Computational Methods for Analyzing and Modeling Gene Regulation and 3D Genome Organization

by

Anastasiya Belyaeva

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

Doctor of Philosophy in Computational and Systems Biology

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2021

Author . . . .
Program of Computational and Systems Biology
January 15, 2021

Certified by . . . .
Caroline Uhler
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Chris Burge
Director, Computational and Systems Biology Graduate Program

Computational Methods for Analyzing and Modeling Gene Regulation and 3D Genome Organization

by

Anastasiya Belyaeva

Submitted to the Program of Computational and Systems Biology
on January 15, 2021, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computational and Systems Biology

Abstract

Biological processes from differentiation to disease progression are governed by gene regulatory mechanisms. Currently large-scale omics and imaging data sets are being collected to characterize gene regulation at every level. Such data sets present new opportunities and challenges for extracting biological insights and elucidating the gene regulatory logic of cells. In this thesis, I present computational methods for the analysis and integration of various data types used for cell profiling. Specifically, I focus on analyzing and linking gene expression with the 3D organization of the genome.

First, I describe methodologies for elucidating gene regulatory mechanisms by considering multiple data modalities. I design a computational framework for identifying colocalized and coregulated chromosome regions by integrating gene expression and epigenetic marks with 3D interactions using network analysis. Then, I provide a general framework for data integration using autoencoders and apply it for the integration and translation between gene expression and chromatin images of naive T-cells.

Second, I describe methods for analyzing single modalities such as contact frequency data, which measures the spatial organization of the genome, and gene expression data. Given the important role of the 3D genome organization in gene regulation, I present a methodology for reconstructing the 3D diploid conformation of the genome from contact frequency data. Given the ubiquity of gene expression data and the recent advances in single-cell RNA-sequencing technologies as well as the need for causal modeling of gene regulatory mechanisms, I then describe an algorithm as well as a software tool, difference causal inference (DCI), for learning causal gene regulatory networks from gene expression data. DCI addresses the problem of directly learning differences between causal gene regulatory networks given gene expression data from two related conditions.

Finally, I shift my focus from basic biology to drug discovery. Given the current COVID19 pandemic, I present a computational drug repurposing platform that enables the identification of FDA approved compounds for drug repurposing and investigation of potential causal drug mechanisms. This framework relies on identifying drugs that reverse the signature of the infection in the space learned by an autoencoder and then uses causal inference to identify putative drug mechanisms.

Thesis Supervisor: Caroline Uhler
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to thank my advisor, Caroline Uhler, for her guidance and support over the course of my PhD. Caroline without exaggeration has been the best advisor one could imagine. Her expertise in a variety of fields from statistics to machine learning and biology and an always positive disposition made it a joy to work in her lab.

I would also like to thank my thesis committee members, David Sontag and Ernest Fraenkel, for being very engaged with my research and providing valuable feedback. My main collaborator G.V. Shivashankar and his lab members, most importantly Saradha Venkatachalapathy and Karthik Damodaran, have been invaluable to this research as well as great people to work with and wonderful hosts in Singapore. I would also like to thank Kaie Kubjas for her work and guidance on the 3D genome reconstruction project.

Thank you to all the members of Caroline's group, in particular, Karren D. Yang, Chandler Squires, Adit Radhakrishnan, Louis Cammarata and Yuhao Wang for collaborating on various projects and creating a positive atmosphere. I would also like to thank remarkable master's and undergraduate students that I had an opportunity to work with: Madeleine Duran, Abigail Katcoff and Basil Saeed.

I am indebted to the MIT community, which has been my home for the past 5 years. I am really thankful to have crossed paths with so many wonderful and inspiring people. Thank you to my program's administrator Jacquie Carota for keeping everything running, inquiring on my behalf regarding any questions I had and always offering a smile. I was lucky to meet exceptional women as part of the executive board of Graduate Women at MIT and blow off some steam with the IDSS hockey team. I thank Max Vilgalys and Marco Miotti for being wonderful office mates in IDSS.

Outside of MIT, I was lucky to come home to my incredible roommates at Marney Street: Angelina, Max, Sebastian, Sam, Mark, Alex, David, Chloe and Hendrik. Angelina, thank you for always being a ray of sunshine. Sam and Sebastian, thanks for starting many traditions from Sunday bread to Halloweens and Christmases. Alex,

thank you for the jokes, your questions and listening. I also want to thank friends that have been there all along: Vasudha Kowtha, Mandy Xu, Tom O'Connell and Tom Chih. Vasudha, thank you for your questions that always give me a bigger perspective on life. Tom Chih for the drop-on-the-floor laughter.

A special thank you goes to my boyfriend Max, who shared the ups and downs of graduate school life with me. I don't think I would have made it to the end without your support. You've been there to provide the much needed emotional support and care. Watching you work on your own PhD has given me perspective and made me feel less alone. As a peer, you have also given me solid scientific advice and helped me work though my own research problems, and I sincerely thank you for that.

Finally, I would like to thank my family for their love and support. They are the reason why I've been able to reach this milestone in my life. I love them above and beyond.

Contents

1 Introduction **19**

1.1 Model of gene regulation 20

1.1.1 Principles of 3D genome folding and role in gene regulation . . 21

1.2 Data types for cell profiling 22

1.3 Computational methods for learning regulatory logic 24

1.3.1 Inferring gene regulatory networks from gene expression 24

1.3.2 Learning regulatory logic beyond a single data modality . . . 26

1.4 Overview of the thesis 27

2 Network analysis identifies chromosome intermingling regions as regulatory hotspots for transcription **31**

2.1 Summary 31

2.2 Introduction 32

2.3 Results 34

2.4 Discussion 45

2.5 Future directions 49

2.6 Methods 50

3 Multi-Domain Translation between Single-Cell Imaging and Sequencing Data using Autoencoders **61**

3.1 Summary 61

3.2 Introduction 62

3.3 Results 63

3.4 Discussion 80
3.5 Future directions 81
3.6 Methods 82

4 Identifying 3D Genome Organization in Diploid Organisms via Euclidean Distance Geometry 93
4.1 Summary 93
4.2 Introduction 94
4.3 Unidentifiability from pairwise distance constraints 98
4.4 Distance constraints between neighboring loci 102
4.5 Identifiability from higher-order contact constraints 103
4.6 Algorithms and implementation 106
4.7 Evaluation on synthetic and real data 111
4.8 Discussion 115
4.9 Future directions 117

5 Learning Causal Differences between Gene Regulatory Networks 119
5.1 Summary 119
5.2 Introduction 120
5.3 Preliminaries 121
5.4 Difference Causal Inference (DCI) algorithm 122
5.5 Evaluation 130
5.6 Discussion 145
5.7 Future directions 146

6 Causal Network Models of SARS-CoV-2 Expression and Aging to Identify Candidates for Drug Repurposing 149
6.1 Summary 149
6.2 Introduction 150
6.3 Results 154
6.4 Discussion 168

6.5 Future directions 171

6.6 Methods 172

A Appendix for Network analysis identifies chromosome intermingling regions as regulatory hotspots for transcription 189

B Appendix for Multi-Domain Translation between Single-Cell Imaging and Sequencing Data using Autoencoders 199

C Appendix for Identifying 3D Genome Organization in Diploid Organisms via Euclidean Distance Geometry 215

D Appendix for Learning Causal Differences between Gene Regulatory Networks 233

E Appendix for Causal Network Models of SARS-CoV-2 Expression and Aging to Identify Candidates for Drug Repurposing 237

List of Figures

2-1 Overview of the proposed quantitative framework for detecting intermingling regions. 36

2-2 Number of chromosomes within an intermingling cluster. 38

2-3 Performance and feature importance for classifying intermingling regions. 40

2-4 Scatterplot of univariate rank of a feature versus predictive rank of a feature when it is combined with all other features. 41

2-5 Classification of intermingling regions into active and inactive clusters. 42

2-6 Transcription factor binding sites (TFBS) and GO terms across active clusters. 44

2-7 Experimental validation. 46

3-1 Schematic of multimodal data integration and translation strategy using our cross-modal autoencoder model. 65

3-2 Performance of our multimodal data integration method (cross-modal autoencoders), deep canonical correlation analysis (DCCA), Seurat, CycleGAN and MAGAN on paired RNA-seq and ATAC-seq data. . . 71

3-3 Analysis of single-cell RNA-seq data and single-cell chromatin images of naive CD4+ T-cells reveals two distinct subpopulations. 73

3-4 Integration of single-cell RNA-seq data and single-cell nuclear images of naive T-cells using our methodology allows translating between chromatin packing and gene expression profiles. 76

3-5 Validation of our model alignment using single-cell immunofluorescence experiments. 79

4-1 Schematic of the diploid genome. 97
4-2 Distance constraints between neighboring beads. 103
4-3 Higher-order distance constraints. 105
4-4 Examples of true and reconstructed points on simulated data. 112
4-5 Performance of our method in the noiseless setting. 112
4-6 Performance of our method in the noisy setting 113
4-7 3D diploid genome reconstruction. 116

5-1 Overview of DCI algorithm. 123
5-2 Example for the DCI algorithm. 127
5-3 Stability selection paths for the difference between two simulated DAGs 130
5-4 Proportion of consistently estimated difference-DAGs on simulated data. 132
5-5 High-dimensional evaluation of the DCI algorithm on synthetic data. 134
5-6 Effects of gene deletions estimated from CROP-seq data. 137
5-7 ROC plot evaluating DCI on CROP-seq data. 137
5-8 Examples of difference gene regulatory networks between naive and activated Jurkat T cells, estimated from the CROP-seq data. 139
5-9 Effects of gene deletions estimated from Perturb-seq data. 140
5-10 ROC plot evaluating DCI on Perturb-seq data. 141
5-11 Examples of difference gene regulatory networks of dendritic cells before and after stimulation with LPS, estimated from the Perturb-seq data. 142
5-12 Estimate of the difference-DAG between the two groups of ovarian cancer patients for the apoptosis and TGF-β pathways. 143
5-13 Run time of DCI, PC and GES on simulated data with variable input size. 144
5-14 Run time of DCI with varying tuning parameters. 145

6-1 Overview of computational drug repurposing platform for COVID-19. 153
6-2 Identification of differentially regulated genes in SARS-CoV-2 infection and aging. 156

6-3 Mining FDA approved drugs by correlating disease and drug signatures using an overparameterized autoencoder embedding. 161
6-4 Comparison of drug signature alignment between A549 and MCF7 and A549 and HCC515 cell types upon using an embedding verus the original space. 162
6-5 Drug target discovery via Steiner tree analysis to identify putative molecular pathways linking differentially expressed genes in SARS-CoV-2 infection and aging. 164
6-6 Causal mechanism discovery of potential drug targets. 167

A-1 Comparison of gene expression between active and inactive clusters. . 190
A-2 Top ranked clusters that appeared in the top 20 clusters across 8 different methods of evaluation. 191
A-3 Determining chromosomes that do not intermingle for predicting negative controls. 192
A-4 Experimental validation using FISH for predicted active cluster on chromosomes 12 and 17. 193
A-5 Experimental validation using FISH for predicted negative control. . . 194
A-6 Percentage of nuclei that intermingle versus do not intermingle for the chromosome pairs 12-17 and 3-20, as found by FISH experiments. . . 195
A-7 Adjusted mutual information between replicate clusterings. 195

B-1 Clustering of peripheral blood mononuclear cell (PBMC) data. 201
B-2 Evaluating optimal number of clusters for naive CD4+ T-cell gene expression data. 202
B-3 Examples of naive CD4+ T-cell nuclei stained with DAPI. 203
B-4 Evaluating optimal number of clusters for naive CD4+ T-cell imaging data. 204
B-5 Cross-modal autoencoder model trained on single-cell RNA-seq and single-cell images of DAPI-stained nuclei. 205
B-7 Validation of inferred latent embedding. 207

B-8 t-SNE visualization of latent space learned by convolutional autoencoder on chromatin imaging data of naive CD4+ T-cells. 208

B-9 Evaluation of robustness of ROC and LDA plots to the choice of architecture (fully-connected versus convolutional layers, number of layers, and latent space dimension) for the cross-modal autoencoder integrating RNA-seq and chromatin imaging. 209

B-10 Evaluation of the robustness of the differential gene expression analysis to the choice of architecture (fully-connected versus convolutional layers, number of layers, and latent space dimension) for the cross-modal autoencoder integrating RNA-seq and chromatin imaging. 210

C-1 Identifiability in the 2D setting with neighboring distance constraints. 216

C-2 Additional examples of true and reconstructed points on simulated data. 223

C-3 The impact of the number of tensor constraints in the noisy setting. . 223

C-4 The impact of ρ in the noisy setting. 224

C-5 Empirical choice of parameter for the distance between neighboring beads. 226

C-6 3D Diploid Genome Reconstruction with a different parameter for the distance between neighboring beads. 227

C-7 3D diploid genome reconstruction with ChromSDE. 228

C-8 Chromosome size versus the mean distance of the chromosome and its homolog away from the center. 229

C-9 The impact of ρ in real data. 229

C-10 3D diploid genome reconstruction with $\rho = 10$. 230

C-11 Haploid distance matrices. 231

D-1 ROC curves for estimating the difference-DAG with 10% percent change between DAGs. 234

D-2 Estimate of the difference-DAG between the two groups of ovarian cancer patients, obtained by PC and GES. 235

E-1 Detailed schematic of our computational drug repurposing platform. . 240

E-2 Analysis of gene expression of A549-ACE2 and A549 cell types with and without SARS-CoV-2 infection. 241

E-3 Analysis of gene expression data from individuals of different aging groups. 241

E-4 Analysis of differentially expressed genes associated with SARS-CoV-2 and aging. 242

E-5 CMap data with cell types and perturbations visualized. 242

E-6 Overview of autoencoder architectures, optimization methods and hyperparameter settings considered. 243

E-7 Receiver operating characteristic (ROC) curves for the agreement in classification between gene expression vectors and reconstructed gene expression vectors obtained using an embedding given by the first 2 principle components, the first 100 principle components, an underparameterized autoencoder and an overparameterized autoencoder. . . . 243

E-8 Quantitative analysis of similarity between drug lists obtained using the latent space embedding as compared to the original and PCA embedding (using 2 PCs). 244

E-9 Quantitative analysis of similarity between drug lists obtained using the overparameterized autoencoder on gene expression data from different MOIs for A549 cells with and without ACE2 receptor. 244

E-10 Terminal node selection for prize-collecting Steiner forest analysis. . . 245

E-11 Parameter selection via sensitivity analysis for prize-collecting Steiner forest analysis. 246

E-12 Interactome obtained from the prize-collecting Steiner forest algorithm (252 nodes and 1,003 edges). 247

E-13 Selection of the prize p for non-terminal SARS-CoV-2 interaction partners (all but EXOSC5, FOXRED2 and LOX) via sensitivity analysis. 248

E-14 Interactome obtained from the prize-collecting Steiner forest algorithm augmented with all other SARS-CoV-2 interaction partners (254 nodes and 1,090 edges). 249

E-15 2-Nearest-Neighborhoods of nodes of interest in the interactome that contains SARS-CoV-2 interaction partners. 250

E-16 Drug target discovery via prize-collecting Steiner forest analysis to identify putative molecular pathways linking differentially expressed genes in SARS-CoV-2 infection without taking into account age-related differential expression. 251

E-17 Evaluation of causal structure discovery algorithms GSP, PC and GES on CMap gene knockout and overexpression data. 252

E-18 Causal networks estimated for A549 and AT2 cells. 252

E-19 Differentially expressed genes for SARS-CoV-2, RSV and IAV infections. 253

E-20 Quantitative analysis of similarity between drug lists obtained using the overparameterized autoencoder on gene expression data from different virus infections. 253

E-21 Drugs and their gene targets obtained from the prize-collecting Steiner tree analysis for IAV and RSV infections in comparison to our findings for SARS-CoV-2. 254

E-22 Selection of correlation threshold for identifying candidate drugs. . . . 255

E-23 Comparison of drug targets resulting from analyzing the CMap dataset with and without removing confounding 1s. 255

List of Tables

A.1 Total sizes of intermingling regions 196

A.2 Datasets and accessions used to obtain the genomic features. 196

A.3 Fold enrichment of high-occupancy target (HOT) regions as compared to low-occupancy target regions (LOT) in active clusters. 198

A.4 Top 15 inactive clusters, ranked by enrichment for H3K9me3. 198

A.5 Top 15 active clusters, ranked by p-value of permutation test based on TFBS. 198

B.1 Network architecture for autoencoder network trained on RNA-seq and ATAC-seq data collected from A549 cells. 211

B.2 Losses and corresponding weights for autoencoder network trained on RNA-seq and ATAC-seq data collected from A549 cells. 211

B.3 Top 10 differentially upregulated genes (average log-fold change > 0) for each cluster in PBMC data set. 212

B.4 Network architecture for RNA-seq and image autoencoder networks. . 213

E.1 Percentage of nodes in the largest connected component of the corresponding causal graph that are targeted by each drug. 256

E.2 Frequency of a drug's presence in the list of final drugs after performing Steiner tree analysis with randomization of gene labels, CMap signatures, terminal genes, and the PPI network (1000 randomization runs). 260

Chapter 1

Introduction

Biological processes from differentiation to disease progression are governed by gene regulatory mechanisms. Although nearly every human cell contains the exact same DNA sequence, there are 200-500 major cell types that have different functions and phenotypes (Roy and Conroy, 2018). These differences arise due to different sets of gene regulatory programs turned on and off in different cell types. Since the advent of genome sequencing, large-scale "omics" data sets are being collected to characterize gene regulation at every level such as at transcriptome, proteome, epigenome and 3D genome conformation levels. Beyond "omics" data, imaging technologies also provide additional measurements on the function and phenotype of the cells. Both omics and imaging data sets are often large and complex and thus advanced computational methods are required for their analysis. In this thesis, we will focus on developing algorithms for analyzing and integrating data types with the goal of facilitating downstream derivation of new hypotheses and predictions that may enable a better understanding of the regulatory logic in the cells. In Section 1.1, we will describe different levels of gene regulation and the interplay between them with a specific focus on the 3D genome organization. In Section 1.2 we will discuss the types of data collected for the study of gene regulation and in Section 1.3 we will cover methods for inferring gene regulatory mechanisms relevant to this thesis. Section 1.4 will provide the outline for the thesis.

1.1 Model of gene regulation

The human genome, which contains the hereditary material of the cell, is composed of approximately 6 billion base pairs and is packed inside the cell nucleus. Protein-coding genes account for 1.5% of the genome with the remaining fraction of the genome associated with regulatory DNA elements, introns, non-coding RNA, repetitive DNA sequences and sequences with yet unknown function (Lander et al., 2001).

The activity of protein-coding genes is controlled by the dynamic interplay between transcription factors, which are proteins that bind to regulatory DNA elements, the epigenome and the 3D conformation of the genome (Stadhouders et al., 2019). As part of biological processes in the human body, cues from the environment trigger signaling pathways, which most prominently converge in the activation or silencing of transcription factors that are then translocated from the cytoplasm of the cell into the nucleus. Transcription factors bind to specific motifs in the DNA sequence, contained within regulatory DNA elements such as promoters or enhancers. Upon binding, transcription factors promote or preclude recruitment of the cellular machinery needed for transcription of gene DNA sequence into mRNA, for chromatin remodeling as well as for DNA and histone modifications (Vaquerizas et al., 2009). However, epigenetic marks such as histone modifications, DNA accessibility and DNA methylation may hinder the access of transcription factors to the DNA. For example, if the DNA is less accessible, the transcription factors would not be able to bind to the DNA (Klemm et al., 2019). Similarly, the 3D conformation of the genome determines which genes and regulatory DNA elements are in physical proximity and thus can act as a facilitator for the rapid activation of the colocalized genes or conversely as a barrier for activation (Bonev and Giacomo, 2016; Stadhouders et al., 2019; Uhler and Shivashankar, 2017a).

1.1.1 Principles of 3D genome folding and role in gene regulation

Since the 3D genome organization is a major player in regulating the cell state, in the following, we will describe the organizing principles of the 3D genome conformation, elucidated by several decades of research. In humans, the approximately two meters of genomic DNA is fit into a micron scale nucleus. Early studies of the nucleus via imaging with fluorescence in situ hybridization of DNA (DNA-FISH) have revealed that the nucleus is not homogeneous but is highly compartmentalized (Bolzer et al., 2005). Chromosomes occupy distinct chromosome territories and neighboring chromosome territories may loop out and intermingle with each other, which has been proposed to be relevant for various gene expression programs such as T-cell activation (Iyer et al., 2012), human antiviral response (Apostolou and Thanos, 2008), olfactory receptor choice (Lomvardas et al., 2006; Monahan et al., 2019), and globin gene activation in erythroid cells (Schoenfelder et al., 2010).

More recently, chromosome conformation capture methods, which measure contact frequencies between genomic loci, have revealed that chromosome territories spatially segregate into megabase pair-long A and B compartments, where A compartments are associated with transcriptionally active and open chromatin regions while B compartments are associated with trancriptional silencing, closed chromatin and repressive epigenetic marks (Lieberman-Aiden et al., 2009; Rao et al., 2014). During cell differentiation it has been shown that up to 35% of the genome switches compartments (Stadhouders et al., 2019; Dixon et al., 2015; Hu et al., 2018; Bonev et al., 2017). Several studies have proposed that segregation into compartments can be partially explained by phase separation, where proteins self-organize into 3D condensates that concentrate specific factors and exclude others, thereby enabling efficient expression of gene regulatory programs (Strom et al., 2017; Erdel and Rippe, 2018; Boija et al., 2018; Chong et al., 2018).

At higher resolution, the A and B compartments can be further divided into topologically associating domains (TADs), which contain DNA that tends to interact with itself more in comparison with DNA from other TADs (Dixon et al., 2012). Finally, chromatin loops, mainly linking DNA regulatory elements such as enhancers and promoters within TADs, can be observed. These loops are also highly cell-type-specific with 80% of promoters exhibiting cell-type-specific 3D interactions (Stadhouders et al., 2019).

Although the exact mechanisms by which the architecture of the genome has an effect and is in turn affected by gene expression are still being elucidated, studies have shown that gene expression and 3D organization are interlinked. For example, analysis of reprogramming of B lymphocytes into pluripotent stem cells revealed a tight coupling between chromatin state, 3D genome architecture and gene expression dynamics. In particular, genes that were positioned within A compartments (*Oct4*) were activated much faster than the ones in B compartments (*Sox2*) (Stadhouders et al., 2018), thereby providing an example of how the 3D genome organization can act as a barrier or facilitator for gene expression. In pluripotent cells, transcription factors such as NANOG have been observed to induce the spatial clustering of genes in 3D for robust maintenance of their expression levels (De Wit et al., 2013). Similarly, polycomb group proteins and their target *Hox* genes are clustered together in 3D to maintain the genes in a silent but poised state (Schoenfelder et al., 2015). Taken together, the 3D genome organization plays an important role in gene regulation and thus analysis linking and integrating this information with gene expression and epigenetic marks may provide a more systematic understanding of the processes inside the cell.

