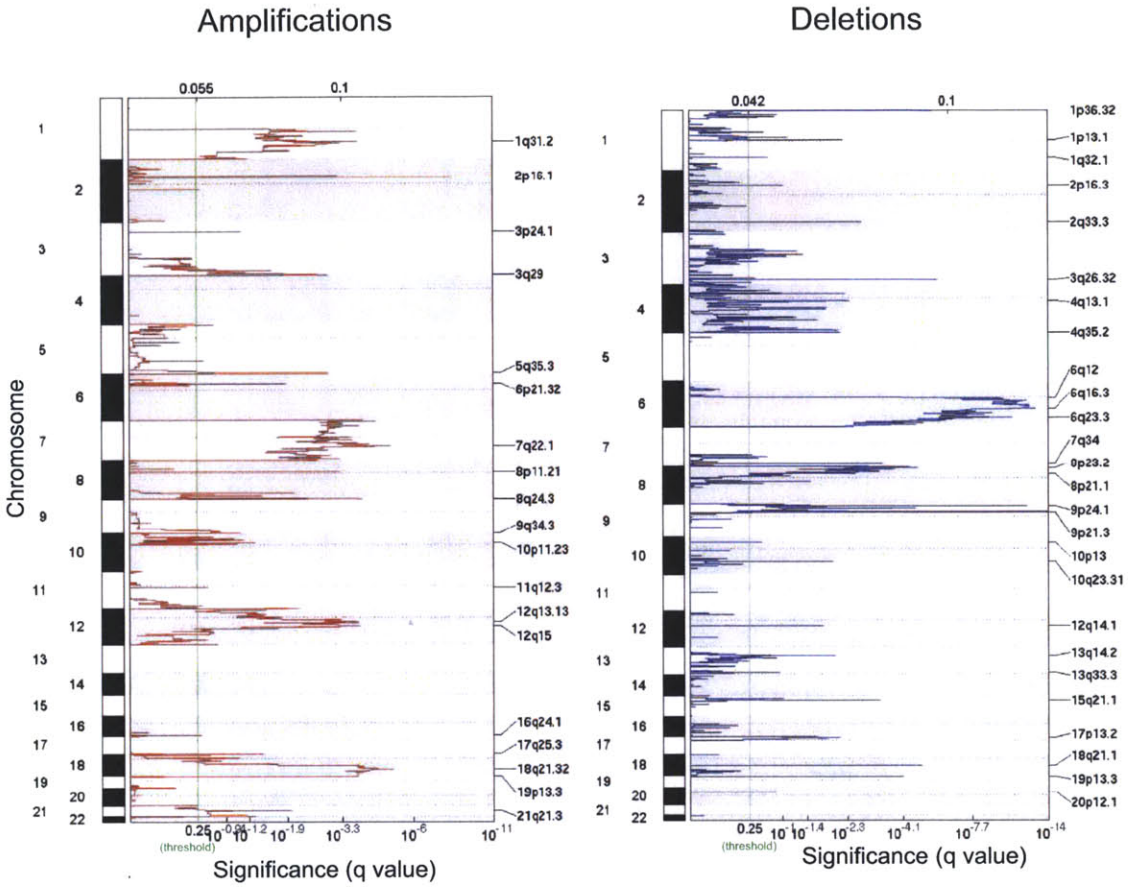

Figure 4-5: GISTIC Peaks. The statistical significance of the aberrations are displayed as FDR q values to account for multiple-hypothesis testing. Chromosome positions are indicated along the y axis with centromere positions indicated by dotted lines. The locations of the peak regions are indicated to the right of each panel (amplification peaks 1 to 19 and deletion peaks 1 to 26).

Figure 4-6: Gene sets enrichment of modules (columns) highlighted in the main text. Colors refer to the gene set source.

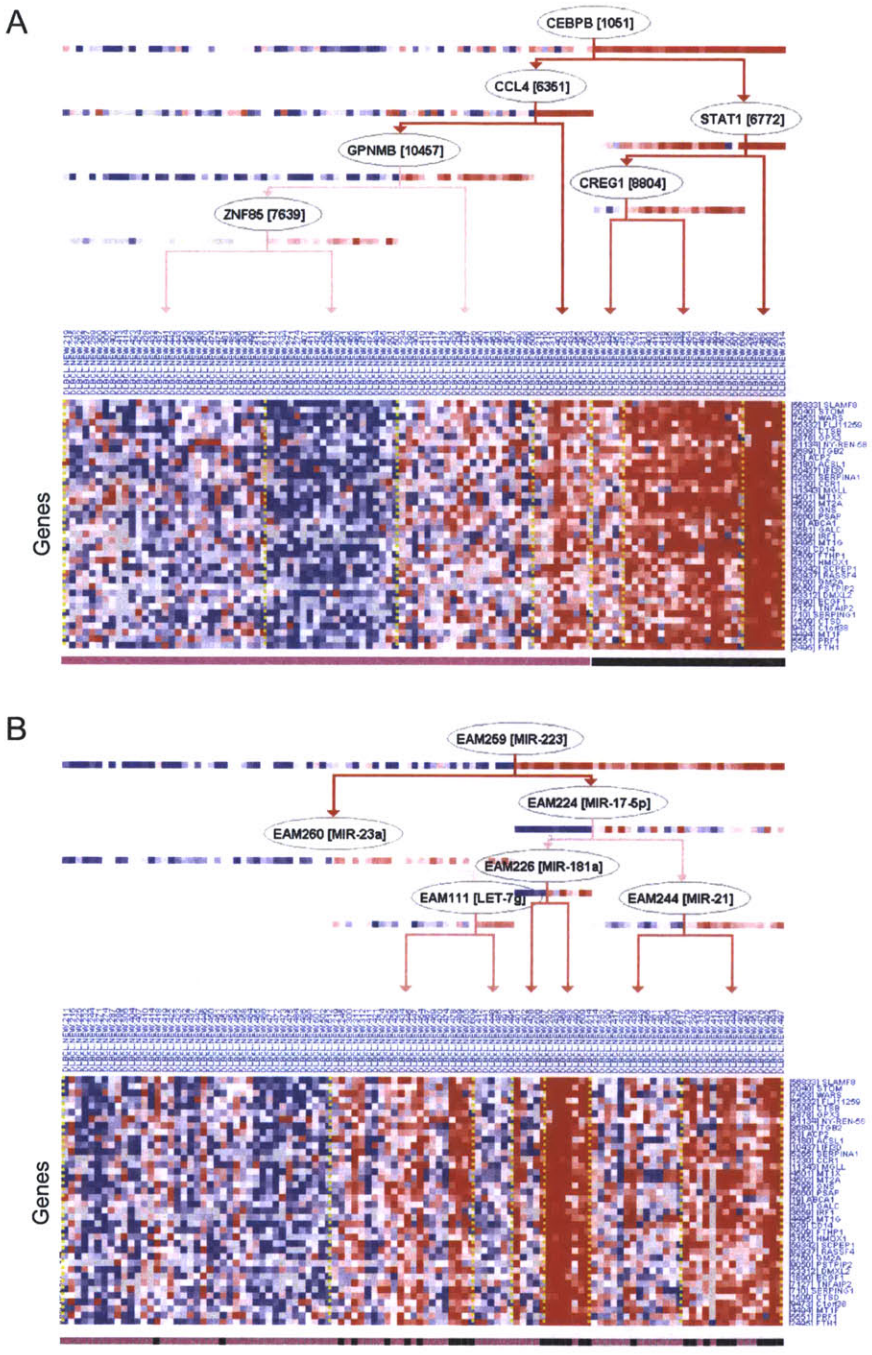

Figure 4-8: Module 1123, in which rows are genes and the tree represents the regulation program with CNAs. The module is globally regulated by CEBP-β (A) and miR-223 (B). Samples in (A) are colored pink or black if they were assigned to the left or to the right of the first split, respectively (bottom row). This label assignment is kept in B (bottom row).

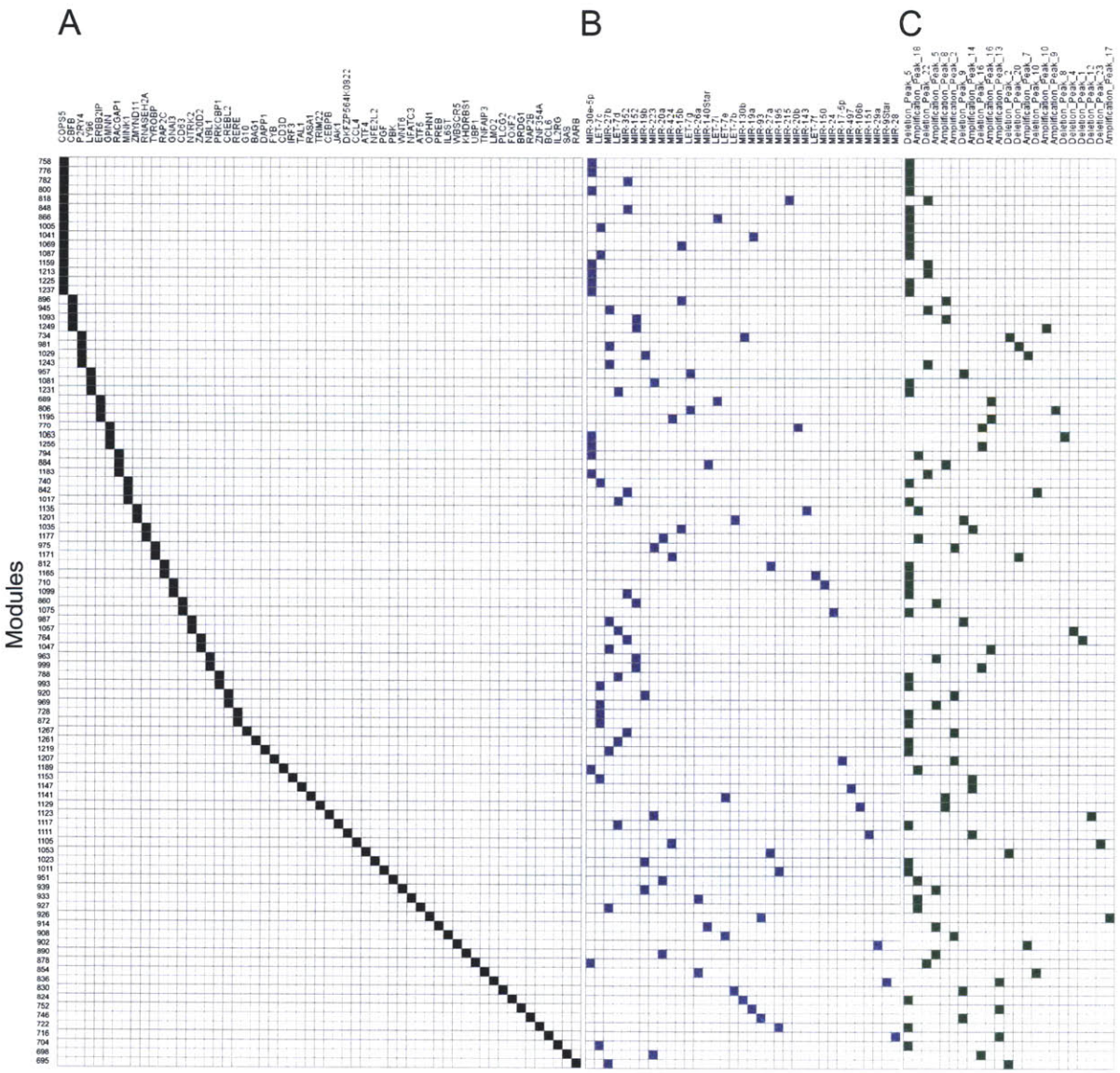

Figure 4-7: Top regulators (columns) for each module (rows) chosen for each data type. (A) transcription factors and signaling molecules. (B) microRNAs. (C) Copy number alterations.

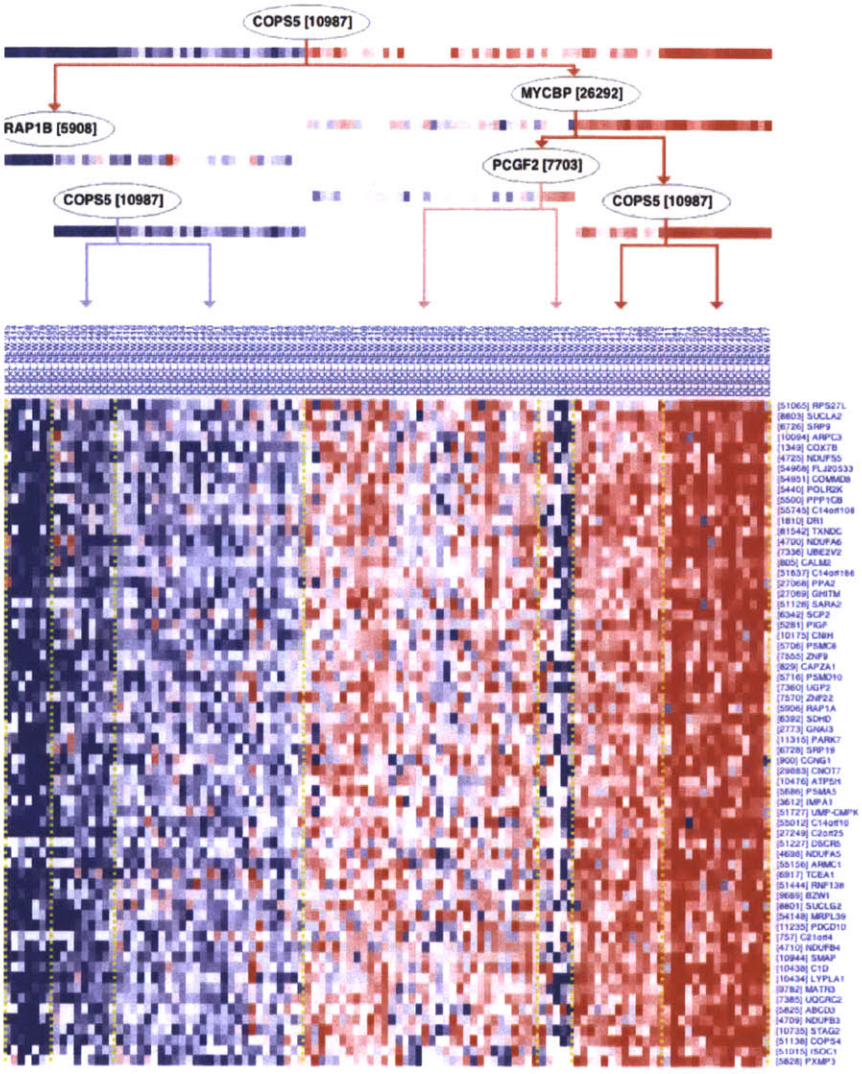

Figure 4-9: Oxidative phosphorylation and mitochondria gene module in DLBCL. COPS5 and MYCBP mRNA levels predict the functional signature. The module show genes in the rows and samples in the columns.