1.2 Data types for cell profiling

A variety of methods have been developed to probe the cell at different levels of gene regulation. In order to measure gene expression, RNA-sequencing (RNA-seq) has been developed in the mid 2000s (Emrich et al., 2007; Lister et al., 2008) and has become a routine part of many molecular biology studies. Quantification of gene expression by RNA-seq results in a genes × samples matrix of counts, where each entry indicates the number of reads mapped to the gene in the sample.

In order to characterize the chromatin state of DNA several methods have been developed such as chromatin immunoprecipitation sequencing (ChIP-seq), DNase I hypersensitive sites sequencing (DNase-seq) and assay of transposase accessible chromatin sequencing (ATAC-seq). ChIP-seq measures histone modifications of the DNA and transcription factor binding to the DNA (Johnson et al., 2007). DNase-seq and ATAC-seq measure chromatin accessibility (Boyle et al., 2008; Buenrostro et al., 2013). ChIP-seq, DNase-seq and ATAC-seq methods provide genome-wide signals and can be represented by a matrix of binned genomic loci × samples or by a list of peak locations for every sample. Currently, the majority of data from sequencing-based methods is collected in bulk and on just a few samples (e. g. 3 replicates), where each sample represents an average over all the cells in the sample. Such bulk measurements preclude studying covariation of signals and heterogeneity of cells in the sample. Recently, single-cell methods that measure the signal for each cell individually have been introduced both for measuring gene expression, e.g. single-cell RNA-seq, and chromatin state, e.g. single-cell ChIP-seq, DNase-seq and ATAC-seq (Rotem et al., 2015; Jin et al., 2015; Buenrostro et al., 2015; Cusanovich et al., 2015).

In order to measure the 3D genome organization, imaging and sequencing technologies have been developed. The most commonly used imaging technique is DNA-FISH, which allows measurement of physical distances between two or a few (e. g. 50) differentially labelled genomic loci of interest in single cells. More recently, sequencing-based methods such as chromosome conformation capture methods, in particular Hi-C (Lieberman-Aiden et al., 2009; Bickmore and Van Steensel, 2013; Schmitt et al., 2016; Dekker and Mirny, 2016), and related methods such as SPRITE (Quinodoz et al., 2018) and GAM (Beagrie et al., 2017) have been developed to probe the spatial organization of the entire genome over a population of cells by measuring contact frequencies between genomic loci. The output of these methods can be represented by a contact frequency matrix of binned genomic loci × binned genomic loci, where each entry represents the contact frequency between two loci. Similar to other sequencing-based methods, Hi-C has also been adapted to enable measurements in single cells

(Nagano et al., 2013).

1.3 Computational methods for learning regulatory logic

Comprehensive modeling of gene regulatory logic is one of the key challenges in modern molecular systems biology. In the following, we will describe several approaches for inferring gene regulatory mechanisms. We break up these approaches into two categories to set up the background and key concepts in this thesis.

1.3.1 Inferring gene regulatory networks from gene expression

Above we have described a complex model of gene regulation, taking into account epigenomics and the 3D conformation of the genome. Although experiments can measure some of these properties, simply utilizing just the gene expression data, which can be collected easily and cheaply, may enable us to reconstruct the gene regulatory networks governing the cells. In this case a gene regulatory network consists of p nodes corresponding to p genes and an edge between two nodes represents a relationship between the two genes. A variety of methods have been proposed to learn such gene regulatory networks (Wang and Huang, 2014). Commonly, the relationship between a pair of genes is defined by a measure of coexpression such as Pearson correlation (Langfelder and Horvath, 2008). Although the computational costs for calculating correlation-based coexpression networks are low, correlation can give misleading results, e. g. two genes might be highly correlated due to a confounding effect of a third gene that is related to both genes. As a result, Gaussian graphical models that capture partial correlations, which is a measure of association between two genes controlling for the effect of all other genes, have been proposed for modeling gene regulatory networks (Friedman et al., 2008). One major drawback of both coexpression networks and Gaussian graphical models is that the resulting graph is undirected and thus causal relationships cannot be represented. Since ultimately understanding a biological system means that one can predict the effect of perturbing that system, it is important that the inferred gene regulatory network can predict the effect of an intervention (e.g. small molecule, overexpression of a transcription factor, knock-out of a gene). This cannot be done using an undirected graph and necessitates modeling a gene regulatory network by a causal (directed) graph.

One of the most common frameworks for representing causal relationships are directed acyclic graphs (DAGs). We will use this framework for inference of gene regulatory networks in Chapters 5 and 6. Let $\mathcal{G} = ([p], E)$ be a DAG with nodes $[p] := \{1, \ldots, p\}$ and directed edges E. A random variable X_i is associated to each node $i \in [p]$. The data is assumed to be generated by a linear structural equation model with Gaussian noise:

$$X = B^T X + \epsilon, \tag{1.1}$$

where B is the weighted adjacency matrix of $\mathcal{G}$ and $\epsilon \sim \mathcal{N}_p(0, \Omega)$ with $\Omega = \text{diag}(\sigma_1^2, \cdots, \sigma_p^2)$. The goal is to learn the DAG $\mathcal{G}$ associated with the variables X.

A standard approach for causal structure discovery is to first infer the conditional independence (CI) relations among the observed variables and then use the CI relations to learn the DAG structure (Spirtes et al., 2000). However, since multiple DAGs can encode the same CI relations, $\mathcal{G}$ can only be identified up to its Markov equivalence class (MEC). An MEC can be represented by a CPDAG, a partially directed graph whose skeleton (underlying undirected graph) is the skeleton of $\mathcal{G}$ and an edge is directed if it has the same direction for all DAGs in the MEC (Verma and Pearl, 1990). Various algorithms have been developed for learning a CPDAG (Glymour et al., 2019; Chickering, 2002; Solus et al., 2017; Spirtes et al., 2000); most prominently the PC algorithm (Spirtes et al., 2000), which treats causal inference as a constraint satisfaction problem with the CI relations as constraints and GES (Chickering, 2002), which greedily searches over the space of MECs to maximize a score function.

1.3.2 Learning regulatory logic beyond a single data modality

When multiple data modalities are available, a number of approaches have been developed to extract biological insights by integrating the different data types. The exact scheme by which the data is integrated will depend on the goals of the analysis and the type of data. For the purposes of this thesis, we will divide the methods into non-exclusive categories of methods based on models, networks, correspondences between data sets and neural networks.

First, the data can be integrated using a statistical model that relates different data modalities. For example, the expression of a gene might be modelled as a function of the accessibility of the regulatory DNA element associated with the gene and the likelihood of the regulatory DNA element being recruited to the gene (Duren et al., 2017). However, the relationships between some modalities such as imaging and sequencing might be unknown or difficult to express by a model.

Second, the data might be integrated via a network where the structure of the network represents interactions (e. g. protein-protein interactions or interactions between genomic loci) and the values of nodes and edges correspond to data from other genomics experiments (Huang and Fraenkel, 2009; Tuncbag et al., 2012; Pancaldi et al., 2016). Network analysis can then be applied to identify a subnetwork of interest, prioritize nodes via network centrality measures or cluster the network to identify densely connected communities that might be functionally relevant. We use these ideas in Chapters 2 and 6.

Third, the data can be integrated using correspondences between samples or features of the datasets. Recent breakthroughs in single-cell technologies have allowed simultaneous profiling of multiple types of molecules within a single cell, resulting in datasets which share samples across the different modalities such as RNA-seq and ATAC-seq (Cao et al., 2018). In this case, methods such as canonical correlation analysis or factor analysis can be used for investigating the relationships between the two data sets (Gundersen et al., 2019; Argelaguet et al., 2018). However, in the typical case samples across the different modalities are not paired and thus some methods rely on correspondences between features such as shared markers or shared data structures of sequencing methods to integrate different modalities (Stuart et al., 2019; Lopez et al., 2019; Stanley et al., 2020; Liu et al.; Amodio and Krishnaswamy, 2018; Liu et al., 2020).

Finally, neural networks have been very successful in domain translation, in particular for image-to-image-translation (Zhu et al., 2017; Amodio and Krishnaswamy, 2019) and thus can also be applied to biological data to translate between different data types (Amodio and Krishnaswamy, 2018; Liu et al., 2020) and thus essentially obtain paired data (discussed in the previous paragraph). In Chapters 3 and 6 we will use autoencoders, which are a special type of neural network, for domain translation and learning a data representation. Briefly, an autoencoder is a neural network that attempts to reconstruct the input data and thereby learn a representation of the data in an unsupervised manner. Given training examples x, the autoencoder consists of an encoder E and a decoder D and is trained by minimizing the reconstruction loss $L(x, D(E(x))$, where L is typically the mean squared error (Baldi, 2012). Typically, the encoder maps the data to a low-dimensional representation and additional regularization may be incorporated into the loss function.

1.4 Overview of the thesis

In this thesis, we first describe methodologies for studying gene regulation by considering multiple data modalities (Chapters 2 and 3). Then, we focus on developing methods for extracting biological insights based on data from a single modality such as contact frequency data describing the spatial organization of the genome (Chapter 4) or gene expression data (Chapter 5). We end by considering how the developed methodologies and ideas can be used for drug repurposing against SARS-CoV-2 given the current COVID19 pandemic (Chapter 6). Chapters 2, 3 and 6 represent the main work of this thesis. The following provides a more detailed overview of each chapter.

In Chapter 2, informed by the models of gene regulation described in Section 1.1 and the role of spatial organization in the regulatory control of the cell, we design a computational framework for identifying chromosome regions that are colocalized and coregulated. Our framework is based on integrating 1D genomic features such as epigenetic marks with 3D interactions using network analysis. The colocalized and coregulated clusters of regions that we identify may be functionally important and could act as an additional layer of transcriptional regulation.

Our work in Chapter 2 provides an example of how data from different modalities can be used to provide novel insights into gene regulation. In the ideal setting, to derive a deep understanding of the cellular state and regulatory logic, from experiments we would be able to measure all aspects of the cell such as the cell's expression profile, proteomic profile, epigenome and 3D conformation, along with their full histories. However, obtaining high-throughput paired measurements of these different data modalities within single cells is still a major challenge requiring significant breakthroughs in single-cell technologies. In Chapter 3, we provide a general framework using autoencoders for data integration across different data modalities. Specifically, we address the gap of integrating modalities with vastly different data structures such as imaging and sequencing data. We apply our methodology for integrating and translating between gene expression and chromatin images of naive T-cells. Our method allows for hypothesis generation to predict the genome-wide expression profile of a particular cell given its chromatin organization and vice-versa. Such a methodology is valuable to understand how features in one dataset translate to features in the other.

While it is valuable to analyze multiple data modalities together, significant challenges still exist for extracting useful information even from a single modality. In Chapters 4 and 5, we focus on designing algorithms for data measuring the spatial organization of the genome and gene expression, respectively.

In Chapter 4, we consider the problem of reconstructing the 3D conformation of the genome from contact frequency data. This is challenging since the data obtained from sequencing technologies does not distinguish between the two homologous copies of DNA that are present in human cells. To alleviate this, we design an efficient algorithm for obtaining the 3D diploid genome configuration.

In Chapter 5, we consider having access only to gene expression data as a readout from a cell and we consider learning causal gene regulatory networks from such data. Unfortunately, application of classical causal inference algorithms to infer gene regulatory networks based on gene expression data is still challenging, requiring high sample sizes and computational resources. In Chapter 5, we circumvent this challenge by focusing on a related problem of learning the difference between gene regulatory networks between two related conditions, which is likely sparse. We propose an algorithm and provide a Python package that infers changes (i.e., edges that appeared, disappeared or changed weight) between two causal graphs given gene expression data from the two conditions.

Finally, while in most of this thesis we focus on designing algorithms that could enable us to understand the basic biology of the cell, in Chapter 6 we turn to using some of the computational frameworks discussed in this thesis such as autoencoders, networks and causal analysis for drug repurposing against SARS-CoV-2.

Chapter 2

Network analysis identifies chromosome intermingling regions as regulatory hotspots for transcription

Parts of this chapter were published as:

Belyaeva, A., Venkatachalapathy, S., Nagarajan, M., Shivashankar, G. V., & Uhler, C. (2017). Network analysis identifies chromosome intermingling regions as regulatory hotspots for transcription. Proceedings of the National Academy of Sciences, 114(52), 13714-13719.

My contributions were to design and implement the computational approach, perform data analysis and write the manuscript.

2.1 Summary

The 3D structure of the genome plays a key role in regulatory control of the cell. Experimental methods such as high-throughput chromosome conformation capture (Hi-C) have been developed to probe the 3D structure of the genome. However, it remains a challenge to deduce from these data chromosome regions that are colocalized and coregulated. Here, we present an integrative approach that leverages 1D functional genomic features (e.g., epigenetic marks) with 3D interactions from Hi-C

data to identify functional interchromosomal interactions. We construct a weighted network with 250-kb genomic regions as nodes and Hi-C interactions as edges, where the edge weights are given by the correlation between 1D genomic features. Individual interacting clusters are determined using weighted correlation clustering on the network. We show that intermingling regions generally fall into either active or inactive clusters based on the enrichment for RNA polymerase II (RNAPII) and H3K9me3, respectively. We show that active clusters are hotspots for transcription factor binding sites. We also validate our predictions experimentally by 3D fluorescence in situ hybridization (FISH) experiments and show that active RNAPII is enriched in predicted active clusters. Our method provides a general quantitative framework that couples 1D genomic features with 3D interactions from Hi-C to probe the guiding principles that link the spatial organization of the genome with regulatory control.

2.2 Introduction

The three-dimensional (3D) structure of the genome plays a key role in regulatory control of the cell. Historically, the spatial organization of the genetic material has been probed with fluorescence in situ hybridization (FISH), and it was shown that chromosome organization is nonrandom. Each chromosome occupies its own territory with gene-dense chromosomes more likely to be in the nuclear interior (Bolzer et al., 2005). As an addition to FISH, chromosome conformation capture methods (3C, 4C, 5C, and Hi-C) have been designed to probe the 3D organization of the genome by measuring the genome-wide contact frequencies over a population of cells (Lieberman-Aiden et al., 2009; Bickmore and Van Steensel, 2013; Schmitt et al., 2016; Dekker and Mirny, 2016). Computational and experimental efforts have largely focused on investigating intrachromosomal contacts. Studies where these interactions have been analyzed together with epigenetic modifications as measured by chromatin immunoprecipitation sequencing (ChIP-seq) showed that epigenetic marks are tightly linked to shaping the architecture of the genome (Dixon et al., 2012; Lan et al., 2012).

Few studies have considered interchromosomal interactions. It was shown that regions on neighboring chromosome territories may loop out and intermingle with each other in a transcription-dependent manner (Iyer et al., 2012; Branco and Pombo, 2006). In addition, a recent study has revealed that intermingling regions are enriched in both, active and repressive epigenetic marks, as well as the active form of RNA Polymerase II (RNAPII) and transcription factors (Maharana et al., 2016). Furthermore, it was identified that genes are spatially colocalized and coregulated by sharing common transcription factors (Schoenfelder et al., 2010; Papantonis et al., 2012) and epigenetic machinery like the polycomb proteins (Bantignies et al., 2011). For example, TNFα-responsive genes (on the same and different chromosomes) have been shown to colocalize upon their stimulation. Their spatial clustering was found to be correlated with their temporal expression patterns (Papantonis et al., 2012). The clustering of genes, transcriptional machinery, and regulatory factors to coordinate expression, also known as transcription factories, has been proposed as a model for gene regulation (Papantonis and Cook, 2013; Chen et al., 2015; Uhler and Shivashankar, 2017a). Collectively, these studies suggest that interchromosomal regions could harbor coregulated gene clusters. However, missing in this picture is a systematic analysis linking 1D epigenetic marks and 3D intermingling regions and their roles in transcription control.

Various methods have been developed to infer the spatial connectivity of the whole genome from Hi-C data. Restraint-based approaches transform Hi-C contact matrices into distances to deduce one consensus structure (Zhang et al., 2013; Lesne et al., 2014; Varoquaux et al., 2014; Segal and Bengtsson, 2015; Serra et al., 2015). However, it remains a challenge to map contact frequencies to spatial distances due to biases in Hi-C matrices (Imakaev et al., 2015). A different approach is to produce an ensemble of structures that could explain the experimental data (Wang et al., 2015; Tjong et al., 2016). Computational methods have largely focused on inferring the 3D genome structure based on Hi-C data alone without leveraging functional genomic data for studying its architecture. A recent study has explored this idea by superimposing ChIP-seq data of three transcription factors (TFs) on the 3D genome architecture inferred from Hi-C and determined functional hotspots in *Saccharomyces* *cerevisiae* (Capurso et al., 2016). Another study used 1D epigenomic tracks to predict 3D interactions (Zhu et al., 2016). But there remains a lack of a general quantitative framework that integrates 1D functional genomic features with 3D intermingling regions to determine a regulatory code for interchromosomal interactions.

In this study, we take a novel approach by integrating Hi-C and functional genomic data in order to predict regions that are colocalized and coregulated in 3D. The model of gene regulation that is captured by our analysis is the *spatial clustering of genomic regions for their coregulation* (Dekker and Misteli, 2015). This mode of gene regulation may enable the cell to coordinate gene expression and activate or repress pathways that are important for cell function in a coordinated manner. We focus on interchromosomal interactions to study chromosome intermingling regions. Using a network analysis approach, we construct a network of chromosomal interactions weighted by correlations in their genomic features at a 250kb resolution. We find that intermingling regions can be divided into active and inactive clusters, where active clusters are hotspots for transcription factor binding. We validate our predictions using FISH by comparing a predicted active cluster versus a predicted negative control and also confirm that active RNAPII is significantly enriched in the predicted active cluster.

2.3 Results

Identification of Intermingling Domains

In order to identify interchromosomal regions that are both spatially colocalized and coregulated, we leveraged spatial information from Hi-C experiments and regulatory information, namely, epigenetic marks, TF ChIP-seq, DNase I hypersensitivity (DNase-seq), and RNA-seq. Our aim was to identify clusters of chromosome regions at the whole genome scale that interact spatially due to similarities in their regulatory features and thus might be coregulated by shared regulatory factors and epigenetic marks. Our method consists of 4 steps outlined in Fig. 2-1: a) identification of highly interacting domains by determining large average submatrices in interchromosomal Hi-C maps, b) superimposing regulatory marks on the interacting domains, c) construction of a network of interacting regions with edges weighted by the correlation of the superimposed marks as a measure of coregulation, and d) network clustering to obtain spatially colocalized and coregulated domains.

We analyzed Hi-C data from IMR90 human lung fibroblast cells at 250kb resolution, obtained from (Rao et al., 2014). After bias correction, filtering, and transforming the data (Methods), we identified a stringent set of highly interacting interchromosomal regions by solving the following submatrix finding problem in Hi-C maps. We sought a contiguous submatrix $U(k \times l)$, that has a high average, τ, within the real-valued data matrix $X(m \times n)$, where each entry is an interchromosomal contact frequency between two 250kb regions. We used the iterative Large Average Submatrix (LAS) algorithm (Shabalin et al., 2009), that balances matrix size and average value, as outlined in *Methods* to discover highly interacting domains. Fig. 2-1a shows the identified domains in the Hi-C contact map for chromosomes 19 and 20. As shown in Fig. 2-1a, the LAS algorithm captures the regions with high intensity in the interchromosomal matrix. Applying this procedure to all pairwise interchromosomal maps yields Fig. 2-1b, where each entry in the matrix corresponds to the number of 250kb regions identified for the particular chromosome pair (FDR $< 4.16 \times 10^{-8}$, Methods). The total size of highly interacting domains across all chromosomes spanned 903.25 Mb (Supplementary Table A.1). Consistent with previous observations (Lieberman-Aiden et al., 2009; Kalhor et al., 2012), Fig. 2-1b shows that gene-dense chromosomes such as 15-17 and 19-22 had a high number of intermingling 250kb regions. In addition, as previously noted (Croft et al., 1999), we found a striking difference between chromosomes 18 and 19 - although these two chromosomes are approximately equal in size, the gene-poor chromosome 18 has a low level of intermingling across most chromosomes, while the gene-rich chromosome 19 tends to intermingle more with other chromosomes.

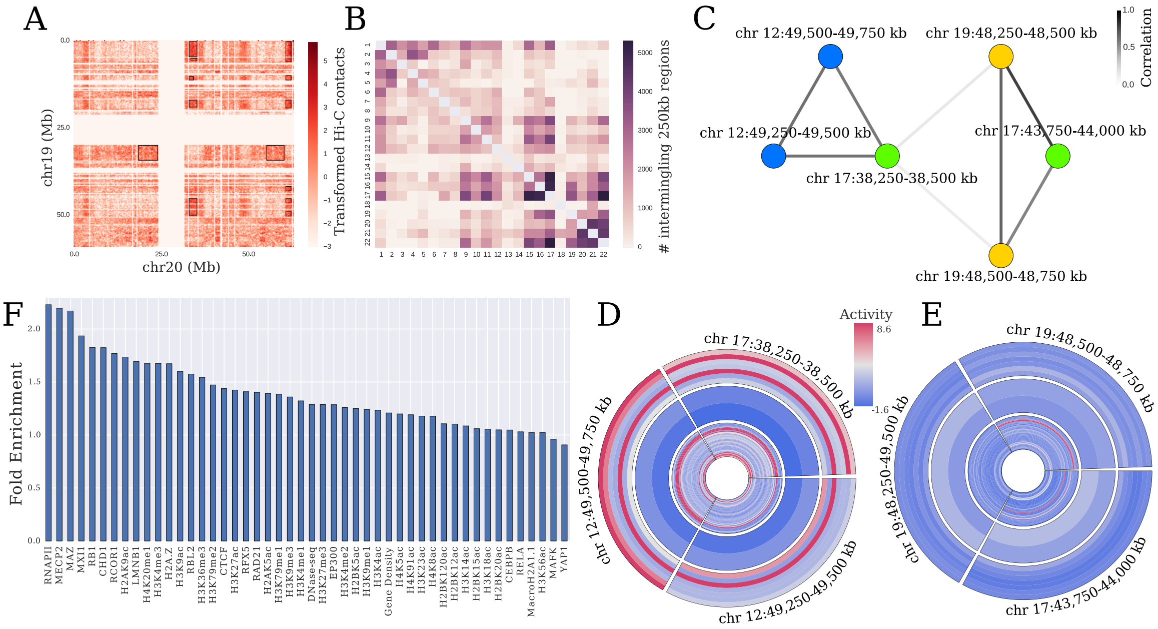


Figure 2-1: Overview of the proposed quantitative framework for detecting intermingling regions. A) Example of an observed interchromosomal Hi-C contact matrix at 250kb resolution after pre-processing and transformation (standardized by mean and standard deviation after log(1+x) transformation) for chromosome 19 and 20 (Methods). Rectangular boxes represent interacting domains for this pair of chromosomes as detected by the LAS algorithm, which finds submatrices with high average. B) Matrix containing the number of interacting 250kb regions identified by the LAS algorithm for each pair of chromosomes. C) Subnetwork of the chromosome interaction network corresponding to two distinct clusters. Nodes are colored by chromosome number. Each node in the network corresponds to a 250kb region. Edges link nodes that are found together in a submatrix (box) as determined by the LAS algorithm. The edge weights are given by the strength of correlation between the genomic features (histone modifications, TF ChIP-seq, DNase-seq, and RNA-seq as listed in Supplementary Table A.2) of adjacent 250kb nodes. D, E) Activity (normalized number of peaks in a 250kb region) of the genomic features for the two clusters obtained by weighted correlation clustering on the subnetwork in C. Each ring corresponds to 1 genomic feature, listed from outer ring to inner ring in Supplementary Table A.2. Features are grouped into active (outer rings - RNA-seq, RNAPII, H3K4me1, H3K4me2, H3K4me3, H3K36me3, H3K9ac), repressive (middle rings - H3K27me3 and H3K9me3), and other (inner rings) categories. F) Fold enrichment of each genomic feature in the intermingling regions (Methods).