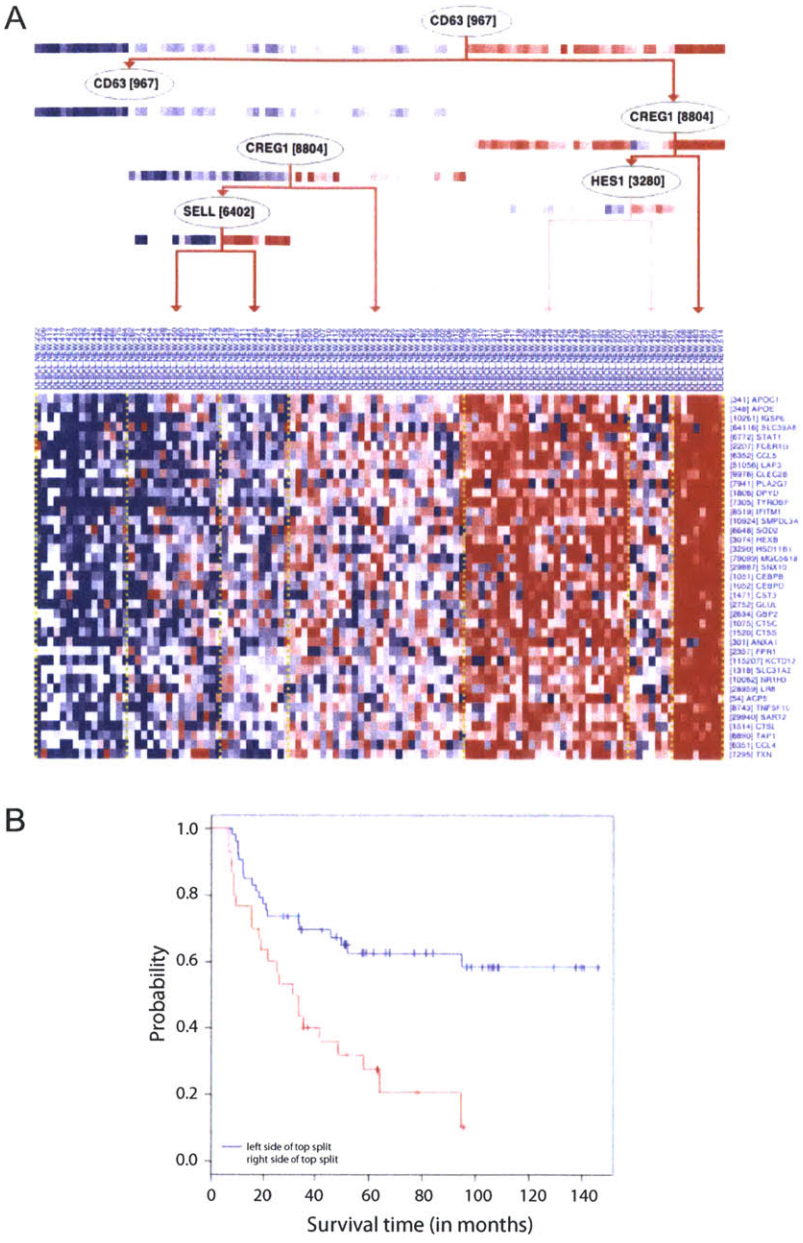

Figure 4-10: (A) The decision tree model of CD63 mRNA level was used to split the 110 samples into two groups. (B) Kaplan-Meier survival curve of samples partitioned in A. Group showing CD63 induction has increased probability of death.

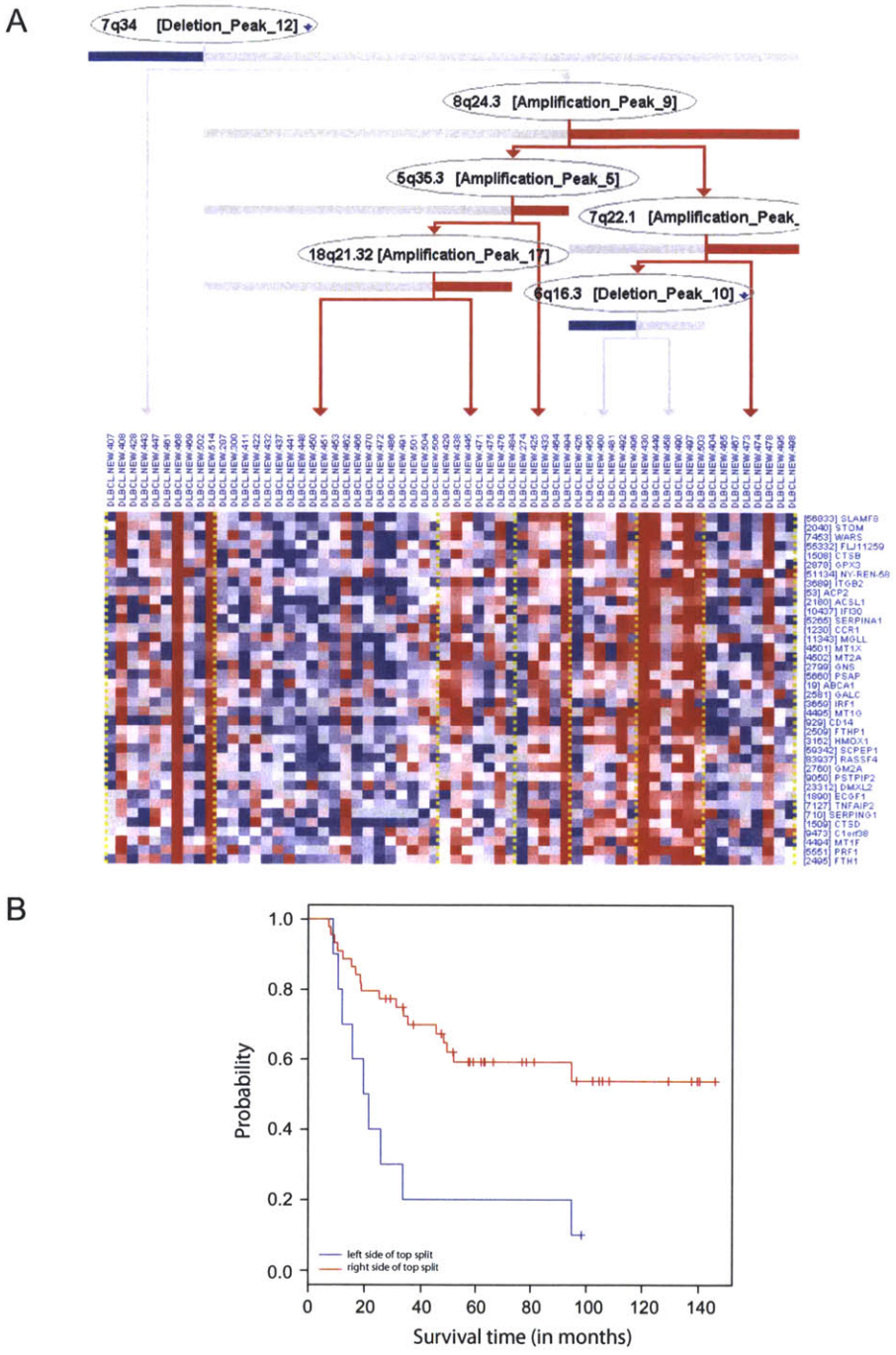

Figure 4-11: Module 1123. (A) The decision tree model identifies deletion in region 7q34 as a top regulator, which was used to split the samples into two groups. In the regulatory program, red and blue mean amplified and deleted, respectively) (B) Kaplan-Meier survival curve of samples partitioned in A. Group showing 7q34 deletion has decreased probability of survival.

Module ID	Top Regulator	# genes in module	# samples left side of split	# samples right side of split	P-value	FDR
746	RAP2B	5	77	33	0.00011	0.0077
926	OPHN1	29	6	104	0.00016	0.0077
1075	CD63	39	69	41	0.00043	0.014
716	BCL6	55	64	46	0.004	0.1
1201	ZMYND11	25	60	50	0.0092	0.17
1011	PGF	16	30	80	0.011	0.17
963	MIR-152	10	65	18	0.0014	0.0696
999	MIR-152	28	70	13	0.0028	0.0696
920	MIR-19b	23	13	70	0.0031	0.0696
927	MIR-27b	24	78	5	0.0035	0.0696
987	MIR-27b	18	78	5	0.0035	0.0696
1123	MIR-223	38	51	32	0.0089	0.147
1093	MIR-152	14	68	15	0.013	0.166
746	MIR-93	5	41	42	0.0164	0.166
860	MIR-152	25	69	14	0.0168	0.166
1249	MIR-152	44	69	14	0.0168	0.166
1231	LET-7d	32	71	12	0.0235	0.1911
908	LET-7e	18	37	46	0.0239	0.1911
1075	MIR-24	39	48	35	0.0279	0.1911
1201	LET-7b	25	52	31	0.0292	0.1911
782	MIR-352	64	64	19	0.0328	0.1911
848	MIR-352	53	64	19	0.0328	0.1911
1099	MIR-352	128	64	19	0.0328	0.1911
1267	MIR-352	25	44	39	0.0354	0.1945
1105	MIR-424	13	42	41	0.0402	0.1959
1171	MIR-424	7	42	41	0.0402	0.1959
975	MIR-223	32	67	16	0.0435	0.1959
1081	MIR-223	37	67	16	0.0435	0.1959
854	MIR-26a	49	51	32	0.0486	0.2007
933	MIR-26a	47	51	32	0.0486	0.2007
1261	LET-7d	40	44	39	0.0522	0.2067
884	MIR-140Star	46	47	36	0.0544	0.2071
830	LET-7b	19	71	12	0.0584	0.2142
926	MIR-93	29	16	67	0.067	0.2369
1057	LET-7d	57	56	27	0.0744	0.2474
788	LET-7d	44	72	11	0.075	0.2474

Table 4.1: List of modules whose top regulator in the regulatory program predicts survival prognosis.

Chapter 5

Conclusions

Chapter 5

Conclusions

5.1 Summary

In this thesis we present several approaches to systematically reconstruct, validate and refine a regulatory circuit. We take advantage of recent advances in genomics to advance our understanding of the molecular mechanisms controlling gene expression programs in dendritic cells and cancer.

In Chapter 2, we use gene expression data and knockdowns to reconstruct a functional regulatory network in mouse dendritic cells exposed to various pathogens. Linking gene expression to the response of a specific cell type after exposure to a pathogen may facilitate therapeutic targeting of specific pathways to enhance human vaccine efficacy or to combat the drivers of a disease. We first used microarrays to characterize transcriptional responses of dendritic cells to various pathogens at different time points. Then, building on the framework of probabilistic graphical models and the elastic net (LARS-EN) regression, we learned an observational model of gene regulation, identifying candidate regulators that act on target genes. The expression of these regulators was then reduced by $> 75\%$ using lentiviral small hairpin RNAs in dendritic cells. Then, a gene expression signature was measured at a selected time point, after exposure to a single treatment that activated many of the pathogen responses. The functional regulatory network reconstructed from these data agreed with the observational model only to a certain extent. The large amount of false-positive

interactions of the observational model is due to the fact that a correct regulator had gene expression profiles that were indistinguishable from other regulators, a clear shortcoming of models lacking functional approaches.

Each perturbation experiment associates a perturbed regulator with targets that are repressed or induced by the perturbation. However, it is expected that these include both direct and indirect targets. To distinguish those, we introduce in Chapter 3 the algorithm Exigo, used to 'prune' likely indirect interactions from perturbation screens. We used a matrix representation of the network topology to evaluate the number of self-avoiding random walks that can link a pair of connected nodes in the original network constructed to describe a set of perturbation experiments. This gives rise to a reference matrix that is used to generate a set of 'equivalent' networks consistent with experiments. The method improves on currently available methods to characterize parsimonious network topologies because Exigo can also analyze global effects of edge removal in networks. Even though Exigo is not itself a network inference procedure (but rather a method to distinguish direct from indirect interactions from experimental screens), we show that combining it with a state-of-the-art network inference method significantly improves inference results.

Lastly, in Chapter 4, we provide a system-level dissection of regulatory mechanisms in tumorigenesis. In particular, we built a module network for diffuse large B-cell lymphoma using multiple high-throughput assays (gene expression microarrays, DNA copy-number SNP arrays, and microRNA arrays) and clinical data. Our analysis identified several modules enriched for functional categories and predicted novel genes and microRNAs that have never been associated to this cancer type. We are conducting an ongoing perturbation screen to confirm the functions of these genes and provide a better understanding of how cellular networks are de-regulated in DLBCL. Overall, this study expands the knowledge about causal and combinatorial relationships that characterize molecular signatures in DLBCL, and provides a systematic approach for the integration and analysis of different types of datasets.

5.2 Future perspectives

While our work advances the systematic reconstruction of regulatory circuits in mammalian systems, the computational approaches are still limited in several aspects. Here, I will discuss these challenges and propose some directions for addressing them.

The initial observational model connects regulators to target genes based on linear dependencies between their temporal profiles. The defining property of linear models is that each regulator contributes to the input of the regulation function independently of the other regulators, in an additive manner. That is, the change in the level of each entity depends on a weighted linear sum of the levels of its regulators. Even though this assumption has a high level of abstraction, it proved, in Chapter 2, to be efficient in selecting regulators that were functional.

The perturbational dataset further allowed us to test the quality of the initial observational model, revealing a lot of false positive interactions. This was mainly due to the fact that a correct regulator had gene expression profiles that were indistinguishable from other regulators. This limited explanatory value shows that a linear model is a crude description of the process of gene regulation. To leverage the complementary power of both models, a possible next step could be to incorporate the interventional data in the learning method to refine the initial model. For example, using a prior to favor edges that correspond to the experimentally identified interactions in the perturbation screen can lead to a more predictive network [130]. Furthermore, genes which are affected differently by perturbations should be more likely assigned to distinct modules, even if their expression profiles are similar, whereas those with similar regulation may still be separated if their expression is distinct. The refined model would then give rise to new hypotheses that could be tested with other single-gene perturbation experiments or combinatorial ones. Iterating this cycle would sequentially improve the model's resolution, providing a deeper understanding of the specificity of the circuit.

The choice of candidate regulators in the observational model can be diversified to expand the scope of regulatory circuits. mRNA profiles can assist in identifying

candidate signaling proteins (e.g., [131]), chromatin factors, large non-coding RNAs, or RNA-binding proteins. In addition, several other global profiling technologies could be used in order to expand the circuit components to study, such as mass spectrometry [132] for metabolite profiling and for measuring protein abundance, and ChIP-seq [133] and HITS-CLIP [134] (also known as CLIP-Seq) to quantify protein-nucleic acid binding.

The scale and quality of networks will improve significantly when large-scale perturbations are coupled with next-generation sequencing [50]. Multiplexed RNA-Seq and ChIP-Seq approaches will allow thousands of transcriptomes to be directly linked to genome-wide binding profiles of thousands of transcription factors. Then it will be easier to discriminate direct from indirect gene regulation and to interpret networks in the context of additional mechanisms of RNA dependent regulation, including microRNA binding and alternative splicing. However, the identification of "non functional " and "redundant" binding is a remaining challenge. Linking predictive networks to function will facilitate the translation of molecular interactions into therapies.

Since many of the network interactions identified by gene knockdown are likely to be indirect, we developed Exigo. This method identifies the interactions that are necessary to explain a set of gene-perturbation experiments. Exigo can be extended beyond purely topological considerations by explicitly considering the confidence weights associated with each interaction in the perturbation screen.

In Chapter 4, we present an unsupervised approach for integrating different types of data. Our integration strategy consisted in examining a module of genes defined by one data type in the context other data types. In fact, it is particularly difficult to combine predictors from different date types in a simultaneous learning procedure since their quality and informativity are usually different. The strategy that we followed can be useful when the scope of genomic experiments performed is so diverse that it is not immediately clear how, or even if, one experiment relates to another. We showed that unsupervised integration can be a powerful discovery tool for finding regulatory associations, which can then be experimentally validated. However, when

we searched for regulatory programs of CNAs for the modules (of mRNA profiles), we noticed that CNAs did not discriminate well between contexts of expression profiles. This may be due to the fact that we were searching for regulatory programs of discrete-value measurements to explain modules of real-valued measurements. Thus, it would be useful to extend the regulatory program search to efficiently evaluate a set of discrete-valued candidate regulators.

The scale and scope of studies made possible by perturbation screens, along with the emergence and refinement of genomic, transcriptomic, and proteomic techniques, are providing a system-wide understanding of gene networks in an increasing number of specific cell types, tissues, and organisms. This holds great promise for the future of network reconstruction. The unbiased, systematic and integrative approaches I describe in this thesis show potential to enhance our understanding of biological systems. These approaches are general and applicable to almost any biological system, as well as practical for most laboratory settings, an important step that has broad implications for the scientific community.

Bibliography

[1] S.A. Ramsey, S.L. Klemm, D.E. Zak, K.A. Kennedy, V. Thorsson, B. Li, M. Gilchrist, E.S. Gold, C.D. Johnson, V. Litvak, G. Navarro, J.C. Roach, C.M. Rosenberger, A.G. Rust, N. Yudkovsky, A. Aderem, and I. Shmulevich. Uncovering a Macrophage Transcriptional Program by Integrating Evidence from Motif Scanning and Expression Dynamics. *PLoS Computational Biology*, 4(3):e1000021, March 2008.

[2] M.A. Beer and S. Tavazoie. Predicting gene expression from sequence. *Cell*, 117 (2):185–198, 2004.