Integration of Functional Genomic Data and Network Analysis

We obtained functional genomic data: TF ChIP-seq, histone modifications, DNase-seq, and RNA-seq data from ENCODE (Dunham et al., 2012), Roadmap Epigenomics (Kundaje et al., 2015), and GEO databases (Supplementary Table A.2). We used this experimental data as a regulatory profile for all 250kb regions that laid within the intermingling domains.

Considering each selected 250kb region as a node, a whole-genome network of chromosomal interactions was constructed as follows. Between chromosomes, the edges in the network were placed between pairs of 250kb regions that laid within the same submatrix as identified by the LAS algorithm. Within chromosomes, edges were placed between loci that fall within the same intrachromosomal domain, as determined in (Rao et al., 2014). After establishing the skeleton of the network, the edge weights were calculated as follows. Since our goal was to determine spatially coregulated regions, we weighted the edges by Spearman's correlation between the genomic profiles (e.g. gene expression, epigenetic marks, DNA accessibility, TF ChIP-seq) of adjacent 250kb regions. This combined approach can mitigate some of the noise associated with using Hi-C contact frequencies alone. In addition, it allows us to identify chromosome intermingling regions with coordinated activity, which might be controlled by the same set of transcription factors or epigenetic marks, as opposed to domains that interact in 3D by chance. A subnetwork containing six 250kb regions from 3 distinct chromosomes is shown in Fig. 2-1c. The edge weights in this subnetwork suggest the presence of two separate clusters.

In order to retrieve intermingling regions that are coregulated, the weighted network of 250kb regions was partitioned into clusters using weighted correlation clustering (Elsner and Schudy, 2009). This approach can for example identify regions that are brought together for transcription, since these would have high RNAPII and low repressive epigenetic marks. This approach indeed found two clusters in the subnetwork shown in Fig. 2-1c. The regulatory profiles of the 6 regions, separated into two clusters, are illustrated in Fig. 2-1d,e. As a consequence of using weighted correlation clustering, the genomic features within a cluster are more similar than across clusters. Interestingly, the particular cluster in Fig. 2-1d is enhanced for active genomic features (we analyzed H3K9ac, H3K36me3, H3K4me3, H3K4me2, H3K4me1, RNAPII, and RNA-seq) and depleted for repressive features (we analyzed H3K27me3 and H3K9me3), while the cluster in Fig. 2-1e is depleted for active features. Using this method, 446 clusters (totaling to 459.5 Mb, Supplementary Table A.1) were identified (p-value $< 2.2 \times 10^{-16}$ under χ^2-test) that consist of at least two nodes and span multiple chromosomes. On average, 2.5 chromosomes interact within one cluster (Fig. 2-2).

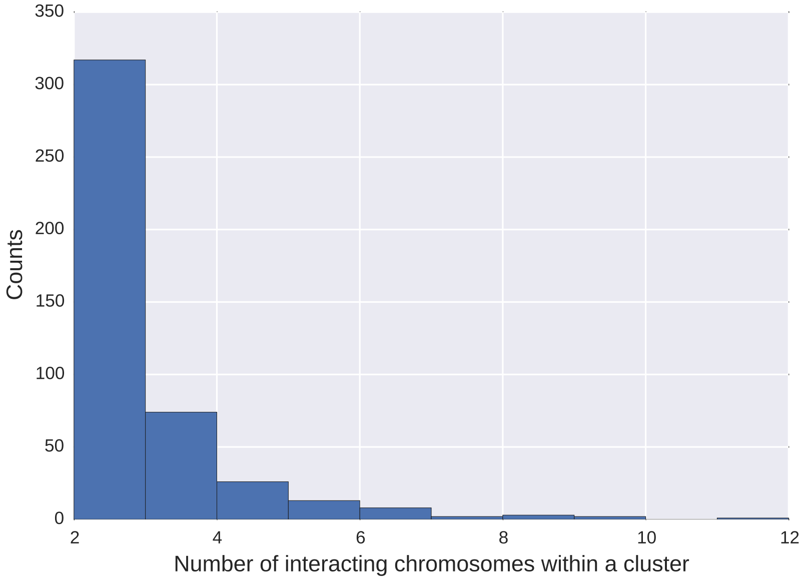


Figure 2-2: Number of chromosomes within an intermingling cluster.

We analyzed the enrichment of regulatory marks in intermingling regions and found that these regions were most enriched for RNAPII, namely by a factor of 2.23 (Fig. 2-1f). We also found the active and repressive marks (e.g., H3K9ac, H3K4me3, and H3K9me3) to be enriched in intermingling clusters, which is consistent with a previous study (Maharana et al., 2016).

Regulatory Features are Predictive of Intermingling

In order to characterize intermingling regions as a whole and evaluate whether they are distinct from non-intermingling regions on a regulatory level, we built a classifier and determined the features that contribute the most to distinguishing between these two classes. These features may represent a mechanism to spatially cluster genes for their coregulation. We annotated 250kb regions as intermingling or non-intermingling based on the results from our network analysis and clustering. We then performed classification based on the associated regulatory profiles (Methods, Supplementary Table A.2). We used eXtreme gradient boosting trees with 10-fold cross-validation to train our classifier. Using all features, the classifier achieves an accuracy of $85\% \pm 5\%$ and the corresponding ROC curve in Fig. 2-3a has an area under the curve (AUC) of 0.77.

To quantify the importance of each feature by itself and in conjunction with all other features, we computed its univariate and multivariate rank based on its depth in the decision trees of the ensemble (Fig. 2-3b and Fig. 2-4). The most important features determined by this analysis are lamin B1 (LMNB1), H3K9me3, H3K56ac, and H2A.Z. The importance of both repressive (H3K9me3, LMNB1) and active (H3K56ac, H2A.Z) marks ties with the observation that intermingling regions contain both active and repressed regions (Pombo and Dillon, 2015). Furthermore, previous mapping of LMNB1 in the genome revealed the presence of lamina-associated domains (LADs) that interact with the lamina on the nuclear envelope, spatially organize chromosomes by anchoring them to the lamina, and display coordinated gene repression (Camps et al., 2015; Guelen et al., 2008a; Finlan et al., 2008). H3K9me3 is enriched in LADs and may facilitate gene silencing in LADs (Guelen et al., 2008a; Shachar et al., 2015). The context-dependent importance of this feature is in line with its low univariate, but high multivariate rank (Fig. 2-4). H3K56ac is a known mark of transcriptionally active chromatin regions (Stejskal et al., 2015; Das et al., 2009). Finally, H2A.Z is enriched at transcription start sites (Barski et al., 2007), indicating its involvement in transcription initiation, and it appears to be a defining feature of intermingling on its own (Fig. 2-4).

Performing step-wise feature elimination shows that approximately 13 features are sufficient for achieving high AUC (Fig. 2-3c) and the corresponding features are annotated by stars in Fig. 2-3b.

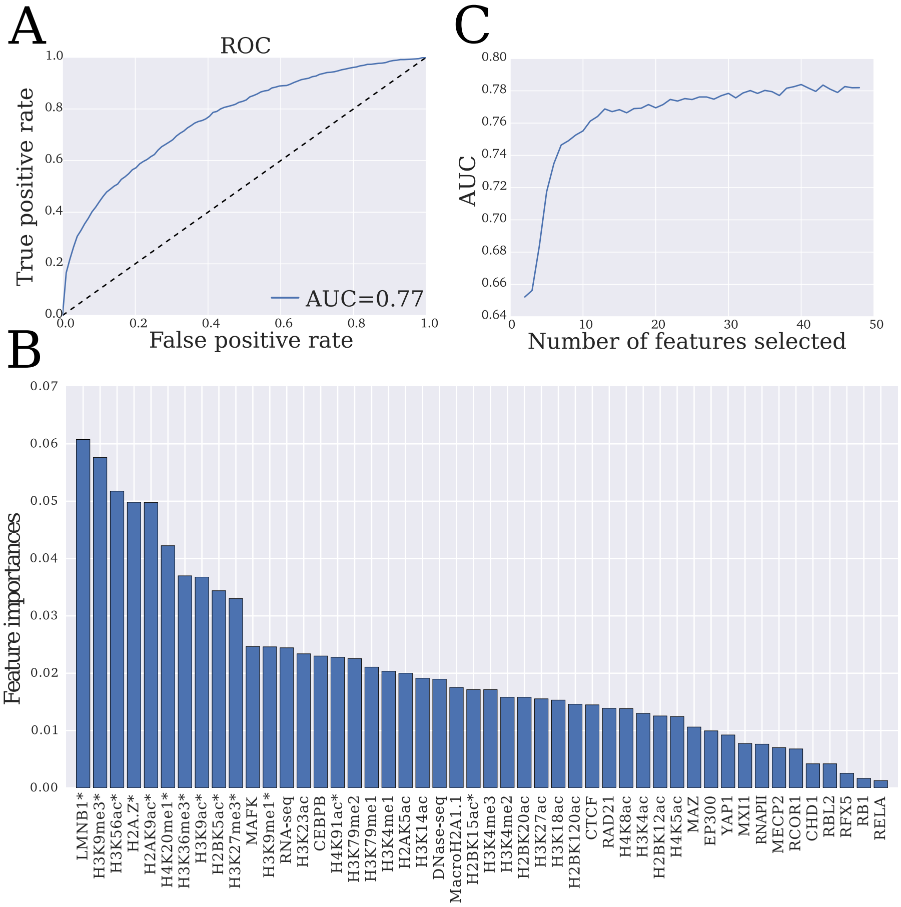


Figure 2-3: Performance and feature importance for classifying intermingling regions. A) ROC curve for eXtreme gradient boosting trees classifier that was trained on genomic features of intermingling versus non-intermingling regions. This results in AUC of 0.77. B) Features ranked in the order of importance (relative depth of feature in the decision tree) for distinguishing intermingling domains. C) AUC when recursively eliminating one feature at a time based on 10-fold cross-validation. Near-optimal performance is reached with 13 features, which are indicated by stars in B.

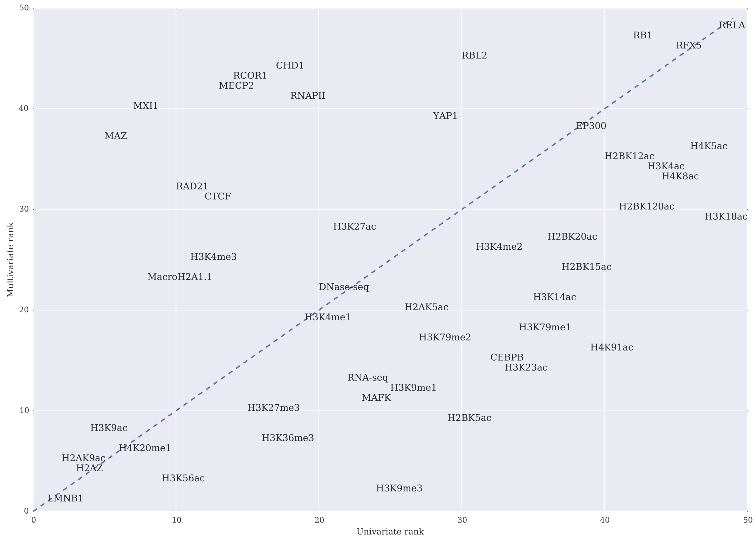


Figure 2-4: Scatterplot of univariate rank of a feature (two-sample Kolmogorov–Smirnov test) versus predictive rank of a feature when it is combined with all other features (relative importance in decision trees) for classification of intermingling versus non-intermingling regions. A similar analysis has been performed in (Whalen et al., 2016) to explore context-dependency of features for classification.

Intermingling Clusters are Divided into Active and Inactive Clusters

While it is interesting to evaluate intermingling regions altogether, studying these on a cluster-by-cluster level may give insights into the links between regulatory processes and spatial colocalization. Based on previous evidence (Simonis et al., 2006) we hypothesized that active regions are clustered with other active regions and inactive regions with other inactive regions. In order to analyze the types of clusters we obtained, we computed the fold enrichment of each cluster for several regulatory features. We found that a high proportion of the clusters - 41.7% (186 clusters) - was enriched for all active marks - RNAPII, H3K9ac, H3K36me3, H3K4me3, and H3K4me1 as shown in Fig. 2-5a (p-value $= 1.398 \times 10^{-5}$ under χ^2-test, Methods). Notably, the majority of clusters were either enriched for all 5 active marks or not enriched for any active mark.

The percentage of clusters enriched for the repressive/inactivating mark H3K9me3 was 38.3% (171 clusters). Interestingly, we observed a clear separation of the intermingling clusters into active and inactive, with only 4% of clusters (18 clusters) that were in both categories as shown in Fig. 2-5b (p-value $= 4.699 \times 10^{-4}$ under χ^2-test, Methods). Active clusters were defined as those clusters enriched for RNAPII (fold enrichment > 1) but not for H3K9me3. Inactive clusters were defined as enriched for H3K9me3 but not for RNAPII. Active clusters also had significantly higher gene expression (p-value $= 0.004$ under t-test) in comparison to inactive clusters (Supplementary Fig. A-1). In addition, high-occupancy target (HOT) regions, i.e. regions that are occupied by many TFs (Li et al., 2016), were overrepresented in active clusters in comparison to low-occupancy target (LOT) regions, by HOT:LOT ratio of 2.94 (Supplementary Table A.3). These findings suggest that active clusters may be hotspots for transcription factor binding.

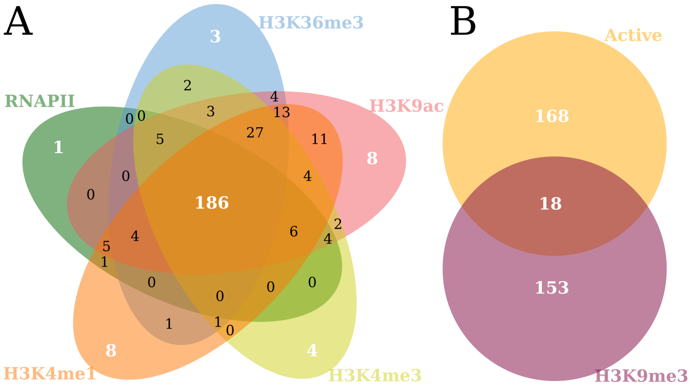


Figure 2-5: Classification of intermingling regions into active and inactive clusters. A) 5-way Venn diagram representing the number of clusters enriched for each active epigenetic mark and RNAPII. Interestingly, many clusters (186 out of 446) are enriched for all 5 active marks. B) Venn diagram of the active clusters (the 186 clusters in the intersection of the 5-way diagram in A) and clusters enriched for the silencing mark H3K9me3. Note that only 18 out of 446 clusters are both active and silenced, showing that the clusters separate into 2 categories of active and inactive clusters.

Active Clusters are Hotspots for Transcription Factor Binding

We probed the active clusters for shared transcription factors that may be involved in colocalizing and coregulating regions in a cluster by analyzing transcription factor binding sites (TFBS). We used the JASPAR 2016 database to obtain the TFBS. This data was overlayed and then filtered using ChIP-seq peaks from all human cell lines available from ENCODE (Dunham et al., 2012) (Methods). This resulted in TFBS for 52 transcription factor motifs. We performed an additional analysis to also consider a larger set of transcription factor motifs (386) by overlaying and filtering the JASPAR 2016 database with a robust set of CAGE peaks from (Forrest et al., 2014), collected across 353 human tissue samples as part of the FANTOM5 project (Methods). This filtering step provided us with a list of potential transcription start sites that contain motifs for the TFs under consideration.

We compared the distributions of TFBS counts per 250kb region for active clusters versus whole genome. Several factors, such as EGR1, YY1, CTCF, and the E2F family of proteins showed a significant increase in TFBS counts under a Mann-Whitney U-test (Fig. 2-6a).

The majority of active clusters contained binding sites for TFs that are shared across regions spanning multiple chromosomes. For example, the cluster studied in Fig. 2-1d involving chromosome 12 and 17 contains binding sites for the TFs USF1 and NRF1 on regions of both chromosomes (Fig. 2-6b). This cluster is formed by the colocalization between two adjacent 250kb regions on chromosome 12 and one region on chromosome 17. Gene ontology (GO) term analysis of the expressed genes in this cluster revealed an enrichment for biological processes related to fibroblasts such as "cytoskeleton dependent intracellular transport" (Fig. 2-6c). On the other hand, we found that inactive clusters contained a low number of TFBS (Supplementary Table A.4), reaffirming the existence of two distinct types of cluster categories for intermingling regions.

Experimental Validation

We ranked the active clusters according to the presence of binding sites for TFs that were shared across multiple chromosomes using a permutation test (Methods). The top 15 active clusters are shown in Supplementary Table A.5. Chromosomes 12 and 17 were consistently found together among the top highly ranked clusters and were thus chosen for experimental validation (Supplementary Fig. A-2). We compared the

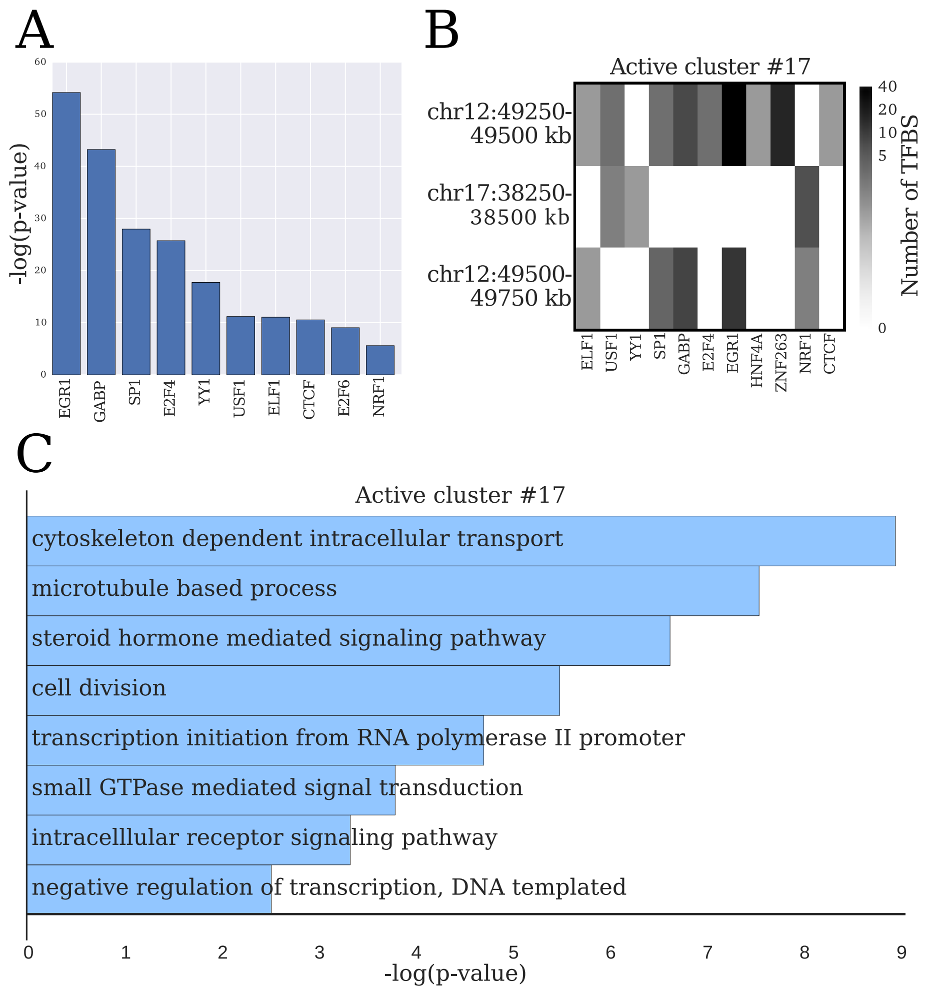


Figure 2-6: Transcription factor binding sites (TFBS) and GO terms across active clusters. A) Top 10 TFs with significantly overrepresented TFBS in active clusters as compared to the whole genome distribution (under Mann-Whitney U test). B) Matrix corresponding to a representative active cluster with number of TFBS for each 250kb region in the cluster. Only TFs containing at least one nonzero column entry are shown. A transcription factor shared among multiple regions in the cluster may indicate its role in colocalization and coregulation of the clustered regions. C) Significantly enriched GO terms computed from the genes that are expressed and colocalized in intermingling cluster shown in B. GO terms were ranked by p-value using DAVID (Huang et al., 2009b,a).

amount of overlap between chromosomes 12 and 17 to a negative control that we obtained by analyzing the network of least interacting chromosomes (Supplementary Fig. A-3). The chromosome territories were identified in BJ fibroblast cells using DNA FISH and visualized using a laser scanning confocal microscope (Fig. 2-7a-f). In order to obtain a representative sample of the population, we imaged at least 200 cells for each chromosome pair. We confirmed that chromosomes 12 and 17 consistently intermingle in a population of cells (Fig. 2-7c, Supplementary Fig. A-4), while the negative control chromosome pair does not (Fig. 2-7f, Supplementary Fig. A-5). To quantify our results, the intermingling degree, i.e., the amount of overlap between the two pairs of chromosome territories, was calculated as explained in *Methods.* We found that the chromosome pair 12 and 17, which was predicted to interact, had a significantly higher intermingling degree than the negative control pair 3 and 20 (Fig. 2-7g, p-value = 0.005 under Welch Two Sample t-test). The percentage of nuclei that were intermingling (intermingling degree $>$ 0) was higher in the predicted pair of interacting chromosomes, 12 and 17, than in the negative control, 3 and 20 (Supplementary Fig. A-6). In addition, we also calculated the enrichment of active RNAPII in the intermingling regions for the aforementioned pairs (*Methods*). We found that the predicted chromosome pair, 12 and 17, which belongs to an active cluster, had significantly higher enrichment for active RNAPII in the intermingling regions as compared to the negative control pair, 3 and 20, (Fig. 2-7h, p-value = 7.125e-05 under Welch Two Sample t-test), showing that the chromosome pair 12 and 17 indeed contains an active mark at the site of intermingling.