[3] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):166–176, 2003.

[4] Z. Hu, P.J. Killion, and V.R. Iyer. Genetic reconstruction of a functional transcriptional regulatory network. *Nature Genetics*, 39(5):683–687, April 2007.

[5] R. Sopko, D. Huang, N. Preston, G. Chua, B. Papp, K. Kafadar, M. Snyder, S.G. Oliver, M. Cyert, and T.R. Hughes. Mapping pathways and phenotypes by systematic gene overexpression. *Molecular Cell*, 21(3):319–330, 2006.

[6] I. Amit, M. Garber, N. Chevrier, A.P. Leite, Y. Donner, T. Eisenhaure, M. Guttman, J.K. Grenier, W. Li, O. Zuk, L.A. Schubert, B. Birditt, T. Shay, A. Goren, X. Zhang, Z. Smith, R. Deering, R.C. McDonald, M. Cabili, B.E. Bernstein, J.L. Rinn, A. Meissner, D.E. Root, N. Hacohen, and A. Regev. Unbiased Reconstruction of a Mammalian Transcriptional Network Mediating Pathogen Responses. *Science*, 326(5950):257–263, October 2009.

[7] M.V. Rockman. Reverse engineering the genotype–phenotype map with natural genetic variation. *Nature*, 456(7223):738–744, 2008.

[8] H.D. Kim, T. Shay, E.K. O'Shea, and A. Regev. Transcriptional Regulatory Circuits: Predicting Numbers from Alphabets. *Science*, 325(5939):429–432, 2009.

[9] I. Amit, A. Citri, T. Shay, Y. Lu, M. Katz, F. Zhang, G. Tarcic, D. Siwak, J. Lahad, J. Jacob-Hirsch, N. Amariglio, N. Vaisman, E. Segal, G. Rechavi, U. Alon, G.B. Mills, E. Domany, and Y. Yarden. A module of negative feedback regulators defines growth factor signaling. *Nature Genetics*, 39(4):503–512, February 2007.

[10] Z. Bar-Joseph, G.K. Gerber, T.I. Lee, N.J. Rinaldi, J.Y. Yoo, F. Robert, D.B. Gordon, E. Fraenkel, T.S. Jaakkola, R.A. Young, and D.K. Gifford. Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21(11):1337–1342, October 2003.

[11] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799, 2004.

[12] N. Novershtern, A. Subramanian, L.N. Lawton, R.H. Mak, W.N. Haining, M.E. McConkey, N. Habib, N. Yosef, C.Y. Chang, T. Shay, G.M. Frampton, A.C.B. Drake, I. Leskov, B. Nilsson, F. Preffer, D. Dombkowski, J.W. Evans, T. Liefeld, J.S. Smutko, J. Chen, N. Friedman, R.A. Young, T.R. Golub, A. Regev, and B.L. Ebert. Densely Interconnected Transcriptional Circuits Control Cell States in Human Hematopoiesis. *Cell*, 144(2):296–309, January 2011.

[13] E. Segal, R. Yelensky, and D. Koller. Genome-wide discovery of transcriptional

modules from DNA sequence and gene expression. *Bioinformatics*, 19(Suppl 1): i273–i282, July 2003.

[14] S.-I. Lee, D. Pe'er, A.M. Dudley, G.M. Church, and D. Koller. Identifying regulatory mechanisms using individual variation reveals key role for chromatin modification. *PNAS*, 103(38):14062–14067, 2006.

[15] S.-I. Lee, A.M. Dudley, D. Drubin, P.A. Silver, N.J. Krogan, D. Pe'er, and D. Koller. Learning a Prior on Regulatory Potential from eQTL Data. *PLoS Genetics*, 5(1): e1000358, January 2009.

[16] U.D. Akavia, O. Litvin, J. Kim, F. Sanchez-Garcia, D. Kotliar, H.C. Causton, P. Pochanard, E. Mozes, L.A. Garraway, and D. Pe'er. An Integrated Approach to Uncover Drivers of Cancer. *Cell*, 143(6):1005–1017, December 2010.

[17] G. Karlebach and R. Shamir. Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9 (10):770–780, September 2008.

[18] D. Das, M. Pellegrini, and J.W. Gray. A primer on regression methods for decoding cis-regulatory logic. *PLoS Computational Biology*, 5(1):e1000269, 2009.

[19] H.J. Bussemaker, B.C. Foat, and L.D. Ward. Predictive Modeling of Genome-Wide mRNA Expression: From Modules to Molecules. *Annual Review of Biophysics and Biomolecular Structure*, 36(1): 329–347, June 2007.

[20] A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[21] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[22] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67:301–320, 2005.

[23] N. Yosef and A. Regev. Impulse Control: Temporal Dynamics in Gene Transcription. *Cell*, 144(6):886–896, March 2011.

[24] I.P. Androulakis, E. Yang, and R.R. Almon. Analysis of Time-Series Gene Expression Data: Methods, Challenges, and Opportunities. *Annual Review of Biomedical Engineering*, 9(1):205–228, August 2007.

[25] R. Bonneau. Learning biological networks: from modules to dynamics. *Nature Chemical Biology*, 4(11):658–664, November 2008.

[26] T. Chen, H.L. He, and G.M. Church. Modeling gene expression with differential equations. *Pacific symposium on biocomputing*, 4(29):4, 1999.

[27] S.Y. Kim, S. Imoto, and S. Miyano. Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Briefings in bioinformatics*, 4(3): 228–235, 2003.

[28] A.C. Lozano, N. Abe, Y. Liu, and S. Rosset. Grouped graphical Granger modeling for gene expression regulatory networks discovery. *Bioinformatics*, 25(12): i110–i118, 2009.

[29] A. Schliep, A. Schonhuth, and C. Steinhoff. Using hidden Markov models to analyze gene expression time course data. *Bioinformatics*, 19(suppl 1):i255–i263, 2003.

[30] P. Zoppoli, S. Morganella, and M. Ceccarelli. TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC Bioinformatics*, 11(1):154, 2010.

[31] M. Baym, C. Bakal, N. Perrimon, and B. Berger. High-resolution modeling of cellular signaling networks. *Proceedings of the 12th annual international conference on Research in computational molecular biology*, pages 257–271, 2008.

[32] D.H. Erwin and E.H. Davidson. The evolution of hierarchical gene regulatory networks. *Nature Reviews Genetics*, 10(2): 141–148, 2009.

[33] A.P. Capaldi, T. Kaplan, Y. Liu, N. Habib, A. Regev, N. Friedman, and E.K. O'Shea. Structure and function of a transcriptional network activated by the MAPK Hog1. *Nature Genetics*, 40(11):1300–1306, October 2008.

[34] A. Wagner. How to reconstruct a large genetic network from n gene perturbations in fewer than n(2) easy steps. *Bioinformatics*, 17(12):1183–1197, 2001.

[35] A. Wagner. Estimating Coarse Gene Network Structure from Large-Scale Gene Perturbation Data. *Genome Research*, 12(2): 309–315, February 2002.

[36] A. Wagner. Reconstructing pathways in large genetic networks from genetic perturbations. *Journal of computational biology*, 11(1):53–60, 2004.

[37] A.V. Aho, M.R. Garey, and J.D. Ullman. The transitive reduction of a directed graph. *SIAM J. Comput.*, 1(2):131–137, 1972.

[38] A. Tresch, T. Beissbarth, H. Sültmann, R. Kuner, A. Poustka, and A. Buness. Discrimination of Direct and Indirect Interactions in a Network of Regulatory Effects. *Journal of computational biology*, 14 (9):1217–1228, November 2007.

[39] S. Kachalo, R. Zhang, E. Sontag, R. Albert, and B. DasGupta. NET-SYNTHESIS: a software for synthesis, inference and simplification of signal transduction networks. *Bioinformatics*, 24 (2):293–295, January 2008.

[40] S. Klamt, R.J. Flassig, and K. Sundmacher. TRANSWESD: inferring cellular networks with transitive reduction. *Bioinformatics*, 26(17):2160–2168, August 2010.

[41] T.I. Lee, N.J. Rinaldi, F. Robert, D.T. Odom, Z. Bar-Joseph, G.K. Gerber, N.M. Hannett, C.T. Harbison, C.M. Thompson, and I. Simon. Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, 298(5594):799, 2002.

[42] C.T. Harbison, D.B. Gordon, T.I. Lee, N.J. Rinaldi, K.D. Macisaac, T.W. Danford, N.M. Hannett, J.B. Tagne, D.B. Reynolds, J. Yoo, E.G. Jennings, J. Zeitlinger, D.K. Pokholok, M. Kellis, P.A. Rolfe, K.T. Takusagawa, E.S. Lander, D.K. Gifford, E. Fraenkel, and R.A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104, 2004.

[43] C.T. Workman. A Systems Approach to Mapping DNA Damage Response Pathways. *Science*, 312(5776):1054–1059, May 2006.

[44] S.-s.C. Huang and E. Fraenkel. Integrating Proteomic, Transcriptional, and Interactome Data Reveals Hidden Components of Signaling and Regulatory Networks. *Science Signaling*, 2(81):ra40–ra40, July 2009.

[45] D. Pe'er and N. Hacohen. Principles and Strategies for Developing Network Models in Cancer. *Cell*, 144(6):864–873, March 2011.

[46] C. Blenkiron, L.D. Goldstein, N.P. Thorne, I. Spiteri, S.F. Chin, M.J. Dunning, N.L. Barbosa-Morais, A.E. Teschendorff, A.R. Green, and I.O. Ellis. MicroRNA expression profiling of human breast cancer identifies new markers of tumor subtype. *Genome Biology*, 8(10):R214, 2007.

[47] The Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216): 1061–1068, September 2008.

[48] Q. Huang. The Plasticity of Dendritic Cell Responses to Pathogens and Their Components. *Science*, 294(5543):870–875, October 2001.

[49] T. Kawai and S. Akira. The roles of TLRs, RLRs and NLRs in pathogen recognition. *International Immunology*, 21(4):317–337, March 2009.

[50] H. Suzuki, J. Gough, and FANTOM Consortium. The transcriptional network that controls growth arrest and differentiation in a human myeloid leukemia cell line. *Nature Genetics*, 41(5):553–562, April 2009.

[51] M. Gilchrist, V. Thorsson, B. Li, A.G. Rust, M. Korb, K. Kennedy, T. Hai,

H. Bolouri, and A. Aderem. Systems biology approaches identify ATF3 as a negative regulator of Toll-like receptor 4. *Nature*, 441(7090):173–178, 2006.

[52] V. Litvak, S.A. Ramsey, A.G. Rust, D.E. Zak, K.A. Kennedy, A.E. Lampano, M. Nykter, I. Shmulevich, and A. Aderem. Function of C/EBP in a regulatory circuit that discriminates between transient and persistent TLR4-induced signals. *Nature immunology*, 10(4):437–443, March 2009.

[53] J. Moffat, D.A. Grueneberg, X. Yang, S.Y. Kim, A.M. Kloepfer, G. Hinkle, B. Piqani, T.M. Eisenhaure, B. Luo, J.K. Grenier, A.E. Carpenter, S.Y. Foo, S.A. Stewart, B.R. Stockwell, N. Hacohen, W.C. Hahn, E.S. Lander, D.M. Sabatini, and D.E. Root. A Lentiviral RNAi Library for Human and Mouse Genes Applied to an Arrayed Viral High-Content Screen. *Cell*, 124 (6):1283–1298, March 2006.

[54] J.C. Kagan, T. Su, T. Horng, A. Chow, S. Akira, and R. Medzhitov. TRAM couples endocytosis of Toll-like receptor 4 to the induction of interferon-beta. *Nature immunology*, 9(4):361–368, February 2008.

[55] S.E. Doyle, S.A. Vaidya, R. O'Connell, H. Dadgostar, P.W. Dempsey, T.T. Wu, G. Rao, R. Sun, M.E. Haberland, R.L. Modlin, and G. Cheng. IRF3 mediates a TLR3/TLR4-specific antiviral gene program. *Immunity*, 17(3):251–263, 2002.

[56] D. Pe'er, A. Regev, and A. Tanay. Minreg: inferring an active regulator set. *Bioinformatics*, 18(suppl 1):S258–S267, 2002.

[57] H.Y. Yu, N.M. Luscombe, J. Qian, and M. Gerstein. Genomic analysis of gene expression relationships in transcriptional regulatory networks. *Trends in Genetics*, 19(8):422–427, 2003.

[58] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B Methodological*, 58(1):267–288, 1996.

[59] A.E. Hoerl. Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58(3):5459, 1962.

[60] B. Efron. Bootstrap methods: another look at the jackknife. *Ann. Statistics*, 7:1–26, 1979.

[61] G.K. Geiss, R.E. Bumgarner, B. Birditt, T. Dahl, N. Dowidar, D.L. Dunaway, H.P. Fell, S. Ferree, R.D. George, T. Grogan, J.J. James, M. Maysuria, J.D. Mitton, P. Oliveri, J.L. Osborn, T. Peng, A.L. Ratcliffe, P.J. Webster, E.H. Davidson, and L. Hood. Direct multiplexed measurement of gene expression with color-coded probe pairs. *Nature Biotechnology*, 26(3):317–325, February 2008.

[62] S. Mangan and U. Alon. Structure and function of the feed-forward loop network motif. *PNAS*, 100(21):11980, 2003.

[63] U. Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, June 2007.

[64] R.A. Irizarry, B. Hobbs, F. Collin, Y.D. Beazer Barclay, K.J. Antonellis, U. Scherf, and T.P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

[65] V Matys, O.V. Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, and K. Hornischer. TRANSFAC® and its module TRANSCompel®: transcriptional gene regulation in eukaryotes. *Nucleic Acids Research*, 34(suppl 1):D108–D110, 2006.

[66] A. Sandelin, W. Alkema, P. Engström, W.W. Wasserman, and B. Lenhard. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Research*, 32(suppl 1):D91–D94, 2004.

[67] G. Badis, M.F. Berger, A.A. Philippakis, S. Talukder, A.R. Gehrke, S.A. Jaeger, E.T. Chan, G. Metzler, A. Vedenko, X. Chen, H. Kuznetsov, C.F. Wang, D. Coburn, D.E. Newburger, Q. Morris, T.R. Hughes, and M.L. Bulyk. Diversity and Complexity in DNA Recognition by Transcription Factors. *Science*, 324(5935): 1720–1723, June 2009.

[68] M.F. Berger, G. Badis, A.R. Gehrke, S. Talukder, A.A. Philippakis, L. Pena-Castillo, T.M. Alleyne, S. Mnaimneh, O.B. Botvinnik, and E.T. Chan. Variation in homeodomain DNA binding revealed by high-resolution analysis of sequence preferences. *Cell*, 133(7):1266–1276, 2008.

[69] S. Mohr, C. Bakal, and N. Perrimon. Genomic Screening with RNAi: Results and Challenges. *Annual Review of Biochemistry*, 79(1):37–64, June 2010.

[70] F. Markowetz. How to understand the cell by breaking it: network analysis of gene perturbation screens. *arXiv.org*, q-bio.MN, October 2009.

[71] S.E. Martin and N.J. Caplen. Applications of RNA Interference in Mammalian Systems. *Annual Review of Genomics and Human Genetics*, 8(1):81–108, September 2007.

[72] D. Marbach, R.J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. *PNAS*, 107(14):6286–6291, April 2010.

[73] A. Pinna, N. Soranzo, and A. de la Fuente. From Knockouts to Networks: Establishing Direct Cause-Effect Relationships through Graph Analysis. *PLoS ONE*, 5 (10):e12912, October 2010.

[74] S.G. Tringe, A. Wagner, and S.W. Ruby. Enriching for direct regulatory targets in perturbed gene-expression profiles. *Genome Biology*, 5(4):R29, 2004.

[75] S. Son and H. Jeong. Reconstruction of a genetic network from gene perturbation data. *Journal-Korean Physical Society*, 48: 208, 2006.

[76] R. Albert, B. DasGupta, R. Dondi, S. Kachalo, E. Sontag, A. Zelikovsky, and K. Westbrooks. A novel method for signal transduction network inference from indirect experimental evidence. *Journal of computational biology*, 14(7):927–949, 2007.

[77] J. Rudnick and G. Gaspari. *Elements of the random walk: an introduction for advanced students and researchers.* Cambridge University Press, 2004.