2.4 Discussion

Understanding the spatial organization of the chromosomes within the cell nucleus has been a major question in cell biology. A number of studies have suggested that the packing of DNA plays a critical role in regulating genomic programs (Bickmore and Van Steensel, 2013). Earlier experiments took advantage of chromosome painting methods and revealed that chromosomes are organized nonrandomly and in a

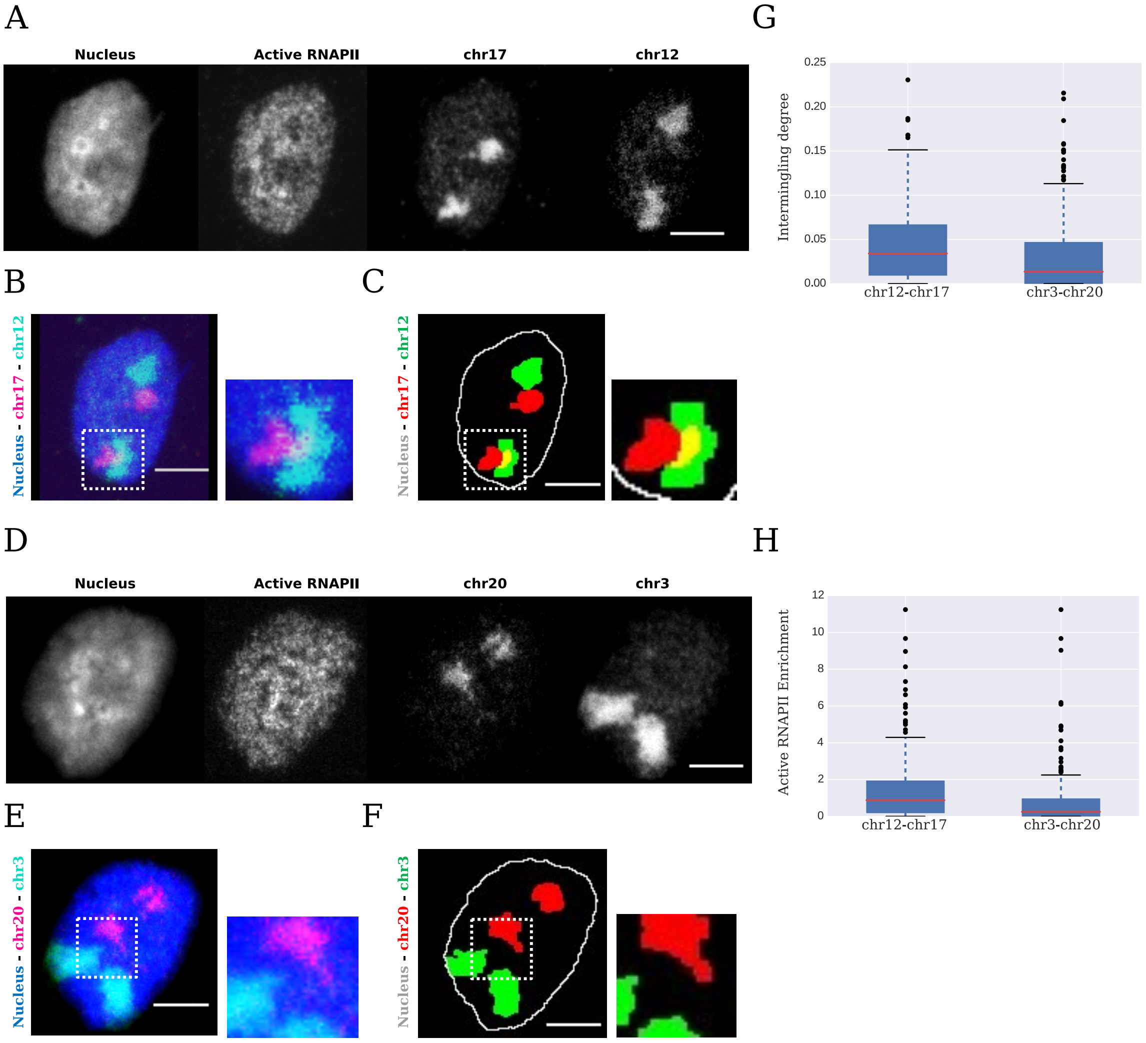


Figure 2-7: Experimental validation. A) Representative images of the maximum intensity Z projections of the nucleus, active RNAPII, and chromosomes 17 and 12, from left to right, respectively. B) Raw image resulting from merging the nuclear (blue) and the two chromosome channels depicting the overlap between chromosomes 17 (purple) and 12 (cyan). C) Image in B) after segmentation with nucleus (white), chromosome 17 (red) and chromosome 12 (green). Yellow regions are the overlapping or intermingling regions. The region in the dotted white boxes has been enlarged. D) Representative images of the maximum intensity Z projections of the nucleus, active RNAPII, and chromosomes 20 and 3, from left to right, respectively. E) Raw image resulting from merging the nuclear (blue) and the two chromosome channels depicting the overlap between chromosomes 20 (purple) and 3 (cyan). F) Image in E) after segmentation with nucleus (white), chromosome 20 (red) and chromosome 3 (green). The region in the dotted white boxes has been enlarged. G) Boxplot depicting intermingling degree between chromosomes 12 and 17 and chromosomes 3 and 20 (p-value = 0.005 under Welch Two Sample t-test). H) Boxplot depicting the enrichment of active RNAPII between chromosomes 12 and 17 and chromosomes 3 and 20 (p-value = 7.125e-05 under Welch Two Sample t-test). The scale bar has a length of 5μm.

cell-type specific manner (Bolzer et al., 2005; Branco and Pombo, 2006; Iyer et al., 2012). Analysis of gene positioning using FISH showed that coregulated genes were coclustered (Papantonis et al., 2012; Schoenfelder et al., 2010). Such clusters of genes were also found to be colocalized with transcription-related machinery such as active RNAPII and TFs (Papantonis et al., 2012; Schoenfelder et al., 2010). Recent developments in chromosome capture technologies further revealed that genome-wide chromosome contact maps are correlated with epigenetic marks (Dixon et al., 2012; Lan et al., 2012). The majority of studies using chromosome conformation capture focused on linking chromatin contacts with epigenetic modifications at the resolution of genes in intrachromosomal regions (Dixon et al., 2012; Lan et al., 2012). However, the coupling between the global organization of chromosomes with genome-wide epigenetic marks and the intermingling regions as an additional layer of transcriptional regulation has not been well studied.

In this study, we developed a network analysis approach to reveal the principles of transcription-dependent chromosome intermingling by taking advantage of 3D contact maps obtained using Hi-C and 1D epigenetic marks, TF ChIP-seq, DNA accessibility, and RNA-seq. Our computational approach focuses on interchromosomal domains, since their organizational principles have been largely unknown. The proposed quantitative framework enables the prediction of chromosome intermingling regions at a genome-wide scale, thereby complementing experimental methods such as FISH that can be used to study specific clusters of interchromosomal interactions. The novelty of our method lies in leveraging 1D genomic features in combination with 3D interactions from Hi-C data. This allows us to study functionally colocalized regions: since interactions can occur by chance in 3D, some intermingling regions may not be of biological relevance. By leveraging epigenetic marks and data from TF binding, DNA accessibility, as well as gene expression, we can determine interchromosomal regions that are colocalized and coregulated.

Our predictions reveal intriguing patterns of chromosome organization and have been validated by FISH experiments. Our findings recapitulate known principles of chromosome interactions, such as the tendency of gene-dense chromosomes to intermingle more frequently (Lieberman-Aiden et al., 2009; Kalhor et al., 2012) and the enrichment of RNAPII in intermingling regions (Maharana et al., 2016), suggesting that RNAPII may play a crucial role in establishing and maintaining chromosome interactions. We observe that the clusters of interchromosomal regions fall broadly into two categories, active and inactive, where active clusters are enriched for active epigenetic marks and RNAPII and inactive clusters are enriched for H3K9me3. Interestingly, we found that active clusters are hotspots for TF binding sites, with several TFs being shared among multiple chromosomes within a cluster. These clusters contain genes with biologically relevant GO terms. We established the predictive power of our model through experimental validation. Using FISH experiments we showed that the predicted intermingling chromosomes interact consistently across a population of cells and that such intermingling regions are enriched for active RNAPII. Our quantitative analysis provides evidence that TF hotspots in active clusters are colocalized with active epigenetic modifications, RNAPII, and have a significantly higher gene expression than inactive clusters, suggesting that the relative positioning of the chromosomes in the cell nucleus is optimized to facilitate the clustering of coregulated genes, TFs, epigenetic modifications, and transcriptional machinery.

Collectively, these findings suggest that the spatial organization of the genomic material in the cell nucleus is optimized for transcription programs. The framework we present here is general and can be applied to analyze any cell type. We showed by experimentally validating the predictions from our model using single-cell imaging methods that population-level genome-wide contact and epigenetic data carries enough information to identify highly interacting regions. However, we anticipate that the power of our method will be increased as more robust single-cell genomic data becomes available.

We believe that our quantitative approach will provide a useful framework to gain insights into the interplay between chromosome reorganization and regulation during processes such as cell differentiation, reprogramming, or the maintenance of homeostasis.

2.5 Future directions

While previous approaches have focused on the identification of interacting regulatory regions within a chromosome such as A and B compartments, TADs and loops (Lieberman-Aiden et al., 2009; Rao et al., 2014), we developed a methodology aimed at identifying interchromosomal colocalized and coregulated regions of the DNA. While our analysis was performed at 250kb resolution due to the sparsity of the interchromosomal contact frequency matrices, it would be interesting to apply our framework at higher resolution to provide a more fine-grained analysis when more high-resolution data becomes available. It would be particularly interesting from a biological perspective to perform such an analysis at the resolution of promoters, where transcription factors bind.

Our framework relied on combining the Hi-C contact frequency data with 1D genomic profiles. The LAS algorithm was necessary for obtaining highly interacting domains between chromosomes. While a simpler approach such as thresholding the Hi-C contact frequency data could have been applied, this would likely lead to missing interactions. Since interchromsomal matrices at 250kb resolution are noisy, a particular pair of loci might show a contact frequency below the threshold while the adjacent loci might have high interaction frequencies. Since adjacent loci on the string of DNA likely have similar 3D interactions, such interactions can be picked up using LAS but would be missed using thresholding. While we used the LAS algorithm for detecting colocalized regions, in future work it would be interesting to explore or develop other approaches with faster run time or use scoring functions that extend beyond Gaussian data.

After applying LAS to detect highly interacting regions, we used correlations between 1D genomic profiles to filter out regions that are colocalized in 3D for reasons unrelated to regulation (e.g. regions that are colocalized and coregulated in 3D force other nearby regions to be in proximity that are not coregulated). We used epigenetic marks, gene expression, DNA accessibility and TF ChIP-seq data to obtain a metric of coregulation between two regions. However, this simple metric may miss many nuances in biology. For example, it is known that enhancers and promoters are associated with different epigenetic marks and thus a simple correlation metric may not capture their regulatory relationship. Therefore, it would be of interest to use prior knowledge or discover directly from the data an accurate metric quantifying coregulation between loci on the DNA.

While we applied our methodology to a single cell type (fibroblasts), in future work it would also be of interest to apply our framework to different cell types and analyze the similarities as well as differences between the identified clusters of co-regulated regions and relate these findings to cell function.

2.6 Methods

Obtaining Hi-C matrices

The Hi-C matrices were obtained from (Rao et al., 2014) at 250kb resolution. Matrices were corrected for bias using interchromosomal matrix balancing based on the Knight-Ruiz algorithm using software in (Durand et al., 2016). Centromeric regions, as well as peri-centromeric regions within 2Mb of the centromere were filtered out. Repeat regions, outliers based on row and column sums (outside of 1.5 $\times$ interquartile range interval) in the Hi-C contact matrix, and regions already masked in (Rao et al., 2014), were removed from the analysis. The final Hi-C matrix was $\log(1 + x)$ transformed and normalized by mean contact frequency and standard deviation, computed over all interchromosomal contact pairs that were not filtered out.

LAS Algorithm for identification of highly interacting regions

The LAS algorithm (Shabalin et al., 2009) takes a real-valued data matrix $X(m \times n)$ as input and outputs contiguous submatrices $U(k \times l)$ that have a high average, τ. This is done via the following iterative algorithm:

Repeat until $\tau\sqrt{kl} < threshold$:

1. Search: greedily, by updating one row and column at a time, find a submatrix

U that maximizes the submatrix score

$$S(U) = -\log\left[(m-k+1)(n-l+1)\Phi(-\tau\sqrt{kl})\right] \tag{2.1}$$

2. Remove: identify rows and columns corresponding to U in X and subtract the submatrix average τ from this set of rows and columns.

The LAS algorithm search space was limited to contiguous submatrices of at most 10Mb × 10Mb in size, i.e. 40×40 submatrices (at 250kb resolution). For each chromosome pair each iteration of the search procedure was initialized at a random contiguous $k \times l$ submatrix in the interchromosomal Hi-C map.

The threshold for the algorithm was chosen based on a Gaussian approximation such that $P(\tau\sqrt{kl} > \mathit{threshold}) = 1E - 15$. This stringent cutoff guarantees that highly interacting regions in the whole-genome Hi-C map are identified with FDR controlled at 4.16×10^{-8} (see the following paragraph). Returning submatrices U, as determined by LAS, for each interchromosomal contact matrix results in a list of highly interacting pairs of 250kb regions.

FWER and FDR computation for Large Average Submatrix (LAS) algorithm

The LAS algorithm takes a real-valued matrix $X(m \times n)$ as input and outputs contiguous submatrices $U(k \times l)$ of high average (Shabalin et al., 2009). The null hypothesis is that the interchromosomal Hi-C matrix is a standard Gaussian random matrix, and the alternative hypothesis is that the interchromosomal Hi-C matrix is a sum of K constant (> 0) submatrices plus a standard Gaussian random matrix, i.e., that the Hi-C contact matrix contains substructure (Lieberman-Aiden et al., 2009). More precisely, each entry in the alternative model can be expressed as

$$x_{i,j} = \sum_{k=1}^{K} \alpha_k I(i \in A_k, j \in B_k) + \epsilon_{ij}, \tag{2.2}$$

where $A_k \subseteq [m]$ and $B_k \subseteq [n]$ are the row and column sets of the kth submatrix, α_k is the constant corresponding to the kth submatrix, ϵ_{ij} are independent noise variables sampled from $\mathcal{N}(0,1)$, and $I(\cdot)$ is the indicator function. Note that $K = 0$ corresponds to the null model. The individual entries in the $\log(1+x)$ transformed Hi-C matrices have an R^2 of 0.972 with the standard normal distribution, which justifies using a standard Gaussian random matrix as null hypothesis.

Let $\tau := Avg(U)$, i.e., the average of the submatrix U. Under the null hypothesis, $\tau\sqrt{kl} \sim N(0,1)$ and thus the probability of observing a $k \times l$ submatrix V with an average of τ or greater is $P(Avg(V) \geq \tau) = \Phi(-\tau\sqrt{kl})$, where Φ is the standard normal cdf. Let A denote the event that there exists a $k \times l$ submatrix V with average greater than or equal to τ in an $m \times n$ matrix. Note that this event is bounded as follows: $P(A) \leq \sum P(Avg(V) \geq \tau)$, where the sum is over all $k \times l$ submatrices in the $m \times n$ matrix. Hence, under the null hypothesis, $P(A) \leq N\Phi(-\tau\sqrt{kl})$, where $N = (m-k+1) \times (n-l+1)$, the total number of contiguous submatrices of size $k \times l$ in an $m \times n$ matrix.

The search space of the LAS algorithm was limited to contiguous submatrices of at most 10Mb × 10Mb in size, which corresponds to 40×40 submatrices (at 250kb resolution). In order to calculate the total number of hypotheses for each interchromosomal matrix, we summed the number of possible contiguous submatrices for all combinations of k and l within the [1,40] range. Considering all pairs of interchromosomal matrices, the total number of hypotheses was 9.33×10^{10}. In our procedure, we applied a p-value threshold, namely $P(\tau\sqrt{kl} > \mathit{threshold}) = 1 \times 10^{-15}$, for the discovery of significant submatrices. Using a formulation based on the Bonferroni correction, we can estimate the familywise error rate (FWER), which is the probability of making at least one type I error. Let p be the p-value threshold, b the number of hypotheses and α the FWER level. The Bonferroni correction rejects the null hypothesis when p-value $\leq \frac{\alpha}{b}$, thereby controlling the FWER at $\leq \alpha$. With our p-value threshold of 1×10^{-15}, the FWER is ≤ 0.0000933.

We can also calculate the false discovery rate (FDR), i.e. the fraction of false discoveries among all discoveries, using the Benjamini–Hochberg procedure. Let a be the number of discoveries, b the number of hypotheses, p the p-value threshold, and α the FDR level. The Benjamini–Hochberg procedure rejects the null hypothesis when p-value $\leq \frac{a}{b}\alpha$, thereby controlling the FDR at $\leq \alpha$. With our p-value threshold of 1×10^{-15} and the resulting number of discoveries $a = 2244$, the FDR is $\leq 4.16 \times 10^{-8}$.

Genomic features

Pre-processed data for 48 features including histone modifications, transcription factor ChIP-seq, DNase-seq and RNA-seq, were retrieved from ENCODE (Dunham et al., 2012), Roadmap Epigenomics (Kundaje et al., 2015), the GEO database, and previous studies (Whalen et al., 2016) (Supplementary Table A.2) for the IMR90 cell line. In order to obtain the genomic profile for a 250kb region, matching the resolution of Hi-C data, the number of peaks overlapping the 250kb region was calculated for each feature. For each feature, the feature matrix was $\log(1 + x)$ transformed and z-scored by computing the mean and standard deviation of the feature across all regions in the genome that were not removed by Hi-C filtering step.

Weighted Correlation Clustering

The weighted network of 250kb regions was partitioned into clusters using weighted correlation clustering on networks (Elsner and Schudy, 2009). This method determines clusters by drawing cluster boundaries across edges with low weights but not across edges with high weights by solving a non-convex minimization problem. Weighted correlation clustering was run in 25 replicates and the clustering with the lowest value of the objective function was chosen for further analysis. The resulting clusterings were robust across replicate runs, as evidenced by the high adjusted mutual information between cluster labels across runs (Supplementary Fig. A-7).

Classification into Intermingling and Non-intermingling Domains

In order to identify features that may be important for chromosome intermingling, a binary classification task was performed. The training and test data consisted of genomic feature profiles (Supplementary Table A.2) for intermingling versus non-intermingling regions, weighted by the number of samples in each class. Classification was done using eXtreme gradient boosting trees with `n_estimators=1000`, `learning_rate=0.1`, `max_depth=5`, `min_child_weight=1` with 10-fold cross-validation. Feature importances were computed by the relative rank of a feature in the decision tree, calculated via `feature_impotances_` function in scikit-learn (Pedregosa et al., 2012) in python. Additionally, features were evaluated using iterative feature elimination by removing one feature at a time and optimizing the AUC.

Fold Enrichment of Genomic Features

Fold enrichment for the intermingling regions as well as for specific clusters was calculated as follows:

$$\frac{\# \text{ bases in cluster and having feature}}{\# \text{ bases in genome}} \Bigg/ \left(\frac{\# \text{ bases in cluster}}{\# \text{ bases in genome}}\right)\left(\frac{\# \text{ bases having feature}}{\# \text{ bases in genome}}\right) \tag{2.3}$$

A fold enrichment of 1 indicates that the two events - belonging to the intermingling regions or a particular cluster and belonging to a particular feature - are independent events.

Comparison to a random network - Stochastic Block Model

To analyze the importance of the spatial interactions for the function and properties of the determined clusters, we performed a comparison based on a "similar" network in which the spatial interactions have been randomized. To be more precise, we generated a network from a stochastic block model, where each chromosome is a community and the edge probabilities within and between communities are computed from the number of interactions in the Hi-C matrix as determined by the LAS algorithm. In order to obtain similar cluster sizes as in the original network, we sampled the edge weights from the observed distribution of edge weights.

Using a 2-sided χ^2-test we tested whether the proportion of intermingling regions in the observed network was equal (H_A: not equal) to the proportion of intermingling regions in a random network. Generating 50 networks from the stochastic block model and using the average proportion of intermingling regions as test statistic, the null hypothesis was rejected with a p-value $< 2.2 \times 10^{-16}$.

Further, in order to test the functional relevance of the determined clusters, we tested using a 1-sided χ^2-test whether the proportion of clusters enriched for all five active marks (RNAPII, H3K9ac, H3K36me3, H3K4me3, H3K4me1) in the observed network was equal (H_A: larger than) in a random network. As in the previous test, generating 50 networks from the stochastic block model and using the average proportion as test statistic, the null hypothesis was rejected with a p-value of 1.398×10^{-5}.

Finally, we tested the regulatory event that active and inactive clusters are spatially separated. To do this, we tested using a 1-sided χ^2-test whether the proportion of clusters enriched for all five active marks and the inactive mark (H3K9me3) in the observed network was equal (H_A: smaller than) in a random network. Following the same procedure as in the previous test, the null hypothesis was rejected with a p-value of 4.699×10^{-4}.

Gene ontology

Expressed genes for IMR90 with reads per kilobase of transcript per million mapped reads (RPKM) > 0 were obtained from ENCODE (Dunham et al., 2012). For each cluster, we identified the genes that resided in the cluster and were expressed. For each cluster, we then performed gene ontology (GO) term analysis on these genes using DAVID (Huang et al., 2009b,a).

Cluster ranking

In order to select clusters for experimental validation, we ranked each cluster based on the number of TFBS present in each region in the cluster. Several methods and databases were used for ranking the clusters in order to choose a robust set of clusters for experimental validation. The whole genome was scanned for TFBS using position frequency matrices (PFMs) from the JASPAR2016 database for humans. MOODS software (Korhonen et al., 2016) was used to identify motif matches. TFBS were further filtered by ChIP-seq from ENCODE (Dunham et al., 2012), resulting in 52 TFs or robust CAGE peaks, resulting in 386 TFs. The CAGE peaks, which indicate transcription start sites, were obtained from the FANTOM5 project (Forrest et al., 2014), which pooled CAGE analysis over 573 human cell samples. These peaks were flanked by 400bp upstream and 50bp downstream as suggested by (Marbach et al., 2016) and overlayed with the TFBS data to obtain the final set of TFBS.

In the following, we explain how we used the determined TFBS to rank the clusters based on a permutation test. First, we constructed a score function to compare observed and randomized matrices. For each cluster we construct a matrix Z of size $m \times n$ consisting of the TFBS counts for each of the m 250kb domains that are clustered together for each of the n transcription factors that were analyzed. The number m may change from cluster to cluster, while the number of considered transcription factors n is the same for all clusters. Let A be the set of TFs that have TFBS on multiple chromosomes in the considered cluster. Then the score function for each cluster is computed as follows:

$$Score(Z) = \sum_{j \in A} \sum_{i=1}^{m} Z_{ij} \tag{2.4}$$

For each matrix Z, a set of 1000 random matrices is generated to compute the background score distribution. Assuming that the number of TFBS for a specific transcription factor is independent of the other transcription factors, a random matrix for a particular cluster with corresponding matrix Z is generated by the following procedure:

1. Let k denote the number of nonzero entries in Z. The probability of having a nonzero entry for each of the n transcription factors is defined by p_j, where $p_j = \frac{\text{\# nonzero entries for TF}_\text{j}}{\text{total \# of nonzero entries}}$. The number of nonzero entries for each tran-

scription factor, x_j, is drawn from a multinomial distribution, $(x_1, \cdots, x_n) \sim Mult(k, p_1, \cdots, p_n)$.

2. After determining the number of nonzero entries for each transcription factor, these nonzero entries must be distributed across the m clustered 250kb regions. Let q_i be the probability of assigning a nonzero entry to that specific region, where $q_i = \frac{\#\text{ nonzero entries in region}_\text{i}}{k}$. For each transcription factor j, the number of nonzero entries for each region, $(y_{1j}, \cdots, y_{mj})$ is drawn from a multinomial, $(y_{1j} \cdots y_{mj}) \sim Mult(x_j, q_1, \cdots, q_m)$.