[78] J. Ponstein. Self-avoiding paths and the adjacency matrix of a graph. *SIAM Journal on Applied Mathematics*, 14(3):600–609, 1966.

[79] E. Estrada, D.J. Higham, and N. Hatano. Communicability Betweenness in Complex Networks. *arXiv.org*, physics.soc-ph, May 2009.

[80] N. Clisby, R. Liang, and G. Slade. Self-avoiding walk enumeration via the lace expansion. *Journal of Physics A: Mathematical and Theoretical*, 40:10973, 2007.

[81] S. Lipschutz and M. Lipson. *Schaum's Outline of Theory and Problems of Discrete Mathematics.* Schaum's Outline Series. McGraw-Hill, 1997. ISBN 9780070380455.

[82] T. Schaffter, D. Marbach, and D. Floreano. GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270, August 2011.

[83] J. Lamb. The Connectivity Map: Using Gene-Expression Signatures to Connect Small Molecules, Genes, and Disease. *Science*, 313(5795):1929–1935, September 2006.

[84] H.E. Stanley. *Introduction to Phase Transitions and Critical Phenomena.* International Series of Monographs on Physics. Oxford University Press, USA, 1987.

[85] J.R. Nevins and A. Potti. Mining gene expression profiles: expression signatures as cancer phenotypes. *Nature Reviews Genetics*, 8(8):601–609, July 2007.

[86] J.S. Hamid, P. Hu, N.M. Roslin, V. Ling, C.M.T. Greenwood, and J. Beyene. Data Integration in Genetics and Genomics: Methods and Challenges. *Human Genomics and Proteomics*, 2009, 2009.

[87] N. Huang, P.K. Shah, and C. Li. Lessons from a decade of integrating cancer copy number alterations with gene expression

profiles. *Briefings in bioinformatics*, 13(3): 305–316, May 2012.

[88] Y. Zhang, J.W.M. Martens, J.X. Yu, J. Jiang, A.M. Sieuwerts, M. Smid, J.G.M. Klijn, Y. Wang, and J.A. Foekens. Copy number alterations that predict metastatic capability of human breast cancer. *Cancer Research*, 69(9):3795, 2009.

[89] R. Shen, A.B. Olshen, and M. Ladanyi. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, 25(22):2906–2912, 2009.

[90] E. Bonnet, T. Michoel, and Y. Van de Peer. Prediction of a gene regulatory network linked to prostate cancer from gene expression, microRNA and clinical data. *Bioinformatics*, 26(18):i638–i644, September 2010.

[91] J.G. Joung and Z. Fei. Identification of microRNA regulatory modules in Arabidopsis via a probabilistic graphical model. *Bioinformatics*, 25(3):387–393, 2009.

[92] Y.J. Cho, A. Tsherniak, P. Tamayo, S. Santagata, A. Ligon, H. Greulich, R. Berhoukim, V. Amani, L. Goumnerova, and C.G. Eberhart. Integrative genomic analysis of medulloblastoma identifies a molecular subgroup that drives poor clinical outcome. *Journal of clinical oncology*, 29(11):1424–1430, 2011.

[93] J.P. Brunet, P. Tamayo, T.R. Golub, and J.P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *PNAS*, 101(12):4164, 2004.

[94] R.S. Schwartz, G. Lenz, and L.M. Staudt. Aggressive lymphomas. *New England Journal of Medicine*, 362(15):1417–1429, 2010.

[95] M.A. Shipp. Molecular signatures define new rational treatment targets in large B-cell lymphomas. *Hematology Am Soc Hematol Educ Program*, 2007(1):265–269, 2007.

[96] S. Monti, K.J. Savage, J.L. Kutok, F. Feuerhake, P. Kurtin, M. Mihm, B. Wu, L. Pasqualucci, D. Neuberg, and R.C.T. Aguiar. Molecular profiling of diffuse large B-cell lymphoma identifies robust subtypes including one characterized by host inflammatory response. *Blood*, 105(5):1851–1861, 2005.

[97] E. Segal, D. Pe'er, A. Regev, D. Koller, and N. Friedman. Learning module networks. *Journal of Machine Learning Research*, 6: 557–588, 2005.

[98] K. Takeyama, S. Monti, J.P. Manis, P.D. Cin, G. Getz, R. Beroukhim, S. Dutt, J.C. Aster, F.W. Alt, T.R. Golub, and M.A. Shipp. Integrative analysis reveals 53BP1 copy loss and decreased expression in a subset of human diffuse large B-cell lymphomas. *Oncogene*, 27(3):318–322, July 2008.

[99] R. Beroukhim, G. Getz, L. Nghiemphu, J. Barretina, T. Hsueh, D. Linhart, I. Vivanco, J.C. Lee, J.H. Huang, and S. Alexander. Assessing the significance of chromosomal aberrations in cancer: methodology and application to glioma. *PNAS*, 104(50):20007, 2007.

[100] The Gene Ontology Consortium. The Gene Ontology project in 2008. *Nucleic Acids Research*, 36(Database):D440–D444, December 2008.

[101] A. Liberzon, A. Subramanian, R. Pinchback, H. Thorvaldsdottir, P. Tamayo, and J.P. Mesirov. Molecular signatures database (MSigDB) 3.0. *Bioinformatics*, 27(12):1739–1740, June 2011.

[102] R.C. Friedman, K.K.H. Farh, C.B. Burge, and D.P. Bartel. Most mammalian mRNAs are conserved targets of microRNAs. *Genome Research*, 19(1):92–105, October 2009.

[103] G.L. Papadopoulos, M. Reczko, V.A. Simossis, P. Sethupathy, and A.G. Hatzigeorgiou. The database of experimentally supported targets: a functional update of TarBase. *Nucleic Acids Research*, 37 (Database):D155–D158, January 2009.

[104] I. Ben-Porath, M.W. Thomson, V.J. Carey, R. Ge, G.W. Bell, A. Regev, and R.A. Weinberg. An embryonic stem cell–like gene expression signature in poorly differentiated aggressive human tumors. *Nature Genetics*, 40(5):499–507, May 2008.

[105] J. Lekstrom-Himes and K.G. Xanthopoulos. Biological role of the CCAAT/enhancer-binding protein family of transcription factors. *Journal of Biological Chemistry*, 273(44):28545–28548, 1998.

[106] W. Sun, W. Shen, S. Yang, F. Hu, H Li, and T.H. Zhu. miR-223 and miR-142 attenuate hematopoietic cell proliferation, and miR-223 positively regulates miR-142 through LMO2 isoforms and CEBP-beta. *Cell Research*, 20(10):1158–1169, 2010.

[107] I. Ivanovska, A.S. Ball, R.L. Diaz, J.F. Magnus, M. Kibukawa, J.M. Schelter, S.V. Kobayashi, L. Lim, J. Burchard, A.L. Jackson, P.S. Linsley, and M.A. Cleary. MicroRNAs in the miR-106b Family Regulate p21/CDKN1A and Promote Cell Cycle Progression. *Molecular and Cellular Biology*, 28(7):2167–2174, March 2008.

[108] H.-I. Trompeter, H. Abbad, K.M. Iwaniuk, M. Hafner, N. Renwick, T. Tuschl, J. Schira, H.W. Müller, and P. Wernet. MicroRNAs MiR-17, MiR-20a, and MiR-106b Act in Concert to Modulate E2F Activity on Cell Cycle Arrest during Neuronal Lineage Differentiation of USSC. *PLoS ONE*, 6(1):e16138, January 2011.

[109] Y. Zhu, Y. Lu, Q. Zhang, J.J. Liu, T.J. Li, J.R. Yang, C. Zeng, and S.M. Zhuang. MicroRNA-26a/b and their host genes cooperate to inhibit the G1/S transition by activating the pRb protein. *Nucleic Acids Research*, 40(10):4615–4625, May 2011.

[110] D. Yu, D. Cozma, A. Park, and A. Thomas-Tikhonenko. Functional Validation of Genes Implicated in Lymphomagenesis: An in VivoSelection Assay Using a Myc-Induced B-Cell Tumor. *Annals of the New York Academy of Sciences*, 1059 (1):145–159, November 2005.