3. By now the positions of nonzero entries within the randomly generated matrix have been chosen, and only the number of TFBS (counts) remain to be assigned to each of the k nonzero entries. For each transcription factor, samples are drawn from the observed count distribution for that transcription factor over active clusters.

Finally, the p-value of the observed score was computed using the background score distribution that we obtained by calculating a score for each of the 1000 randomly generated matrices described above. In order to ensure stability of the ranking procedure, the background distribution was computed in 10 replicates, resulting in 10 different p-values. We observed that the p-values across different runs were consistent. For each replicate, we obtained the cluster rankings based on their p-values. The final rank of each cluster was computed from the median rank across the 10 replicate runs.

Negative controls - chromosomes that do not intermingle

As negative controls, we identified by a whole-genome analysis analysis pairs of chromosomes that do not intermingle (in Hi-C) and are anti-correlated in terms of genomic features (Fig. A-3). First, we determined chromosome pairs for which the LAS analysis did not result in any intermingling regions. These chromosomes formed the nodes of a network with edges drawn between pairs of chromosomes with no intermingling regions. The weight of the edges was calculated as $1 - |\rho|$, where ρ is the correlation between the genomic features averaged over the whole chromosome. Chromosomes 3 and 20 were chosen as a negative control pair, since they were representative of the network of non-intermingling chromosomes.

Cell Culture and Chromosome FISH

BJ fibroblast cells were cultured in Low Glucose DMEM (Life Technologies, USA) supplemented with 10% (vol/vol) FBS (GIBCO, Life Technologies, USA) at 37C in 5% CO2. BJ fibroblast cells were cultured overnight on fibronectin-coated cleaned glass slides. Cells were then washed with 1× PBS to remove cell culture medium followed by incubation on ice for 5-8 minutes, with 0.25% Triton in CSK buffer (100 mM NaCl, 300 mM Sucrose, 3 mM MgCl2, 10 mM PIPES with pH 6.8). Cultured cells were fixed with 4% PFA (Paraformaldehyde) for 10 minutes, briefly rinsed with 0.1 M Tris-HCl and washed with 1× PBS wash. This was followed by permeabilization with 0.5% Triton for 10-15 minutes. The cells were then incubated overnight in 20% glycerol at 4C and subjected to 5-6 freeze-thaw cycles in liquid nitrogen. After this, cells were washed with 1× PBS a few times, before and after treatment with 0.01% HCl for 5-10 minutes, followed by protein digestion with 0.002% porcine pepsin (Sigma Aldrich, USA) in 0.01N HCl at 37C for 4 minutes. Cells were then fixed with 1% PFA for 4 minutes, briefly rinsed in 1× PBS before being treated with RNAse (Promega, USA, 200 microgram/ml made in 2× SSC-0.3M sodium chloride and 30mM trisodium citrate) at 37C for 15-20 minutes to digest RNA. The cells were then washed with 2× SSC and equilibrated in 50% Formamide / 2 SSC (pH 7.4) overnight at 4C. Hybridization was set up the following day. Chromosome fish probes (Chrombios, Germany) tagged with different fluorophores were thawed to room temperature and mixed with hybridization buffer provided by the supplier. The DNA was denatured in 50% Formamide / 2× SSC at 85C for 2-3 minutes and then incubated with the fluorescently labeled human chromosome FISH probe mix. The slides were then sealed with a Sigmacote (Sigma Aldrich, USA) coated hydrophobic coverslip and rubber cement to incubate for 18-48 hours in a moist chamber at 37C with shaking. At the end of the incubation period, slides were washed three times in 50% Formamide / 2× SSC at

45C and 0.1× SSC at 60C. After the last stringent wash with 50% Formamide made in 0.1× SSC at 45C, the nuclei were blocked in 5% BSA solution made in 2× SSC and then subjected to primary and secondary antibodies diluted in 5% BSA solution made in 2× SSC. In case indirect labels such as chromosome probes conjugated with biotin and digoxigenin (DIG), were used during hybridization detection step, the procedure also involved the use of fluorophore labeled streptavidin/avidin and anti-DIG. The primary antibodies used here were: RNAPII CTD repeat YSPTSPS (phospho S5) (Abcam - ab5131, 1:500 dilution), mouse monoclonal (21H8) to DIG (Abcam-ab420; 1:500 dilution). Finally, the nuclei were stained with Hoechst 33342 (Sigma Aldrich, USA) for 10 minutes and then mounted with Prolong Gold antifade mounting medium (Life Technologies, USA), sealed with a coverslip, and imaged.

Confocal Imaging and Image Analysis

Slides for chromosome FISH were scanned using a Nikon A1 Confocal microscope (Nikon, USA) with a 100×, 1.4 NA oil objective. Stacks of 12-bit gray scale two-dimensional images were obtained with a pixel size of 130 nm in XY direction and 500 nm in the Z direction and used for the quantitative evaluation. The image analysis was performed using a custom code in ImageJ2. The code first identified the nuclear boundary using Otsu 3D thresholding method. This was followed by the identification of chromosome territories in the nuclear region using ReyniEntropy 3D thresholding. The threshold for identifying signal and background in each image was determined using the intensity histogram from the 3D image stack. The thresholded image was binarized. The overlapping region between two chromosomes, i.e. the intermingling region (IMR), was identified by performing the AND function over the 3D binary stacks of both chromosomes. The chromosome and IMR volumes were computed by summing up the volumes of the non-zero voxels in the respective binary images. The intermingling degree was calculated by dividing the volume of the IMR between two chromosomes by the total volume of the two chromosomes. Similarly, the amount of active RNAPII in the nucleus and the IMR was obtained by passing the RNAPII image and the binary images of the nucleus and IMR through the AND filter, respectively. The enrichment of active RNAPII in the intermingling regions was obtained by dividing the mean intensity of active RNAPII in the IMR by the mean intensity of the active RNAPII in the entire nucleus. R was used for testing statistical significance and for data visualization.

Code availability

The code for interchromosomal network construction via LAS and for the identification and analysis of clusters is available at `http://github.com/anastasiyabel/functional_chromosome_interactions`. The code for performing the image analysis is available at `http://github.com/SaradhaVenkatachalapathy/Chromsome-intermingling-region-indentifcation-and-characterisation-of-protein-levels`.

Chapter 3

Multi-Domain Translation between Single-Cell Imaging and Sequencing Data using Autoencoders

Parts of this chapter have been accepted for publication to Nature Communications:

Dai Yang, K.*, Belyaeva, A.*, Venkatachalapathy, S., Damodaran, K., Radhakrishnan, A., Katcoff, A., Shivashankar, G. V., & Uhler, C. (2020). Multi-Domain Translation between Single-Cell Imaging and Sequencing Data using Autoencoders.

* These authors contributed equally. My contributions were to implement the models, design and perform model and data analysis, and write the manuscript.

3.1 Summary

The development of single-cell methods for capturing different data modalities including imaging and sequencing have revolutionized our ability to identify heterogeneous cell states. Different data modalities provide different perspectives on a population of cells, and their integration is critical for studying cellular heterogeneity and its function. While various methods have been proposed to integrate different sequencing data modalities, coupling imaging and sequencing has been an open challenge. We here present an approach for integrating vastly different modalities by learning

a probabilistic coupling between the different data modalities using autoencoders to map to a shared latent space. We validate this approach by integrating single-cell RNA-seq and chromatin images to identify distinct subpopulations of human naive CD4+ T-cells that are poised for activation. Collectively, our approach provides a framework to integrate and translate between data modalities that cannot yet be measured within the same cell for diverse applications in biomedical discovery.

3.2 Introduction

Recent evidence has highlighted the importance of the 3D organization of the genome to regulate cell-type specific gene expression programs (Uhler and Shivashankar, 2017b; Zheng and Xie, 2019). High-throughput and high-content single-cell technologies have provided important insights into genome architecture (using imaging and chromosome capture methods) (Finn et al., 2019; Stevens et al., 2017; Ramani et al., 2017) as well as detailed genome-wide epigenetic profiles and expression maps (using various sequencing methods) (Klein et al., 2015; Macosko et al., 2015; Buenrostro et al., 2015). However, obtaining high-throughput paired measurements of these different data modalities within single cells is still a major challenge requiring significant breakthroughs in single-cell technologies.

Different data modalities provide different perspectives on a population of cells and their integration is critical for studying cellular heterogeneity and its function (Fig. 3-1a). Current computational methods allow the integration of datasets of the same modality (Butler et al., 2018; Haghverdi et al., 2018; Trong et al., 2020) or of different modalities with the same data structure such as various sequencing measurements (Stuart et al., 2019; Lopez et al., 2019; Stanley et al., 2020; Liu et al.). We here present a computational framework based on autoencoders for integrating and translating between different data modalities with very distinct structures. Several works have proposed using autoencoders for domain adaptation (in particular batch correction) in the context of biological data (Lopez et al., 2018; Amodio et al., 2019). Different from these works, our method uses autoencoders to integrate and translate between different data modalities that may have very different representations. A separate line of work has proposed using neural networks to directly translate between pairwise modalities in an unsupervised manner (Amodio and Krishnaswamy, 2019; Zhu et al., 2017) or with side information (Liu et al., 2020; Amodio and Krishnaswamy, 2018). These methods tend to focus on modalities with similar representations (e.g., image-to-image-translation) and directly translate between pairs of modalities without learning a common latent representation of the data. In contrast, our work maps each data distribution to a common latent distribution using an autoencoder. This not only enables data integration and translation between arbitrary modalities in a globally consistent manner, but, importantly, it also enables performing downstream analysis such as clustering across multiple modalities at once. Other work has proposed coupled autoencoders to translate between paired biological data (Gundersen et al., 2019), which differs from our method that does not require paired data. Building on Makhzani et al., 2015, we align the latent space of an autoencoder using adversarial training and leverage this technique for data integration and/or translation. In particular, our framework can be applied to integrate and translate imaging and sequencing data, which cannot yet be obtained experimentally in the same cell, thereby providing a methodology for hypothesis generation to predict the genome-wide expression profile of a particular cell given its chromatin organization and vice-versa. Such a methodology is valuable to understand how features in one dataset translate to features in the other.

3.3 Results

Cross-modal autoencoders: Multi-domain data integration and translation using autoencoders

To integrate and translate between data modalities with very distinct structures, we propose a new strategy of mapping each dataset to a shared latent representation of the cells (Fig. 3-1a-b). This mapping is achieved using autoencoders (Baldi, 2012;

LeCun et al., 2015; Ngiam et al., 2011), neural networks consisting of an encoder (mapping to the latent space) and a decoder (mapping back to the original space), whose architectures can be customized to the specific data modality (Fig. 3-1b-c). Combining the encoder and decoder modules of different autoencoders enables translating between different data modalities at the single-cell level (Fig. 3-1d). To enforce proper alignment of the embeddings obtained by the different autoencoders, we employ a discriminative objective function to ensure that the data distributions from the different modalities are matched in the latent space. When prior knowledge is available, an additional term in the objective function can be used that encourages the alignment between specific markers or the anchoring of certain cells. In the following, we formally introduce our framework.

We formalize the multi-modal data integration problem within a probabilistic framework. Each modality or dataset presents a different view of the same underlying population of cells. Formally, we consider cells from each modality $1 \leq i \leq K$ as samples of a random vector X_i that are generated independently based on a common latent random vector Z:

$$X_i = f_i(Z, N_i), \quad \forall i = 1, \ldots, K, \tag{3.1}$$

where f_i are deterministic functions, Z has distribution P_Z, and N_i are noise variables. The domain of Z, here denoted by $\mathcal{Z}$, represents some underlying latent representation space of cell state, and each function f_i represents a map from cell state to data modality i. For simplicity of notation, we assume for the remainder of this section that each X_i is 1-dimensional and obtained via a deterministic function of Z, so that the noise variables N_i can be ignored. This model implies the following factorization of the joint distribution $P_{\mathbf{X}}$ (with density $p_{\mathbf{X}}$) of the data over all modalities:

$$p_{\mathbf{X}}(\mathbf{x}) = \int_{\mathcal{Z}} \Pi_{i=1}^{K} \, p_{X_i|Z}(x_i|z) p_Z(z) dz, \tag{3.2}$$

where p_Z is the probability density of Z, and $p_{X_i|Z}$ is the conditional distribution of X_i given Z that reflects the generative process. Multi-modal data integration

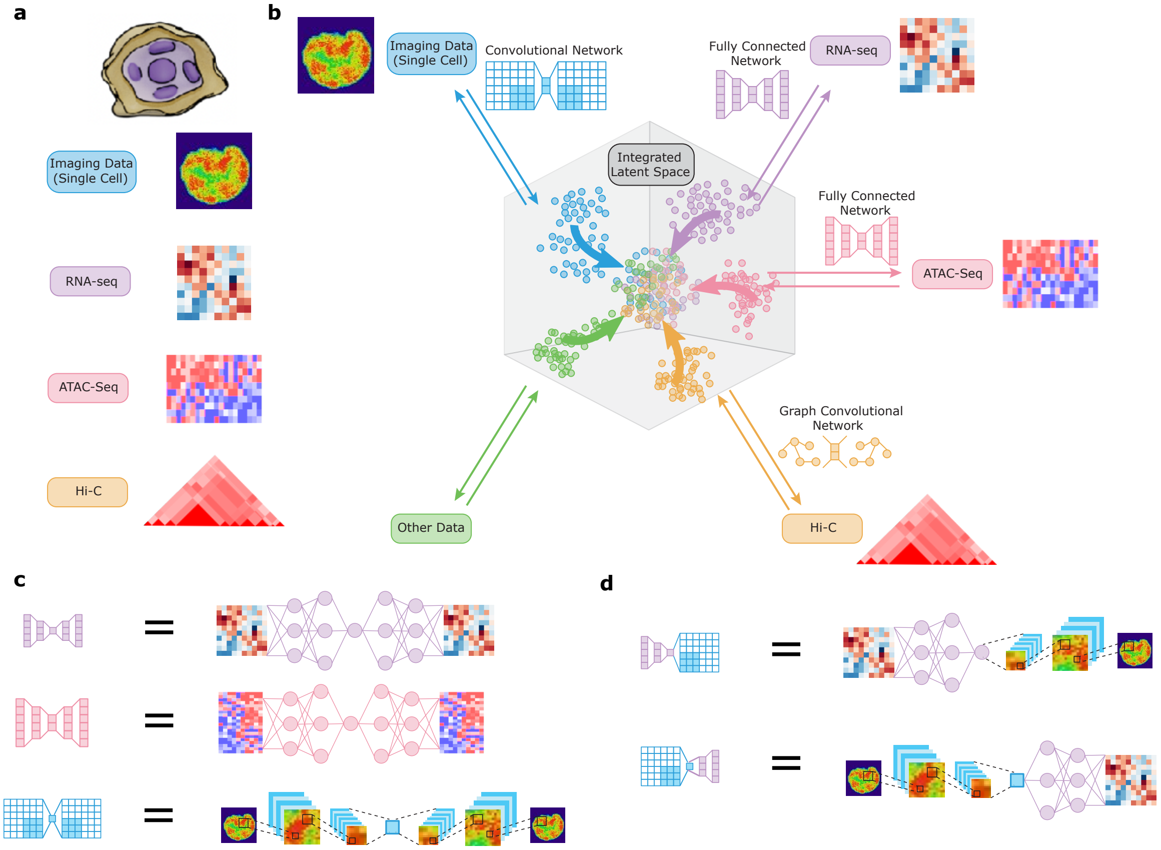


Figure 3-1: Schematic of multimodal data integration and translation strategy using our cross-modal autoencoder model. (a) Each modality or dataset (represented by different colors) presents a different view of the same underlying population of cells of interest. (b) Our computational strategy to integrate multiple modalities involves embedding each dataset into a shared space that represents the latent state of the cells, such that the distributions of each dataset mapped into the latent space are aligned. (c) The embedding of each dataset is performed using an autoencoder, a neural network with separate encoder and decoder modules, whose architectures can be customized to the specific data modality (autoencoders for each modality are represented by different colors). (d) Combining the encoder and decoder modules of different autoencoders enables translation between different data modalities at the single-cell level.

can then be formalized as the problem of learning conditional distributions $P_{X_i|Z}$ as well as the latent distribution P_Z based on samples from the marginal distributions $P_{X_1}, P_{X_2}, \dots P_{X_K}$, which are given by the datasets. Note that the assumption that each X_i is obtained via a deterministic function of Z implies that the latent distribution of each dataset is the same. However, by including the noise variables N_i as in Equation (3.2), our method extends to the case where only a subset of the latent dimensions is shared between the different modalities and the remaining dimensions are specific to each modality.

When the latent distribution P_Z is known, then learning the conditional distributions $P_{X_i|Z}$ given the marginals $P_{X_1}, P_{X_2}, \dots, P_{X_K}$ can be solved by learning multiple autoencoders. Specifically, for each domain $1 \leq i \leq K$, we propose training a regularized encoder-decoder pair (E_i, D_i) to minimize the loss

$$\mathbb{E}_{x \sim P_{X_i}} \left[L_1(x, D_i(E_i(x))) + \lambda L_2(E_i \# P_{X_i} | P_Z) \right], \tag{3.3}$$

where $\lambda > 0$ is a hyperparameter, L_1 is the (Euclidean) distance metric, L_2 represents a divergence between probability distributions, and $E_i \# P_{X_i}$ is the distribution of X_i after embedding to the latent space $\mathcal{Z}$. Translation from domain i to j is accomplished by composing the encoder from the source domain with the decoder from the target domain, i.e.,

$$X_{i \to j}(x_i) := D_j(E_i(x_i)). \tag{3.4}$$

The autoencoders obtained by minimizing the loss in Equation (3.3) satisfy various consistency properties; see (Yang and Uhler, 2019).

Since P_Z is not usually known in practice, it must also be estimated from the data. This can be done using the following approaches: (i) learn P_Z by training a regularized autoencoder on data from a single representative domain; or (ii) alternate between training multiple autoencoders until they agree on an invariant latent distribution. The first approach is typically more stable in practice, while the second captures variability across multiple domains and is therefore more suitable for integrating multiple datasets. Note that P_Z is by no means unique; there are multiple solutions that can result in the same observed data distributions in the different domains.

To be concrete, an invariant latent distribution based on two domains $i, j \in \{1, \ldots, K\}$ is learned as follows. Let $\hat{P}_{Z_{i'}}, i' \in \{i, j\}$ denote the empirical latent distribution based on the encoded data from domain i', i.e. $\hat{P}_{Z_{i'}} = E_{i'} \# P_{X_{i'}}$. Then for domain i, we optimize the objective

$$\min_{E_i, D_i} \mathbb{E}_{x \sim P_{X_i}} L_1(x, D_i \circ E_i(x)) + \lambda L_2(E_i \# P_{X_i} | P_{\hat{Z}_j}), \tag{3.5}$$

while for domain j, we optimize the objective

$$\min_{E_j, D_j} \mathbb{E}_{x \sim P_{X_j}} L_1(x, D_j \circ E_j(x)) + \lambda L_2(E_j \# P_{X_j} | P_{\hat{Z}_i}). \tag{3.6}$$

In practice, we parameterize (E_i, D_i) by neural networks and minimize the objective function via stochastic gradient updates. In particular, L_2 can be chosen to be the discriminative loss,

$$L_2(P|Q) := \max_f \ \mathbb{E}_{x \sim P} \log f(x) + \mathbb{E}_{x \sim Q} \log(1 - f(x)), \tag{3.7}$$

which is equivalent to the Jensen-Shannon divergence up to a constant factor. In practice, the model architecture of each autoencoder is selected based on the input data representation (e.g., fully-connected network for gene expression data and convolutional network for images). The dimensionality of the latent distribution is a hyperparameter that is tuned to ensure that the autoencoders are able to reconstruct the respective data modalities well. For sequencing data, PCA can be used to obtain an initial estimate of the intrinsic dimensionality of the data, which can then be fine-tuned by analyzing the reconstruction loss of the model. For imaging data the reconstruction quality can also be assessed qualitatively (see Supplementary Fig. B-5) and a variational autoencoder with a small weight on the KL divergence regularization term can be used to improve image generation quality.

Incorporating prior knowledge

Prior knowledge is sometimes available to guide the integration of different data modalities. For example, there may be knowledge of alignment of specific markers or clusters, or knowledge of certain samples from different datasets corresponding to the same cell, i.e., the same point in the latent space. In this case, training of the autoencoders can be guided by additional loss functions that incorporate the prior knowledge.

Discriminative loss to align shared markers/clusters among datasets: If there are shared markers or clusters that are present in two datasets, they can be aligned by replacing L_2 above with the following discriminative loss that is conditioned on these factors:

$$L_2(P|Q) := \max_f \mathbb{E}_{x,y\sim P} \log f(x,y) + \mathbb{E}_{x,y\sim Q} \log(1 - f(x,y)), \tag{3.8}$$

where P and Q are now joint distributions over the data and the markers and/or clusters. This approach is valid for both discrete and continuous values of the cluster/marker y. For example, in (Yang and Uhler, 2019), this approach was used to align a continuous differentiation marker between RNA-seq and ChIP-seq data. Alternatively, if the markers or clusters can take m discrete values (i.e., $1, \ldots, m$), then we can add a simple classifier model $p_\theta(Y|Z)$ with parameters θ and minimize the loss

$$\sum_{\text{modality } i} \mathbb{E}_{x,y\sim P_i} \sum_{j=1}^{m} 1(y = j)\, p_\theta(Y = j|Z = E_i(x)) \tag{3.9}$$

with respect to θ and the parameters of the encoders E_i; here P_i is the distribution of the ith data modality. This loss function encourages data points with the same class label irrespective of the data modality to be clustered together in the latent space.

Anchor loss to match paired samples: If (x_1, x'_1), $(x_2, x'_2), \ldots$, (x_m, x'_m) are corresponding points from two datasets that are embedded by encoders E and E', we can add the following anchor loss,

$$\sum_{i=1}^{m} \|E(x_i) - E'(x'_i)\| \tag{3.10}$$

to minimize their distance in the latent embedding space.

Model validation on paired single-cell RNA-seq and ATAC-seq data

Recent technological advances have made it possible to obtain paired single-cell RNA-seq and ATAC-seq data. Such paired data was collected from human lung adenocarcinoma–derived A549 cells treated with dexamethasone (DEX) for $0, 1$, or 3 hours in (Cao et al., 2018). While our autoencoder framework is designed to integrate vastly different data structures, in the following we show that our framework is competitive with previous methods for the simpler problem of integrating different modalities with similar data structures. For details on the implementation see Methods, Supplementary Table B.1 and Supplementary Table B.2. Since the RNA-seq and ATAC-seq data was collected in the same cell, we could evaluate the accuracy of our method in matching samples from RNA-seq to ATAC-seq (and vice-versa). We evaluated the accuracy of the matching by the following two measures: (a) the fraction of cells whose cluster assignment ($0, 1$, or 3 hours treatment with DEX) is predicted correctly based on the latent space embedding, and (b) k-nearest neighbors accuracy, i.e., the proportion of cells whose true match is within the k closest samples in the latent space (in ℓ_1-distance).

In Fig. 3-2, we compare our cross-modal autoencoder model to methods that align modalities in the latent space, namely deep canonical correlation analysis (DCCA) (Andrew et al., 2013), which determines a nonlinear transformation of the two datasets to maximize the correlation of the resulting representations, as well as to Seurat, a prominent method for biological data intergration of similar modalities (Butler et al., 2018; Stuart et al., 2019). In addition, we compare our cross-modal autoencoder model to two additional methods that do not rely on the latent space for alignment of modalities, namely CycleGAN (Zhu et al., 2017) and MAGAN (Amodio and Krishnaswamy, 2018). Similar to the CycleGAN, our cross-modal autoencoder does not require paired samples, which is advantageous for many modalities, where the process of data collection often results in destruction of the cell (e.g. RNA-seq) and thus the same cell cannot be used in another assay measuring a different modality (e.g. imaging). However, if additional information is available such as shared markers measured in all modalities and/or paired data, similar to the MAGAN approach, this prior information can be incorporated through addition of new terms in the loss function (see incorporating prior knowledge section). In terms of comparisons in the latent space, our autoencoder framework outperforms Seurat and is competitive with DCCA for integrating single-cell RNA-seq and single-cell ATAC-seq data both in terms of fraction of cells assigned to the correct cluster (Fig. 3-2a) as well as k-nearest neighbor accuracy (Fig. 3-2b). While paired data was only used to evaluate the accuracy in Fig. 3-2a-b, Fig. 3-2c-e explore the setting in which paired data on a fraction of samples is used for training. Although paired data is not necessary for our method, such prior knowledge can be incorporated using the anchor loss described above, which ensures that paired samples are close in the latent space. Fig. 3-2c-d show that our autoencoder model outperforms DCCA, CycleGAN and MAGAN when trained on varying amounts of paired data. In fact, as shown in Fig. 3-2e, our autoencoder model trained with just 25% of the paired samples has similar performance to DCCA trained on all (i.e. 100%) of the paired samples, thereby indicating that our method is practical and competitive also in the setting where some paired data is available.

Experimental validation on single-cell RNA-seq and chromatin images of naive CD4+ T-cells

We applied our method to integrate single-cell RNA-seq data with chromatin images in order to study the heterogeneity within naive T-cells. T-cell activation is a fundamental biological process and identifying naive T-cells poised for activation is critical to understanding immune response (Smith-Garvin et al., 2009). Moreover, linking genome organization with gene expression generates hypotheses that can be tested experimentally to validate our methodology.

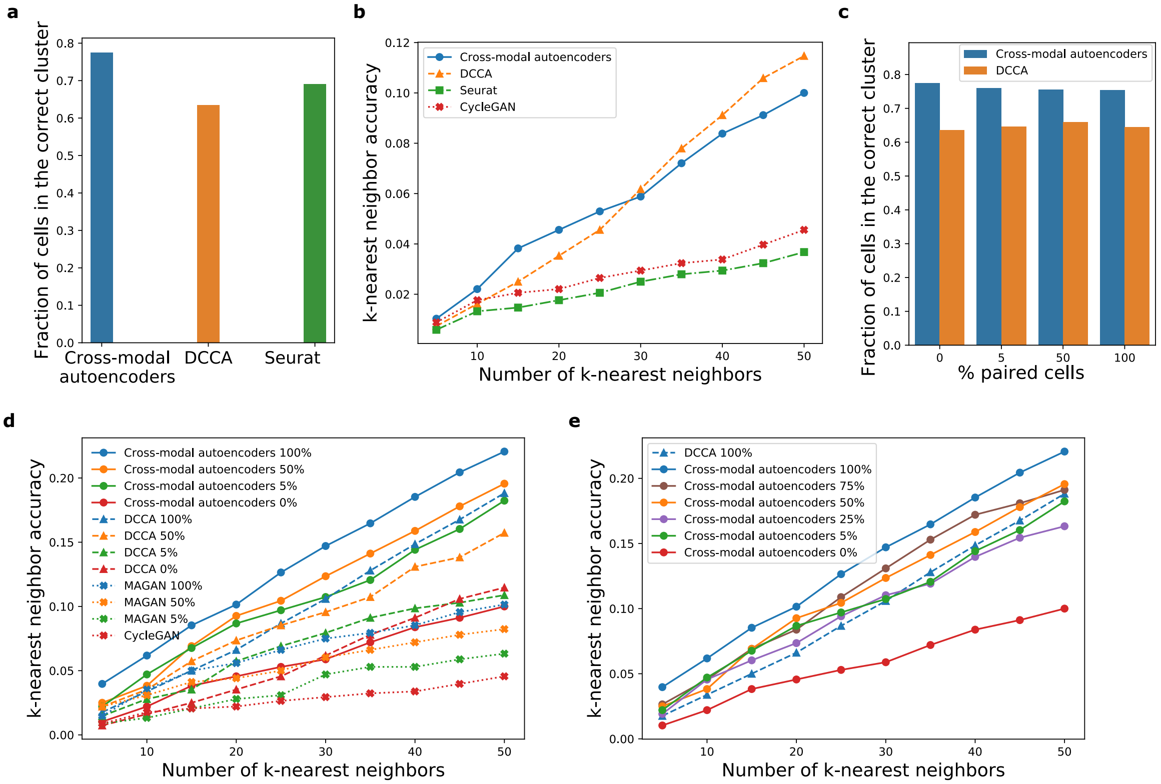


Figure 3-2: Performance of our multimodal data integration method (cross-modal autoencoders), deep canonical correlation analysis (DCCA), Seurat, CycleGAN and MAGAN on paired RNA-seq and ATAC-seq data. (a) Fraction of cells that were assigned to the correct treatment time cluster based on their embedding in the integrated latent space that was learned by fitting our cross-modal autoencoder model, DCCA, or Seurat. (b) k-nearest neighbor accuracy for quantifying the quality of matching between local neighborhoods for our cross-modal autoencoder model, DCCA, Seurat and CycleGAN trained with 0% supervision (no paired samples). (c) Fraction of cells that were assigned to the correct treatment time cluster for our cross-modal autoencoders and DCCA trained with varying amount of paired samples. (d) k-nearest neighbor accuracy for our cross-modal autoencoders, DCCA, MAGAN and CycleGAN trained with $0, 5, 50$ and 100% of the paired samples. (e) k-nearest neighbor accuracy for our cross-modal autoencoder model trained with varying amount of paired samples versus DCCA trained on all paired samples. In (a-c) colors denote different domain translation methods and in (d-e) colors denote different levels of supervision (paired samples). Additionally, different markers denote different domain translation methods.

Single-cell RNA-seq analysis of naive CD4+ T-cells revealed two distinct subpopulations

We analyzed single-cell RNA-seq data of human peripheral blood mononuclear cells (PBMCs) from (Zheng et al., 2017); for details on the analysis see Methods. We used known markers to identify naive and activated (CD4+) T-cells (Fig. 3-3a, Supplementary Fig. B-1 and Supplementary Table B.3. An in-depth analysis of the naive T-cell population revealed two distinct subpopulations (see Methods). The number of subpopulations / clusters was obtained via two separate analyses, namely by maximizing the silhouette coefficient (Supplementary Fig. B-2a) and by minimizing the Bayesian information criterion (Supplementary Fig. B-2b). The co-association matrix shown in Fig. 3-3b, which quantifies how often each pair of cells was clustered together for different clustering methods, shows that the two clusters were highly robust to the choice of clustering method. Differential gene expression and gene ontology (GO) enrichment analysis indicated that one cluster corresponded to quiescent cells while the other was poised for activation, with an expression profile more similar to that of activated T-cells (Fig. 3-3c-d). Specifically, we observed that one of the two clusters of naive CD4+ T-cells contained "immune response" and "cell activation" as one of the top significant GO terms as well as a well-known activation marker IL32 as one of the differentially expressed (DE) genes.

Analysis of single-cell chromatin images of naive CD4+ T-cells revealed two distinct subpopulations

Given the link between expression and chromatin organization (Uhler and Shivashankar, 2017a), we hypothesised the presence of two subpopulations of naive T-cells with distinct chromatin packing features. To test this, we carried out DAPI-stained imaging experiments of naive CD4+ human T-cells and analyzed their chromatin organization (Methods, Fig. 3-3e, and Supplementary Fig. B-3). We extracted image features by quantifying the chromatin density in concentric spheres with increasing radii (Methods, Fig. 3-3f). Cluster analysis based on the extracted features revealed

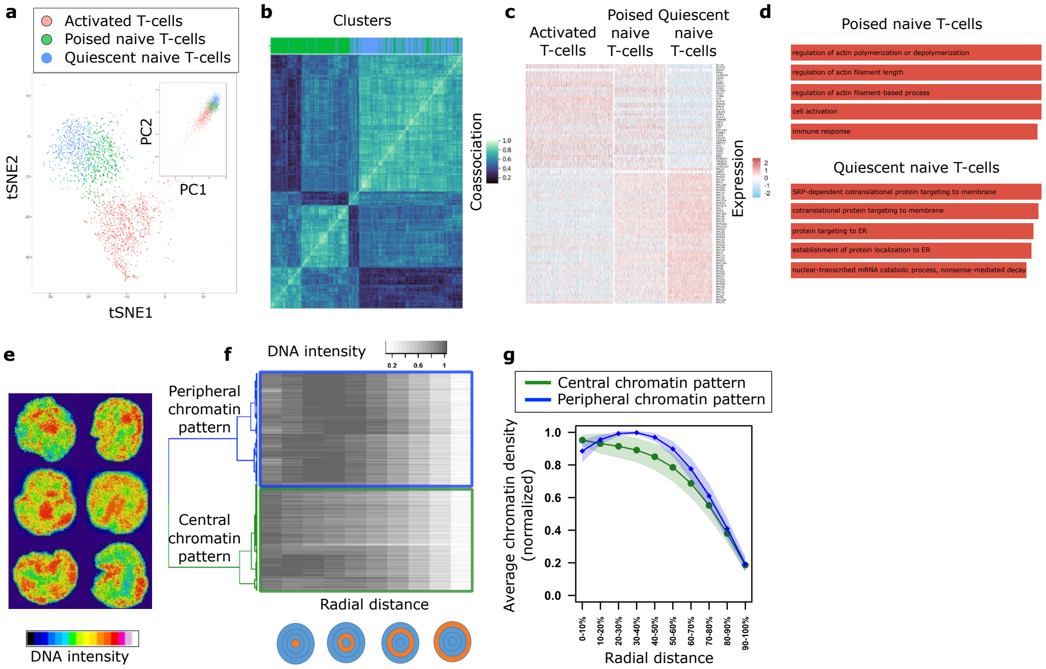


Figure 3-3: Analysis of single-cell RNA-seq data and single-cell chromatin images of naive CD4+ T-cells reveals two distinct subpopulations respectively. (a) t-SNE and PCA (inset) embeddings of single-cell RNA-seq data derived from (Zheng et al., 2017). Cluster analysis reveals activated (red) population of T-cells and naive population of T-cells divided into two subpopulations (poised and quiescent, denoted in green and blue, respectively). (b) Consensus clustering plot demonstrating the robustness of quiescent (blue) and poised (green) clusters of naive T-cells to various clustering methods. Gene expression data was clustered using k-means, Gaussian mixture models and spectral clustering based on a k-nearest neighbor graph with $k \in \{10, 20, 50, 100\}$ with 100 initializations for each method. (c) Differential gene expression analysis between the blue and green subpopulations reveals two distinct gene expression programs. The green subpopulation of naive T-cells is more similar to the activated T-cells and hence poised for activation, while the blue subpopulation shows an upregulation of ribosomal genes and has a relatively more quiescent expression profile. (d) Gene ontology enrichment analysis of marker genes for quiescent and poised naive T-cell subpopulations supports two distinct gene expression programs. (e) Examples of DAPI-stained nuclear images of naive CD4+ T-cells. (f) Cluster analysis of the 3D nuclear images is performed by first quantifying the chromatin signal in concentric spheres with increasing radii, and then using hierarchical clustering on these spatial chromatin features. The features were clustered using hierarchical clustering with complete linkage based on the distance matrix obtained from 1-Spearman's correlation. (g) Average chromatin signal , calculated using $n = 729$ cells from two biologically independent replicates, (mean represented by the solid line and standard deviation represented by shading) in concentric spheres with increasing radii for central (green) and peripheral (blue) clusters. One cluster has higher concentration of chromatin in the central region of the nucleus (green), while the other cluster has higher concentration of chromatin in the peripheral region of the nucleus (blue).

two distinct subpopulations of cells, with higher chromatin density in the central and peripheral nuclear regions respectively (Fig. 3-3g, Supplementary Fig. B-4). These observations are consistent with previous experiments in mouse naive T-cells that also showed two subpopulations with distinct chromatin organization patterns, where naive T-cells with more central heterochromatin were shown to be poised for activation (Gupta et al., 2012).

Autoencoder framework allows integrating and translating between single-cell expression and imaging data

Up to this point, we had observed two subpopulations of naive T-cells based on a separate analysis of gene expression (from single-cell RNA-seq data) and chromatin packing (from single-cell imaging data).

To link the identified subpopluations from the unpaired datasets, we used our cross-modal autoencoder framework to integrate the single-cell RNA-seq data with the chromatin images (Methods and Supplementary Table B.4), thereby enabling translation between the two data modalities at the single-cell level (Fig. 3-4a and Supplementary Fig. B-5). Visual inspection of the latent representations indicates that the subpopulations from the two datasets are appropriately matched (Fig. 3-4b and Supplementary Fig. B-7). To quantitatively assess whether our methodology aligns imaging features and gene expression features in a consistent manner, we next analyzed the latent embeddings as well as the results of translation between the two datasets. Consistent with other methods used for data integration and translation in the biological domain, where the goal is to provide a matching between samples in the observed datasets (Stuart et al., 2019), our evaluation is based on the full dataset used for training rather than a held-out evaluation set.

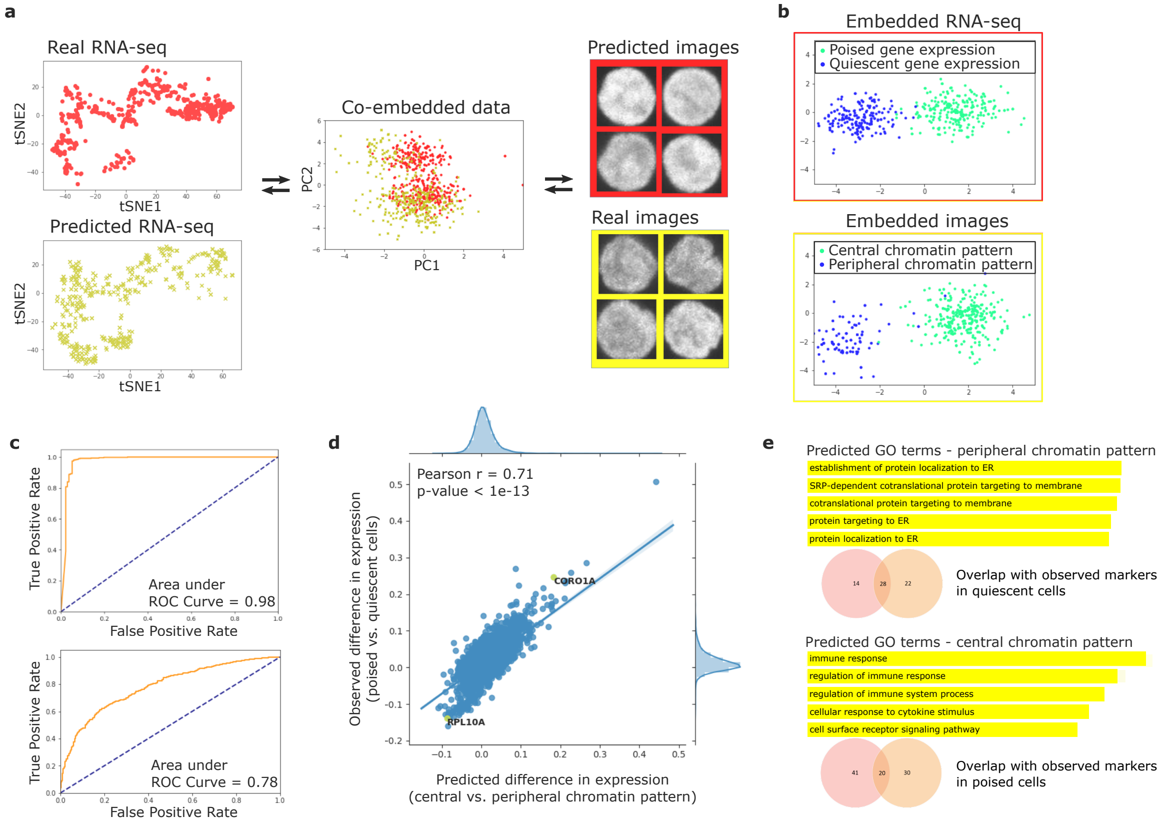

a
Real RNA-seq
tSNE2
tSNE1
Predicted RNA-seq
tSNE2
tSNE1
Co-embedded data
PC2
PC1
Predicted images
Real images
b
Embedded RNA-seq
Poised gene expression
Quiescent gene expression
Embedded images
Central chromatin pattern
Peripheral chromatin pattern
c
True Positive Rate
False Positive Rate
Area under
ROC Curve = 0.98
True Positive Rate
False Positive Rate
Area under
ROC Curve = 0.78
d
Pearson r = 0.71
p-value < 1e-13
CORO1A
RPL10A
Observed difference in expression
(poised vs. quiescent cells)
Predicted difference in expression
(central vs. peripheral chromatin pattern)
e
Predicted GO terms - peripheral chromatin pattern
establishment of protein localization to ER
SRP-dependent cotranslational protein targeting to membrane
cotranslational protein targeting to membrane
protein targeting to ER
protein localization to ER
14
28
22
Overlap with observed markers
in quiescent cells
Predicted GO terms - central chromatin pattern
immune response
regulation of immune response
regulation of immune system process
cellular response to cytokine stimulus
cell surface receptor signaling pathway
41
20
30
Overlap with observed markers
in poised cells

Figure 3-4: Integration of single-cell RNA-seq data and single-cell nuclear images of naive T-cells using our methodology allows translating between chromatin packing and gene expression profiles. (a) Illustration of data integration and translation: (left) t-SNE plots of observed single-cell RNA-seq data (red) and single-cell RNA-seq data translated from single-cell images (yellow); (middle) PCA visualization of single-cell RNA-seq data (red) and single-cell imaging data (yellow) embedded in 128-dimensional latent space; (right) examples of observed single-cell images (yellow) and images translated from single-cell RNA-seq data (red). (b-e) Evidence that our data integration methodology correctly aligns gene expression features and imaging features. (b) Linear Discriminant Analysis (LDA) plots of single-cell RNA-seq (top) and imaging (bottom) datasets embedded in the latent space. The clusters with more quiescent (blue) and poised (green) gene expression programs from the RNA-seq dataset are aligned with the clusters with peripheral (blue) and central (green) chromatin patterns from the imaging dataset. (c) (top) Receiver Operating Characteristic (ROC) curve illustrating performance of a classifier trained to distinguish between peripheral and central chromatin patterns in images when evaluated on images translated from RNA-seq data. (bottom) ROC curve illustrating performance of a classifier trained to distinguish between quiescent and poised gene expression programs when evaluated on RNA-seq data translated from images. High performance of both classifiers indicates that the alignment of the clusters in the latent space in (b) also holds in the original gene expression and imaging spaces. The dotted line represents random guessing based on evenly-distributed classes. (d) Differential gene expression analysis between cells with central and peripheral chromatin pattern performed on the predicted gene expression matrix translated from images using our methodology. The predicted fold-change of gene expression based on images is strongly correlated with the observed fold-change of gene expression between quiescent and poised naive T-cells from the actual RNA-seq dataset. (e) Analysis of gene ontology (GO) enrichment terms of cells with central and peripheral chromatin pattern based on the predicted gene expression matrix translated from images using our methodology shows a high overlap between predicted markers (orange) from the imaging dataset and actual markers (red) from the RNA-seq dataset.

ROC analysis on translated datasets indicates that imaging and gene expression features are consistently aligned

In order to assess whether translated image (or RNA-seq respectively) datasets are still able to separate poised and quiescent subpopulations (or central and peripheral subpopulations respectively) and analyze if the clusters obtained separately from gene expression and imaging datasets align with each other, we performed Receiver

Operating Characteristic (ROC) analysis on the translated datasets. For RNA-seq, we first trained a random forest classifier (using 100 trees in a forest with 2 as the maximum depth of a tree) on the RNA-seq data with labels based on poised versus quiescent clustering of naive CD4+ T-cell gene expression data. This classifier learned the genes that separate the two clusters. Next, we translated chromatin images into RNA-seq using our autoencoder method and assessed the performance of the pre-trained classifier on its ability to separate central versus peripheral clusters on images translated to RNA-seq (Fig. 3-4c, top). Similarly, to assess translation of RNA-seq into images, we trained a classifier to separate central versus peripheral chromatin patterns. Then, we translated RNA-seq data into images and evaluated the performance of the pre-trained classifier in being able to separate poised versus quiescent clusters (Fig. 3-4c, bottom). The area under the curve (AUC) was computed for both of these tasks. The high AUCs demonstrate that classifiers trained to distinguish between the subpopulations in the original datasets also performed well when evaluated on the translated datasets.

Strong correlation of DE genes between original RNA-seq and images translated to RNA-seq indicates consistent alignment

Imaging datasets can provide a rich quantification of cells, such as their chromatin organization. Based on image analysis, subpopulations of cells with different characteristics may be found (e.g., central versus peripheral chromatin organization), and it is often of interest to study which genes might be markers of each subpopulation such that these subpopulations can be separated for example using antibodies against the marker genes. However, generally the full gene expression and imaging features cannot be measured in the same cell. Our computational framework can translate chromatin images into RNA-seq and calculate the predicted mean difference in expression between the subpopulations (e.g. for central versus peripheral chromatin organization). As shown in Fig. 3-4d, the observed mean difference in expression is strongly correlated with the predicted mean expression difference. In addition, we obtained a set of marker genes associated with central and peripheral chromatin organization by performing two-sided Welch's t-test on the generated RNA-seq data (considering marker genes for each cluster to be the top 50 genes that had the highest mean difference in expression for the two clusters as well as p-value < 0.05 after adjustment for multiple hypothesis testing using the Benjamini–Hochberg procedure). Note the considerable overlap between the true and predicted marker genes (Fig. 3-4e). We also performed GO analysis on the marker genes for each cluster; we report the top 5 GO biological process terms with lowest p-values (FDR adjusted p-value < 0.05). In summary, in the gene expression matrix translated from the imaging dataset, we found that the differential expression of genes was strongly correlated with the true observed differential gene expression and that the predicted and observed marker genes showed considerable overlap.