[111] M. Cacciatore, C. Guarnotta, M. Calvaruso, S. Sangaletti, A.M. Florena, V. Franco, M.P. Colombo, and C. Tripodo. Microenvironment-Centred Dynamics in Aggressive B-Cell Lymphomas. *Advances in Hematology*, 2012, 2012.

[112] D.J. Wong, D.S.A. Nuyten, A. Regev, M. Lin, A.S. Adler, E. Segal, M.J. van de Vijver, and H.Y. Chang. Revealing Targeted Therapy for Human Cancer by Gene Module Maps. *Cancer Research*, 68(2): 369–378, January 2008.

[113] A.S. Adler, M. Lin, H. Horlings, D.S.A. Nuyten, M.J. van de Vijver, and H.Y. Chang. Genetic regulators of large-scale transcriptional signatures in cancer. *Nature Genetics*, 38(4):421–430, March 2006.

[114] B. Hoffman, A. Amanullah, M. Shafarenko, and D.A. Liebermann. The proto-oncogene c-myc in hematopoietic development and leukemogenesis. *Oncogene*, 21(21):3414–3421, 2002.

[115] I.S. Lossos, D.K. Czerwinski, A.A. Alizadeh, M.A. Wechser, R. Tibshirani, D. Botstein, and R. Levy. Prediction of survival in diffuse large-B-cell lymphoma based on the expression of six genes. *New England Journal of Medicine*, 350(18): 1828–1837, 2004.

[116] L.M. Rimsza, M.L. LeBlanc, J.M. Unger, T.P. Miller, T.M. Grogan, D.O. Persky, R.R. Martel, C.M. Sabalos, B. Seligmann, R.M. Braziel, E. Campo, A. Rosenwald, J.M. Connors, L.H. Sehn, N. Johnson, and R.D. Gascoyne. Gene expression predicts overall survival in paraffin-embedded tissues of diffuse large B-cell lymphoma treated with R-CHOP. *Blood*, 112(8):3425–3433, October 2008.

[117] T. Akasaka, H. Akasaka, C. Ueda, N. Yonetani, Y. Maesako, A. Shimizu, H. Yamabe, S. Fukuhara, T. Uchiyama, and H. Ohno. Molecular and clinical features of non-Burkitt's, diffuse large-cell lymphoma of B-cell type associated with the c-MYC/immunoglobulin heavy-chain fusion gene. *Journal of clinical oncology*, 18(3): 510–510, 2000.

[118] U. Vitolo, G. Gaidano, B. Botto, G. Volpe, E. Audisio, M. Bertini, R. Calvi,

R. Freilone, D. Novero, and L. Orsucci. Rearrangements of bcl-6, bcl-2, c-myc and 6q deletion in B-diffuse large-cell lymphoma: clinical relevance in 71 patients. *Annals of oncology*, 9(1):55–61, 1998.

[119] W.Y. Au, D.E. Horsman, R.D. Gascoyne, D.S. Viswanatha, R.J. Klasa, and J.M. Connors. The spectrum of lymphoma with 8q24 aberrations: a clinical, pathological and cytogenetic study of 87 consecutive cases. *Leuk Lymphoma*, 45(3):519–28, 2004.

[120] L. Sun, P. Xie, J. Wada, N. Kashihara, F.Y. Liu, Y. Zhao, D. Kumar, S.S. Chugh, F.R. Danesh, and Y.S. Kanwar. Rap1b GTPase ameliorates glucose-induced mitochondrial dysfunction. *Journal of the American Society of Nephrology*, 19(12): 2293–2301, 2008.

[121] M.G. Vander Heiden, L.C. Cantley, and C.B. Thompson. Understanding the Warburg Effect: The Metabolic Requirements of Cell Proliferation. *Science*, 324(5930): 1029–1033, May 2009.

[122] P. Gao. Micrornas and cancer metabolism. In William C.S. Cho, editor, *MicroRNAs in Cancer Translational Research*, pages 485–497. Springer Netherlands, 2011. ISBN 978-94-007-0298-1.

[123] M.S. Kwon, S.-H. Shin, S.-H. Yim, K.Y. Lee, H.-M. Kang, T.-M. Kim, and Y.-J. Chung. CD63 as a biomarker for predicting the clinical outcomes in adenocarcinoma of lung. *Lung Cancer*, 57(1):46–53, 2007.

[124] I.S. Lossos, C.D. Jones, R. Warnke, Y. Natkunam, H. Kaizer, J.L. Zehnder, R. Tibshirani, and R. Levy. Expression of a single gene, BCL-6, strongly predicts survival in patients with diffuse large B-cell lymphoma. *Blood*, 98(4):945–951, 2001.

[125] L.C. Cerchietti, E.C. Lopes, S.N. Yang, K. Hatzi, K.L. Bunting, L.A. Tsikitas, A. Mallik, A.I. Robles, J. Walling, L. Varticovski, R. Shaknovich, K.N. Bhalla, G. Chiosis, and A. Melnick. A purine scaffold Hsp90 inhibitor destabilizes BCL-6 and has specific antitumor activity in BCL-6–dependent B cell lymphomas. *Nature Medicine*, pages 1–9, November 2009.

[126] L.C. Cerchietti, S.N. Yang, R. Shaknovich, K. Hatzi, J.M. Polo, A. Chadburn, S.F. Dowdy, and A. Melnick. A peptomimetic inhibitor of BCL6 with potent antilymphoma effects in vitro and in vivo. *Blood*, 113(15):3397–3405, April 2009.

[127] N.E. Ladendorff, S. Wu, and J.S. Lipsick. BS69, an adenovirus E1A-associated protein, inhibits the transcriptional activity of c-Myb. *Oncogene*, 20(1):125–132, 2001.

[128] H. Masselink and R.A. Bernards. The adenovirus E1A binding protein BS69 is a corepressor of transcription through recruitment of N-CoR. *Oncogene*, 19(12): 1538–1546, 2000.

[129] J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A.A. Margolin, S. Kim, C.J. Wilson, J. Lehr, G.V. Kryukov, D. Sonkin, A. Reddy, M. Liu, L. Murray, M.F. Berger, J.E. Monahan, P. Morais, J. Meltzer, A. Korejwa, J. Jan-Valbuena, F.A. Mapa, J. Thibault, E. Bric-Furlong, P. Raman, A. Shipway, I.H. Engels, J. Cheng, G.K. Yu, J. Yu, P. Aspesi, M. de Silva, K. Jagtap, M.D. Jones, L. Wang, C. Hatton, E. Palescandolo, S. Gupta, S. Mahan, C. Sougnez, R.C. Onofrio, T. Liefeld, L. MacConaill, W. Winckler, M. Reich, N. Li, J.P. Mesirov, S.B. Gabriel, G. Getz, K. Ardlie, V. Chan, V.E. Myer, B.L. Weber, J. Porter, M. Warmuth, P. Finan, J.L. Harris, M. Meyerson, T.R. Golub, M.P. Morrissey, W.R. Sellers, R. Schlegel, and L.A. Garraway. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483 (7391):603–307, March 2012.

[130] S. Mukherjee and T.P. Speed. Network inference using informative priors. *PNAS*, 105(38):14313, 2008.

[131] N. Chevrier, P. Mertins, M.N. Artyomov, A.K. Shalek, M. Iannacone, M.F. Ciaccio, I. Gat-Viks, E. Tonti, M.M. DeGrace, K.R. Clauser, M. Garber, T.M. Eisenhaure, N. Yosef, J. Robinson, A. Sutton,

M.S. Andersen, D.E. Root, U. von Andrian, R.B. Jones, H. Park, S.A. Carr, A. Regev, I. Amit, and N. Hacohen. Systematic Discovery of TLR Signaling Components Delineates Viral-Sensing Circuits. *Cell*, 147(4):853–867, November 2011.

[132] A.C. Gingras, M. Gstaiger, B. Raught, and R. Aebersold. Analysis of protein complexes using mass spectrometry. *Nature Reviews Molecular Cell Biology*, 8(8):645–654, 2007.

[133] P.J. Park. ChIP–seq: advantages and challenges of a maturing technology. *Nature Publishing Group*, 10(10):669–680, September 2009.

[134] R.B. Darnell. HITS-CLIP: panoramic views of protein-RNA regulation in living cells. *Wiley Interdisciplinary Reviews - RNA*, 1(2):266–286, August 2010.