Experimental validation of matching via protein immunofluorescence staining

Our model generates predictions of gene expression programs based on patterns of chromatin density (Fig. 3-4e). To validate these results experimentally, we chose two genes, *CORO1A* and *RPL10A*, which were predicted to be strongly upregulated in the naive T-cell subpopulations with central and peripheral patterns of chromatin density respectively (Fig. 3-4d, Fig. 3-5a). We analyzed the immunofluorescence staining data of these proteins obtained along with chromatin images (Fig. 3-5b). Consistent with the model predictions, we found that CORO1A was upregulated in the cells with central chromatin pattern, while RPL10A was upregulated in the images with peripheral chromatin pattern (Fig. 3-5c and Supplementary Methods and Supplementary Fig. B-8). These results altogether demonstrate that our method properly aligns the gene expression and image features that characterize two distinct subpopulations of human naive T-cells, and suggests that peripheral and central enrichment of chromatin are associated with gene expression programs for more quiescent and poised naive CD4+ T-cells respectively (Fig. 3-5d).

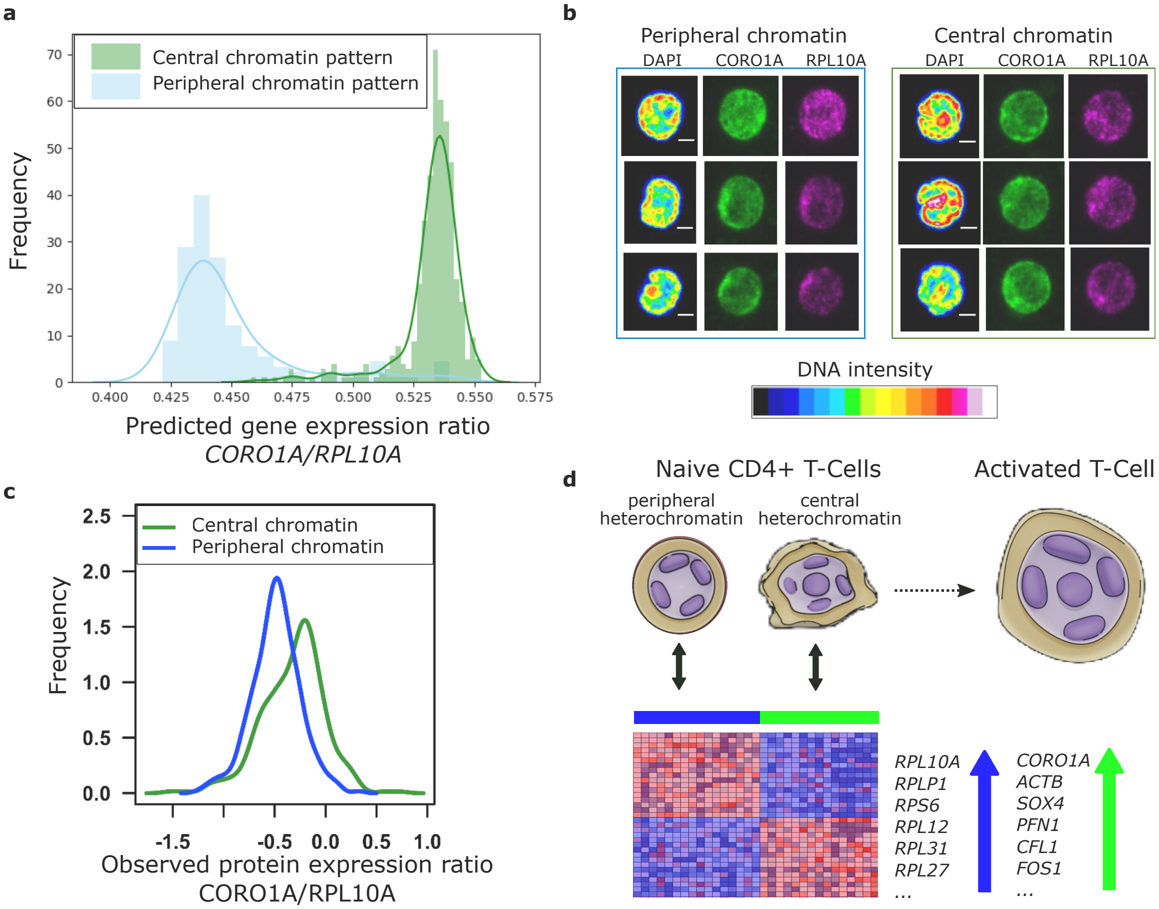


Figure 3-5: Validation of our model alignment using single-cell immunofluorescence experiments. (a) Histograms of predicted *CORO1A*/*RPL10A* gene expression ratio in cells with central (green) and peripheral (blue) chromatin pattern based on the gene expression matrix translated from the imaging dataset. Our model predicts the upregulation of *CORO1A* and *RPL10A* in the cells with central and peripheral chromatin pattern respectively. (b) Examples of immunofluorescence staining data of CORO1A and RPL10A proteins collected along with the chromatin images. (c) Histograms of measured CORO1A/RPL10A protein ratio in cells with central (green) and peripheral (blue) chromatin pattern. Consistent with the model prediction, CORO1A and RPL10A proteins are upregulated in the cells with central and peripheral chromatin pattern respectively (p-value $< 2.2 \times 10^{-16}$, two-sided Welch's t-test). (d) Schematic of the two naive T-cell subpopulations characterized by our multimodal analysis, in which peripheral and central patterns of chromatin density are associated with gene expression programs for quiescent (blue) and poised (green) naive CD4+ T-cells respectively. The up and down arrows represent which genes are upregulated and downregulated respectively as predicted by our model.

3.4 Discussion

In summary, we presented a powerful approach to integrate and translate between different data modalities of very different structures, namely single-cell chromatin images and RNA-seq. Using our cross-modal autoencoder methodology, we established a quantitative link between chromatin organization and expression, jointly characterizing a subpopulation of naive T-cells that is poised for activation using both data modalities. Additionally, we validated our model's predictions of gene expression using protein fluorescence experiments.

While we used our method to align RNA-seq and imaging datasets, we have presented a general framework that can be adapted to numerous other biological problems. As indicated in Figure 3-1, our framework can be used to integrate datasets of different modalities simply by incorporating autoencoder architectures tailored to those modalities. For example, Hi-C data could be integrated using a graph neural network and multi-channel cell images using a convolutional neural network with different input channels. Also, while we focused on aligning datasets each containing two distinct clusters, our method can be applied to datasets with other distributions as long as the samples are taken from the same cell population. For example, in applications where there are no clear clusters in the datasets, our method can be used to align continuous markers between datasets by conditioning the adversarial loss on the values of the continuous marker (Equation 3.8). In applications where there is some shared signal between modalities as well as signal that is individual to each modality, our model can be extended by introducing a subset of latent dimensions that is specific to each modality. Empirically validating this aspect of our model is a potential direction for future work. An important consideration, however, is that while our method can be applied for data integration and cross-modal alignment in generic contexts, depending on the data distributions, there may be multiple alignments that satisfy the same objective function. Additional constraints (in the form of prior knowledge) should be added to these models where possible to enforce alignments that are biologically accurate. Overall, we envision an iterative process of biological discovery where our predictive model is used for hypothesis generation (for example linking particular image features to particular gene regulatory modules), the hypotheses are validated (or disproved) experimentally, and the new experimental results now serve as additional data (prior knowledge) for improving the alignment of the model. In summary, our methodology can be applied generally to integrate single-cell datasets that cannot yet be measured in the same cell by using a different autoencoder for each data modality, and as such has broad implications for the integration of spatial transcriptomics (Ståhl et al., 2016), proteomics (Irish et al., 2006) and metabolomics (Zenobi, 2013) datasets. In particular, our methodology can be applied to generate hypotheses and predict the functional landscape of single cells in a tissue section where only limited functional data is available by acquiring chromatin imaging data.

3.5 Future directions

We proposed a methodology that enables integration and translation between different data modalities by learning a shared latent space. We explored the performance of our method in two settings: for the integration of gene expression and DNA accessibility data as well as gene expression and imaging data. While our approach is powerful, the task of integrating different modalities is very challenging and various improvements in terms of accuracy should be explored. How challenging the task is given the current data quality is highlighted by the observation that DCCA with 100% supervision (having fully paired data) achieves low accuracy. Application areas such as natural language processing have seen great successes in accurate domain translation. Quite surprisingly, for natural language processing, learning separate embeddings for each data modality and aligning them post hoc using for example Procrustes algorithm works quite well. For the biological applications mentioned in this work where integration and translation between very different data modalities is required, separate estimation of data embeddings is not likely to perform well. Therefore, an important focus of future work are improvements on how to learn a better shared latent space. Since the state of a cell is governed by the underlying regulatory mechanisms, one approach could be to learn a latent space that identifies these underlying causal units. This is particularly important for imaging data, where it is not immediately clear how information from the pixels can be summarized into causal factors. Since common biological phenomena should be consistently observed across different modalities, considering multiple data modalities will be critical for the identification of such causal units. On the other hand, different data modalities may not share exactly the same information. Therefore, it would also be of interest to learn a latent space, where a subset of the latent dimensions are specific to each modality. Related questions are the identification of the modality that contains the most information about the biological phenomenon, learning features that are identifiable only in a certain modality and discovering which features are most different across modalities.

In terms of biological questions, it would be interesting to apply our autoencoder framework to novel datasets that are beginning to be collected at scale. For example, recent technological advances in spatial transcriptomics allow simultaneous imaging of cells and the collection of gene expression data on a few hundred to thousands of genes. Our approach could be used to impute the expression profiles of the remaining genes based on the cell images. Finally, exploring the integration of more than two data modalities would also be an interesting avenue for future work.

3.6 Methods

Model validation on paired RNA-seq and ATAC-seq data

We obtained paired RNA-seq and ATAC-seq data collected in the same cell from (Cao et al., 2018). Specifically, we used paired data collected from human lung adenocarcinoma–derived A549 cells treated with dexamethasone (DEX) for $0, 1$, or 3 hours. We downloaded the single-cell RNA-seq data from the GEO accession number GSE117089, corresponding to (Cao et al., 2018). For the ATAC-seq data, instead of using raw matrix of peaks by cells, we acquired a transcription factor (TF) motif by cells matrix from the authors, which was computed as described in (Cao et al., 2018) by counting occurrences of each motif in all accessible sites for each cell, resulting in 815 TF motifs. For single-cell RNA-seq data we considered genes that were determined to be differentially expressed by (Cao et al., 2018), keeping genes with q-value > 0.05. Both single-cell RNA-seq and ATAC-seq were $\log(x+1)$ transformed and normalized to zero mean and unit variance. The number of cells that were shared between TFs × cells matrix from ATAC-seq and genes × cells matrix from RNA-seq was 1874, therefore our model had to learn a latent embedding and translate between data sets with different number of features, i.e. for single-cell RNA-seq a matrix of 2613 genes × 1874 cells and for single-cell ATAC-seq a matrix of 815 TFs × 1874 cells.

We trained our cross-modal autoencoder model to embed the single-cell RNA-seq and ATAC-seq into the same latent space (dimensionality of 50), which allows mapping and translation of samples from one space to the other. Our model's architecture consisted of fully connected layers with input, hidden layers and output sizes listed in Supplementary Table B.1. In order to train the model we minimized the weighted sum of losses listed in Supplementary Table B.2. The model was trained in Pytorch with learning rate of 0.0001 and batch size of 32 for 4000 epochs using Adam with $\beta_1 = 0.5, \beta_2 = 0.999$ and weight decay of 0.0001.

Since the RNA-seq and ATAC-seq data was collected in the same cell, we could evaluate the accuracy of our method in matching samples from RNA-seq to ATAC-seq (and vice-versa). For evaluation, we created an 80-20 training-test split of the paired data. To measure the accuracy of matching RNA-seq and ATAC-seq samples in the latent space or in the original space for methods that do not rely on the latent space, we used the following k-nearest neighbors accuracy, calculated on the test set:

$$\text{k-NN}(\mathrm{A}, \mathrm{B}) = \frac{\sum_i 1(b_i' \in a_i^k)}{n}, \tag{3.11}$$

where n is the length of the test set, A and B are sets of vectors, with b_i as a vector in B and a_i as its pair in A, and b_i' and a_i' are the encoded versions of b_i and a_i in the latent space. The set a_i^k contains the k nearest neighbors of a_i' in A', the set of vectors in A projected into the latent space. Since k-NN(A, B) does not necessarily equal k-NN(B, A), we computed the average of these metrics. We used ℓ_1 distance for distance computations.

In order to quantify whether our model maps cells to the correct treatment time cluster, we computed the fraction of cells in the test split that had the correct cluster assignment. In order to assign cells to a cluster, we trained a simple logistic regression classifier on the latent space using cells in the training split and their corresponding treatment time labels. Subsequently, the trained classifier was used to predict treatment time labels on the cells in the test set and the accuracy of the classifier was quantified.

First, we compared our method against deep canonical correlation analysis (DCCA), which uses paired samples between two domains to learn a shared embedding of the two domains by maximizing the total correlation (Andrew et al., 2013). The model for DCCA consisted of two neural networks, one for each domain. For ATAC-seq data, the input to the model was a matrix with 815 features, followed by 815 hidden nodes with sigmoid activation, and a final output layer of size 50. For RNA-seq data, the input to the model was a matrix with 2613 features, followed by 2613 hidden nodes with sigmoid activation, and a final output layer of size 50. Finally, as in (Andrew et al., 2013), linear CCA was applied to the output layers of the two neural networks corresponding to the two different domains. DCCA jointly learns the parameters for both neural networks such that the correlation of the final output layer between the domains is maximized. DCCA was trained using RMSProp with learning rate of 10^{-3}, batch size of 1024 for 100 epochs. Regularization parameter of 10^{-9} was applied to the networks.

For both our cross-modal autoencoder method and DCCA, we explored the use of samples whose pairing is known between the two domains (i.e., anchored cells in both datasets), which is available in some applications. To make use of the pairing information in our cross-modal autoencoder model, we included an additional term in the loss function corresponding to the mean absolute error between the paired training points in the latent space. While our method based on autoencoders does not require paired samples, DCCA does. In order to train DCCA with 0% paired samples, we randomly generated paired samples using the treatment time labels of the cells as follows. For each point with a particular treatment time label, we sampled 100 random points with the same label to use as its paired samples.

We additionally compared our method against a popular method for data integration, Seurat version 3.0 (Butler et al., 2018; Stuart et al., 2019). Briefly, this method assumes that the features across different modalities are the same and learns a shared embedding using CCA based on this assumption. In order to apply Seurat to this particular dataset, we used the Seurat pipeline as follows: in order to obtain from ATAC-seq data a matrix that has the same features as the gene expression matrix, the ATAC-seq data was transformed into a gene activity matrix using the CreateGeneActivityMatrix function in Seurat 3.0. We normalized and scaled the data using the NormalizeData and ScaleData functions in Seurat 3.0. Finally, a shared CCA embedding was learned using the FindTransferAnchors functionality in Seurat 3.0. Similar to our cross-modal autoencoder and DCCA, we used the inferred CCA embedding to quantify the method's performance. Note that Seurat was fit using both training and test data, thereby giving Seurat an advantage over the other methods.

Finally, we compared our method against CycleGAN (Zhu et al., 2017), a prominent deep learning method for domain translation, which ensures that source samples are recovered back after mapping source samples to target domain and back to the source domain. We used the code provided by the authors of CycleGAN at `http://github.com/junyanz/pytorch-CycleGAN-and-pix2pix` to translate ATAC-seq to RNA-seq and RNA-seq to ATAC-seq. We modified the architecture of the generator and discriminator networks to handle non-image data and match the architecture of our cross-modal autoencoder. In particular, the generator for translating ATAC-seq to RNA-seq consisted of a sequence of fully-connected layers with the following sizes: 815, 815, 815, 100, 50, 100, 2613, 2613, 2613. Similarly, the generator for translating RNA-seq to ATAC-seq consisted of a sequence of fully-connected layers with the following sizes: 2613, 2613, 2613 815, 815, 815, 100, 50, 100, 815, 815, 815.

The discriminator model for ATAC-seq data took as input 815 features, followed by 815 hidden nodes and then 100 hidden nodes with a final output layer of size 1. The discriminator model for RNA-seq data took as input 2613 features, followed by 2613 hidden nodes and then 100 hidden nodes with a final output layer of size 1. All models used leaky ReLU as activation. The CycleGAN was trained for 2000 epochs with a learning rate of 0.0002 and batch size of 32. We evaluated the model only in terms of the k-nearest neighbor accuracy since the fraction of cells in the correct cluster was meant to evaluate the quality of the latent space. The k-nearest neighbor accuracy of the CycleGAN was computed in the original instead of the latent space since the model does not rely on the latent space for domain translation. Similarly, we compared our method against MAGAN (Amodio and Krishnaswamy, 2018), which has an additional correspondence loss term that ensures the measurements coming from the same sample should be close to each other. We trained MAGAN by providing 5%, 50% and 100% of paired samples in the training data for the correspondence loss.

Gene expression data of naive CD4+ T-cells

We used gene expression data corresponding to human peripheral blood mononuclear cells (PBMCs) collected in (Zheng et al., 2017); the filtered cell by gene matrix was downloaded from `https://support.10xgenomics.com/single-cell-gene-expression/datasets/2.1.0/pbmc8k`. We analyzed the PBMC 8k data set since it had the highest number of reads per cell. Since the data was already filtered, we only applied minor additional filtering such as removing cells with high proportion of counts in mitochondrial genes ($\geq 10\%$), which reduced the number of cells from 8381 to 8371 cells. After cell filtering, we performed gene filtering by removing mitochondrial genes and keeping genes which had at least 10 cells expressing the gene with a count > 1, resulting in 7633 remaining genes.

After cell and gene filtering, we followed a standard analysis pipeline using Seurat (version 2.3.0) (Butler et al., 2018; Stuart et al., 2019). We normalized the gene expression measurements for each cell by the total expression for that cell and scaled the result using the median total expression across cells. The scaled result was $\log(x+$

1) transformed. We z-scored the data and applied PCA to obtain 30 components, which were used for t-SNE and clustering analysis. The t-SNE embedding for all cells, computed using default parameters, is shown in Supplementary Fig. B-1a. We clustered the data using default clustering in Seurat (FindClusters function in Seurat version 2.3.0) with resolution parameter of 0.4, which resulted in 13 clusters, shown in Supplementary Fig. B-1a. Briefly, the clustering method in Seurat constructs a k-nearest neighbor graph and adjusts the edge weights between cells based on Jaccard similarity of their local neighborhoods. The resulting graph is clustered using the Louvain algorithm to obtain cell clusters. In order to determine the identity of each cluster we performed differential expression analysis using the default Wilcoxon rank sum test in Seurat (FindAllMarkers function in Seurat version 2.3.0). We list the top 10 differentially expressed genes for each cluster in Supplementary Table B.3.

From the clustering analysis of all PBMCs and annotation using differentially expressed genes, we were able to determine that cluster 1 generally corresponds to naive CD4+ T-cells (differential overexpression of CCR7, LEF1, TCF7), cluster 2 corresponds to cytotoxic T-cells (differential overexpression of GZMK, NKG7, CCL5), cluster 3 corresponds to activated CD4+ T-cells (differential overexpression of IL7R, IL32) and cluster 4 corresponds to naive CD8+ T-cells (differential overexpression of CD8A, CD8B, LEF1, CCR7) (Willinger et al., 2006; Ding et al., 2019). Supplementary Fig. B-1b provides a t-SNE plot of all PBMCs, colored by the expression of known markers genes, further corroborating our cell type annotation.

Gene expression analysis of naive CD4+ T-cells

We aimed to explore potential heterogeneity in naive CD4+ T-cell gene expression in relation to CD4+ T-cells that already underwent activation. We performed a feature selection step, keeping genes which had average log-fold change of > 0.05 between naive and activated CD4+ T-cells (and vice-versa), resulting in 1187 genes. Similar to the analysis of PBMCs (see the previous section), we applied PCA for dimensionality reduction on the selected genes, keeping the top 30 components and clustered the naive CD4+ T-cells using the default clustering method in Seurat version 2.3.0

with resolution of 0.8 (Supplementary Fig. B-1c). Based on differential expression analysis and t-SNE embedding, the smallest cluster (shown in grey in Supplementary Fig. B-1c) was determined to belong to the CD8+ T-cell population since the top differentially overexpressed genes for this small cluster were CD8A and CD8B. Therefore, this small cluster was removed from the downstream gene expression analysis of the naive CD4+ T-cells. In order to characterize the remaining two subpopulations, we performed differential expression analysis on the two subpopulations of naive CD4+ T-cells using Wilcoxon rank sum test. We defined marker genes as all genes with Bonferroni-corrected p-value of < 0.05. Fig. 3-3c, shows the resulting heatmap for the genes that are markers between poised and quiescent subpopulations of naive T-cells and are also part of the 1187 genes that have an average log-fold change of > 0.05 between naive and activated CD4+ T-cells (and vice-versa). Gene ontology analysis was performed on these marker genes overexpressed in each cluster (average log-fold change > 0) using g:Profiler (Reimand et al., 2007, 2016), keeping the top 5 gene ontology biological process terms with lowest p-values (Fig. 3-3d). All reported p-values (after adjusting for multiple hypothesis testing using the Benjamini–Hochberg procedure) were ≤ 0.05.

Since the identification of the two subpopulations of naive T-cells is an important step in our analysis, we thoroughly evaluated its robustness with respect to number of clusters and clustering methods. We re-clustered the data corresponding to naive CD4+ T-cells using Seurat version 2.3.0 with different resolution parameters, i.e. $0.9, 1.1$ and 1.15 to obtain $3, 4$ and 5 clusters respectively. We computed the silhouette coefficient for each clustering, observing that the number of clusters corresponding to 2 gave the highest score (Supplementary Fig. B-2a). This suggests that using 2 clusters is optimal. We also fit a Gaussian mixture model to the data and computed the Bayesian information criterion (BIC) for a model with $1, 2, 3, 4$ and 5 mixture components (across 100 randomly initialized trials). Also with this method the model with 2 components resulted in the lowest mean BIC, suggesting again that 2 clusters is optimal for this data (Supplementary Fig. B-2b). To test the robustness with respect to different clustering methodologies, we also used k-means, Gaussian mixture models and spectral clustering based on a k-nearest neighbor graph with $k \in \{10, 20, 50, 100\}$ to cluster the data. We performed 100 different initializations for each method and computed the co-association matrix, which quantifies how often each pair of cells was clustered together; the result is shown in Fig. 3-3b. We observe that the chosen clustering given by Seurat is in strong agreement with the other methods and that the clusters are highly robust to the choice of clustering method.

Autoencoder training for integration and translation between single-cell RNA-seq data and single-cell chromatin images

Images were normalized to range between $[0, 1]$ and RNA-seq matrix was $\log(x + 1)$ normalized. Since the imaging dataset is more difficult to reconstruct in comparison to the RNA-seq dataset, we first pretrained the image autoencoder to reconstruct single-cell chromatin images for 850 epochs using the reconstruction loss and the discriminative loss in Equation [3.9]. Subsequently, we trained the full model consisting of the pretrained image autoencoder, the RNA-seq autoencoder, and latent space discriminator using reconstruction loss and discriminative loss with hyperparameters $\lambda_1 = 0.1, \lambda_2 = 1$. The architectures of all networks are shown in Supplementary Table B.4. Models were trained with the Adam optimizer with a learning rate of 1e-3. In Supplementary Fig. B-9 and B-10, we show that our findings are robust to the choice of architecture (fully-connected versus convolutional layers, number of layers, as well as latent space dimension).

Cell culture and immunostaning

CD4+/CD45RA+ naive helper T-cells from human peripheral blood were purchased from AllCells. These cells were revived and cultured in media (RPMI-1640 + 10% FBS + 1% pen-strep) as per the manufacturer's instructions. The cells for the experiments were used within two days upon revival.

Cells in media were allowed to adhere to Poly-lysine coated slides for 30 minutes. Cells were then fixed with 4% Paraformaldehyde (Sigma) for 30 minutes and washed with

PBS three times, which also removed unattached cells. Permeabilization was done with 0.5% Triton X-100 (Sigma) for 10 minutes followed by PBS washes. Blocking was done with 5% BSA in PBS for 30 minutes and incubated with primary and secondary antibodies as per the dilution and incubation time recommended by the manufacturer. The primary antibodies used in this study are anti-RPL10A antibody (Abcam, ab174318, dilution 1/200) and Anti-Coronin 1a/TACO antibody (Abcam ab14787, dilution 1/150). Cells were washed with PBS (+0.1% Tween) three times after primary and secondary antibody incubation. During the final step, excess liquid was removed by slanting the slides. ProLong® Gold Antifade Mountant with DAPI (ThermoFischer Scientific) was added to these slides and allowed to cure for 24 hours. Coverslips were then sealed and imaged using a confocal microscope.

Confocal microscopy and image analysis

1024 × 1024 and 12-bit multi-channel images were obtained using a Nikon A1R confocal microscope. Z-stack images were captured using a 100× objective with a pixel size of 0.1 μm and 0.5 μm depth. Images were processed and further analyzed using custom programs in Fiji and R (see below in code availability).

The nuclear boundaries were segmented in 3D using the DAPI channels to identify individual nuclei. These nuclei were eroded by 0.5 microns in x, y, and z iteratively until the volume of the eroded nucleus was less than 10 cubic microns. Then the mean intensity of each 3D ring (width 0.5 microns) in the nucleus was computed for all cells. The intensity fraction was calculated by normalizing the mean ring intensity for each nucleus (maximum= 1). Linear interpolation was then used to compute the intensity fraction of rings that occupy 0-10% to 90-100% volume fraction of the nucleus. The heatmaps were visualized using functions from gplots, RColorBrewer and dendextend.

In order to calculate the cellular levels of proteins, the 3D nuclear object was dilated by 2 microns in x, y and z. This was efficient as the cells were all spherically shaped with high karyoplasmic index. The total intensity in the 3D cellular object was computed for each protein channel and their ratio was obtained for each cell.

Data availability

The data for model validation on paired single-cell RNA-seq and ATAC-seq is publicly available and was obtained from GSE117089 (Cao et al., 2018). The RNA-seq data for integration of RNA-seq and chromatin images is publicly available and was obtained from `https://support.10xgenomics.com/single-cell-gene-expression/datasets/2.1.0/pbmc8k`. The chromatin images are available at Zenodo from DOI: 10.5281/zenodo.4265737.

Code availability

The code for model training is available at (Yang et al., 2020a): `https://github.com/uhlerlab/cross-modal-autoencoders`. Code containing the image processing scripts for the analysis of the primary images is available at (Yang et al., 2020b): `http://github.com/SaradhaVenkatachalapathy/Radial_chromatin_packing_immune_cells`. Data analysis was performed using standard libraries and software such as scikit-learn, scipy, numpy, seaborn and R.

Chapter 4

Identifying 3D Genome Organization in Diploid Organisms via Euclidean Distance Geometry

Parts of this chapter are under review.
Belyaeva, A., Kubjas, K., Sun, L., & Uhler, C. (2020). Identifying 3D Genome Organization in Diploid Organisms via Euclidean Distance Geometry.
My contributions were to implement the methods, design and perform method and data analysis, and write the manuscript. I include theoretical results and some of the proofs in the main text and the remaining proofs in Appendix C for completeness.

4.1 Summary

The spatial organization of the DNA in the cell nucleus plays an important role for gene regulation, DNA replication, and genomic integrity. Through the development of chromosome conformation capture experiments (such as 3C, 4C, Hi-C) it is now possible to obtain the contact frequencies of the DNA at the whole-genome level. We study the problem of reconstructing the 3D organization of the genome from such whole-genome contact frequencies. A standard approach is to transform the contact frequencies into noisy distance measurements and then apply semidefinite programming (SDP) formulations to obtain the 3D configuration. However, neglected in such reconstructions is the fact that most eukaryotes including humans are diploid and therefore contain two copies of each genomic locus. We prove that the 3D organization of the DNA is not identifiable from distance measurements derived from contact frequencies in diploid organisms. In fact, there are infinitely many solutions even in the noise-free setting. We then discuss various additional biologically relevant and experimentally measurable constraints (including distances between neighboring genomic loci and higher-order interactions), and we prove identifiability under these conditions. Furthermore, we provide SDP formulations for computing the 3D embedding of the DNA with these additional constraints and show that we can recover the true 3D embedding with high accuracy from both noiseless and noisy measurements. Finally, we apply our algorithm to real pairwise and higher-order contact frequency data and show that we can recover known genome organization patterns.

4.2 Introduction

It is now well established that the spatial organization of the genome in the cell nucleus plays an important role for cellular processes including gene regulation, DNA replication, and the maintenance of genomic integrity (Dekker, 2008; Uhler and Shivashankar, 2017a,b). Notably, a recent study (Wang et al., 2018a) showed a causal link between three-dimensional (3D) genome organization and gene regulation, where gene repositioning was induced and subsequent changes in gene expression were observed. This motivates the development of methods to reconstruct the 3D structure of the genome to study its functions.

The genetic information in cells is contained in the DNA, which is organized into chromosomes and packed into the cell nucleus. Chromosome confirmation capture techniques (such as 3C, 4C, Hi-C, Capture-C) have enabled the interrogation of the contact frequencies between pairs of genomic loci at the whole-genome scale (Dekker et al., 2002; Simonis et al., 2006; Lieberman-Aiden et al., 2009; Hughes et al., 2014). In Hi-C, for example, interacting chromosome regions are crosslinked (i.e., frozen), the DNA is then fragmented, the crosslinked fragments are ligated, and paired-end sequencing is applied to the ligation products and mapped to a reference genome (Lieberman-Aiden et al., 2009). By binning the genome and ascribing each read pair into the corresponding bin, one obtains a contact frequency matrix between genomic loci that is commonly of the size $10^6 \times 10^6$.

Different computational approaches for reconstructing the 3D genome organization from contact frequency data have been considered. Distance-based approaches convert contact frequencies f_{ij} into spatial distances d_{ij} and find a Euclidean embedding of the points in 3D (Duan et al., 2010; Zhang et al., 2013; Lesne et al., 2014; Rieber and Mahony, 2017). Ensemble methods such as MCMC5C and BACH (Rousseau et al., 2011; Hu et al., 2013) learn a set of possible 3D structures by defining a probabilistic model for contact frequencies and generating an ensemble of structures via MCMC sampling. Other ensemble methods include molecular dynamics simulations that model DNA as a polymer and output an ensemble of 3D structures (Lieberman-Aiden et al., 2009; Mirny, 2011; Di Pierro et al., 2016; Qi and Zhang, 2019). Finally, statistical methods directly model contact counts instead of distances, using for example the Poisson distribution (Varoquaux et al., 2014), and maximize the log-likelihood of the data to infer the 3D genome organization.

Almost all existing methods make the simplifying assumption that the genome is haploid, when in fact most organisms of interest including humans are diploid, i.e. there are two copies of each chromosome known as *homologous chromosomes*. For example, human cells contain two copies of 23 chromosomes each. The challenge is that the contact frequency data from chromosome conformation capture experiments is generally *unphased*, meaning that the copies of each chromosome cannot be distinguished. As a result, if the DNA is modeled as a string of beads containing two copies of each bead i for $1 \leq i \leq n$ (Figure 4-1), then the measured contact frequencies result in an $n \times n$ matrix, from which we would like to infer the 3D embedding of $2n$ points. This problem cannot be solved by classical methods for 3D genome reconstruction such as those mentioned above. With significant experimental efforts, phased data can be obtained (1000 Genomes Project Consortium et al., 2012, 2015) and used in order to reconstruct the 3D genome organization (Cauer et al., 2019). However, such data is rare and costly.

In this study, we provide a computational method for inferring the 3D diploid organization of the genome without relying on phased data. In particular, we consider a distance-based approach and use Euclidean distance geometry to obtain the 3D diploid structure of the genome. The precise mathematical problem considered in this study is as follows and illustrated in Figure 4-1. DNA is modeled as a string of beads, that contains two copies of each bead $1 \leq i \leq n$. We would like to infer the location of the two copies of each bead, which we denote by $x_i \in \mathbb{R}^3$ and $y_i \in \mathbb{R}^3$. Since for unphased data, the two copies of each bead cannot be distinguished, the problem is to identify the 3D configuration ($2n \times 3$ matrix), i.e. $x_1, \ldots x_n, y_1, \ldots y_n \in \mathbb{R}^3$ (up to translation and rotation), from the composite distance measurements D_{ij}, $1 \leq i \neq j \leq n$ ($n \times n$ matrix), corresponding to the sum of the distances between either copy of bead i and j, i.e.,

$$D_{ij} = \|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2.$$

In the haploid or phased setting, this problem boils down to the standard Euclidean distance geometry problem. This problem has a long history: in the classical setting with no missing values, this problem can be solved via the classical multidimensional scaling (cMDS) algorithm that is based on spectral decomposition followed by dimensionality reduction; see (Cox and Cox, 2000) for an overview. Other approaches for the Euclidean embedding and completion problems, including in the presence of missing values, are non-convex formulations (Fang and O'Leary, 2012; Mishra et al., 2011) as well as semidefinite relaxations (Alfakih et al., 1999; Fazel et al., 2003; Cayton and Dasgupta; Lu et al., 2005; Weinberger et al., 2007; Zhang et al., 2016).

A naive approach in the unphased diploid setting is to assume that the four distances that make up our measured composite distance D_{ij} are equal and solve the corresponding Euclidean embedding problem. However, it is evident from single-cell imaging studies that the four distances in D_{ij} can be wildly different (Bolzer et al.,

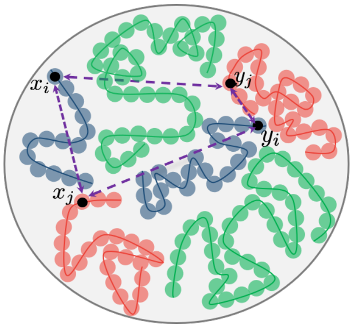


Figure 4-1: Schematic of the diploid genome. Nucleus with green, blue and red curves depicting three homologous pairs of chromosomes. In the unphased setting, the measured distance between loci i and j corresponds to the sum of the four distances (denoted in purple) between two pairs of homologous loci x_i, y_i and x_j, y_j.

2005; Nir et al., 2018). Hence this approach cannot provide realistic embeddings. While, a simple dimension argument ($6n$ variables versus $\binom{n}{2}$ constraints) suggests that the 3D genome configuration could be uniquely identifiable, one of the main results of our study is that the 3D diploid genome configuration is not identifiable from unphased data. In fact, we show that there are infinitely many configurations that satisfy the constraints imposed by D_{ij}, even in the noiseless setting (Section 4.3, Theorem 4.3.1).

We therefore consider additional biologically relevant and experimentally measurable constraints and study identifiability of the 3D diploid structure under these constraints. First, we take into account distances between neighboring beads, i.e. $\|x_i - x_{i+1}\|^2$ and $\|y_i - y_{i+1}\|^2$ on each chromosome. While we show that this yields unique identifiability for configurations in 2D, there are still infinitely many configurations in 3D, which is of primary interest for genome modeling (Section 4.4, Propositions 4.4.1, 4.4.2). To obtain identifiability, we consider adding constraints based on contact frequencies between three or more loci simultaneously. The measurement of such higher-order contact frequencies has recently been enabled by experimental assays such as SPRITE (Quinodoz et al., 2018), C-walks (Olivares-Chauvet et al., 2016) and GAM (Beagrie et al., 2017). We prove that this information can be used to uniquely identify the 3D genome organization from unphased data in the noiseless setting (Section 4.5, Theorem 4.5.1).

Finally, we provide an SDP formulation for obtaining the 3D diploid configuration from noisy measurements (Section 4.6) and show based on simulated data that our algorithm has good performance and that it is able to recover known genome organization patterns when applied to real contact frequency data collected from human lymphoblastoid cells (Section 4.7).

4.3 Unidentifiability from pairwise distance constraints

We denote the true but unknown coordinates of the homologous loci by x_i^* and y_i^* and the corresponding noiseless distances by D_{ij}^* while the symbols x_i and y_i denote the variables that we want to solve for. From a biological perspective the relevant setting is when $x_i, y_i \in \mathbb{R}^3$. However, many of our results hold more generally and we will state these in $\mathbb{R}^d$. The main result of this section is Theorem 4.3.1, which characterizes the set of solutions given by the constraints D_{ij}^*. In particular, it establishes non-identifiability of the 3D genome structure from pairwise distance measurements in unphased data.

Theorem 4.3.1. *The set of points $(x_1, \ldots, x_n, y_1, \ldots, y_n) \in (\mathbb{R}^d)^{2n}$ satisfying*

$$D_{ij}^* = \|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2 \quad \textit{for all} \;\; 1 \leq i \neq j \leq n \tag{4.1}$$

is equal (up to translations and rotations in $\mathbb{R}^d$ and permutations of x_i and y_i) to the set of points satisfying

$$x_i + y_i = x_i^* + y_i^* \quad \textit{and} \quad \|x_i\|^2 + \|y_i\|^2 = \|x_i^*\|^2 + \|y_i^*\|^2 \quad \textit{for all} \;\; 1 \leq i \leq n. \tag{4.2}$$

As a consequence, the measurements D_{ij}^* identify the location of each pair of homologous loci (x_i, y_i) up to a sphere with center $(x_i^* + y_i^*)/2$ and radius $\|x_i^* - y_i^*\|/2$. Namely, the points x_i, y_i lie opposite to each other anywhere on this sphere. Unless $x_i^* = y_i^*$ for all i, i.e., all spheres have radius 0, this set is infinite in dimensions $d > 1$

and hence the configuration is unidentifiable.

In the remainder of this section, we will prove Theorem 4.3.1. The two inclusions in Theorem 4.3.1 are proven in Lemmas 4.3.2 and 4.3.4. In Lemma 4.3.3 it is shown that the distance $\|x_i - y_i\|$ within each homologous pair is fixed given the pairwise distances D^*_{ij}. This result is used to prove Lemma 4.3.4.

Lemma 4.3.2. *Let* $(x_1, \dots, x_n, y_1, \dots, y_n) \in (\mathbb{R}^d)^{2n}$ *satisfy*

$$x_i + y_i = x_i^* + y_i^* \text{ and } \|x_i\|^2 + \|y_i\|^2 = \|x_i^*\|^2 + \|y_i^*\|^2 \text{ for all } 1 \le i \le n. \tag{4.3}$$

Then

$$\|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2 = D^*_{ij} \text{ for all } 1 \le i \ne j \le n.$$

Proof. Observe that for each pair x_i, y_i satisfying the equations (4.3), it holds that

$$\begin{aligned} D^*_{ij} &= 2 \cdot (\|x_i^*\|^2 + \|y_i^*\|^2) + 2 \cdot (\|x_j^*\|^2 + \|y_j^*\|^2) - 2(x_i^* + y_i^*) \cdot (x_j^* + y_j^*) \\ &= 2 \cdot (\|x_i\|^2 + \|y_i\|^2) + 2 \cdot (\|x_j\|^2 + \|y_j\|^2) - 2(x_i + y_i) \cdot (x_j + y_j) \\ &= \|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2. \end{aligned}$$

This completes the proof. □

Next we will show that the distance between homologous pairs is uniquely determined by the D^*_{ij}

Lemma 4.3.3. *Let $d \le 3$ and $n \ge 2d + 3$. Then for each $1 \le i \le n$ the quantity $\|x_i - y_i\|$ is identifiable from the constraints imposed by the D^*_{ij}, i.e., for any solution $(x_1, \dots, x_n, y_1, \dots, y_n) \in (\mathbb{R}^d)^{2n}$ to the equations defined by the D^*_{ij} in (4.1), the quantity $\|x_i - y_i\|$ is constant.*

The constraint $d \le 3$ is due to our proof technique. The condition $n \ge 2d + 3$ is necessary for unique identifiability of the distance between homologous pairs of loci.

Proof. Without loss of generality we assume that $i = 1$ and show that $\|x_1 - y_1\|$

is equal to some constant. First, we perform a shift on the solution so that $x_1 = -y_1 = v$. Since shifts preserve distances, they in particular preserve the equality constraints (4.1). Hence,

$$D^*_{1j} = \|v - x_j\|^2 + \|v - y_j\|^2 + \| - v - x_j\|^2 + \| - v - y_j\|^2.$$

Expanding this out into dot products and simplifying yields

$$D^*_{1j} = 4\|v\|^2 + 2(\|x_j\|^2 + \|y_j\|^2).$$

Let $j \neq k$ be both not equal to 1. Then substituting the above leads to

$$D^*_{1j} + D^*_{1k} - D^*_{jk} = 8\|v\|^2 + 2(x_j + y_j) \cdot (x_k + y_k).$$

Defining $T_{jk} := D^*_{1j} + D^*_{1k} - D^*_{jk}$ and $s_j := \sqrt{2}(x_j + y_j)$, this is equivalent to

$$T_{jk} - 8\|v\|^2 = s_j \cdot s_k.$$

Let T' be the $(d+1) \times (d+1)$ submatrix of T satisfying $T'_{ij} = T_{i+1,j+d+2}$, i.e. the rows of T' correspond to the rows $2, 3, \ldots, d+2$ of T and the columns of T' correspond to the columns $d+3, d+4, \ldots, 2d+3$ of T. We now show that for generic configurations $\det(T') \neq 0$. Since $\det(T')$ can be written as a polynomial in the coordinates x_i and y_i, then $\det(T') \neq 0$ for generic configurations as long as it does not identically vanish. Hence it suffices to present one configuration where $\det(T')$ is nonzero. For $d \leq 3$ we can check this using random configurations.

Since T' has full rank, then the matrix determinant lemma implies that

$$\det(T' - 8J\|v\|^2) = (1 - 8\|v\|^2 1^T (T')^{-1} 1) \det(T'), \tag{4.4}$$

where 1 denotes the all ones vector. Note that the scalar $1^T T'^{-1} 1$ is fixed and $(\det T') \neq 0$. Furthermore, since $T' - 8J\|v\|^2$ is formed from the dot products between d-dimensional vectors, it has rank at most d and therefore $\det(T' - 8J\|v\|^2) = 0$ due to

$T' - 8J\|v\|^2$ being a $(d+1)\times(d+1)$ matrix. Hence, $(1-8\|v\|^2 1^T(T')^{-1}1)\det(T') = 0$, which is a linear equation in terms of $\|v\|^2$. As a consequence, it has a unique solution for $\|v\|^2$ and thus the distance between the homologous pair x_1, y_1 is fixed as long as $n \geq 2d+3$. □

We next characterize all solutions to the constraints imposed by the D^*_{ij}.

Lemma 4.3.4. *Let* $(x_1, \ldots, x_n, y_1, \ldots, y_n) \in (\mathbb{R}^d)^{2n}$ *be a solution to*

$$\|x_i - x_j\|^2 + \|x_i - y_j\|^2 + \|y_i - x_j\|^2 + \|y_i - y_j\|^2 = D^*_{ij} \text{ for all } 1 \leq i \neq j \leq n.$$

Then

$$x_i + y_i = x^*_i + y^*_i \text{ and } \|x_i\|^2 + \|y_i\|^2 = \|x^*_i\|^2 + \|y^*_i\|^2 \text{ for all } 1 \leq i \leq n$$

up to translations and rotations in $\mathbb{R}^d$ *and permutations of* x_i *and* y_i.

Proof. Without loss of generality we perform a translation on the solution such that $x_1 = -y_1 = v$ for some vector v. By Lemma 4.3.3 the quantity $\|x_k - y_k\|$ is constant for each $1 \leq k \leq n$ and thus also $\|v\|$ is constant. Since for any $j \neq 1$ it holds that $D^*_{1j} = 4\|v\|^2 + 2(\|x_j\|^2 + \|y_j\|^2)$, also $\|x_j\|^2 + \|y_j\|^2$ is constant and hence $\|x_i\|^2 + \|y_i\|^2 = \|x^*_i\|^2 + \|y^*_i\|^2$ for all $1 \leq i \leq n$.

Similarly to the proof of Lemma 4.3.3, if we define $T_{jk} = D^*_{1j} + D^*_{1k} - D^*_{jk}$ and $s_j = \sqrt{2}(x_j + y_j)$, we find that

$$T_{jk} - 8\|v\|^2 = s_j \cdot s_k.$$

Because we have access to the diagonal constraints now, this relationship holds for all j, k and not just $j \neq k$. Thus $T - 8J\|v\|^2$ is a symmetric $(n-1)\times(n-1)$ matrix admitting a rank d factorization. Let S be the matrix formed with the vectors s_j. We then have $T - 8J\|v\|^2 = SS^T$. There is a result on rank factorizations of symmetric matrices that any other factorization $T - 8J\|v\|^2 = S'S'^T$ satisfies $S = S'Q$ for some orthogonal matrix Q (Krislock, 2010, Proposition 3.2). Thus for any other solution

s'_j, we have $s_j = s'_j Q$, implying all solutions are simply orthogonal transformations of each other (rotations, reflections, etc.)

In summary, we have shown that once we have fixed $x_1 + y_1 = 0$ via translation, then the quantities $x_j + y_j$ are unique up to orthogonal transformations and the quantities $\|x_j\|^2 + \|y_j\|^2$ are unique. □

4.4 Distance constraints between neighboring loci

In Section 4.3, we showed that the 3D genome configuration is not identifiable from pairwise distance constraints available from typical (unphased) contact frequency maps. In order to gain identifiability, we next consider adding other biological constraints to the problem formulation that are generally available or can be measured. In particular, since DNA can be viewed as a string of connected beads, we use the distance between adjacent beads as an additional constraint. The distance between neighboring beads can be derived empirically or from imaging studies (Müller et al., 2010; Jungmann et al., 2014); see also our experimental results in Section 4.7. The additional mathematical constraints are:

$$\|x_i - x_{i+1}\| = \|x_i^* - x_{i+1}^*\| \text{ and } \|y_i - y_{i+1}\| = \|y_i^* - y_{i+1}^*\| \text{ for } 1 \leq i \leq n-1,$$

where $x_1^*, x_2^*, \ldots, x_n^*$ and $y_1^*, y_2^*, \ldots, y_n^*$ correspond to consecutive beads on homologous chromosomes; see Figure 4-2.

In this section we show the following results: under the additional distance constraints between neighboring loci, we prove that identifiability can be obtained in the 2D setting (Proposition 4.4.1). However, in the 3D setting we prove that there are still infinitely many 3D configurations even with these additional distance constraints (Proposition 4.4.2).

Proposition 4.4.1. *For $n \geq 3$, there are unique points $x_1, \ldots, x_n, y_1, \ldots, y_n \in \mathbb{R}^2$*

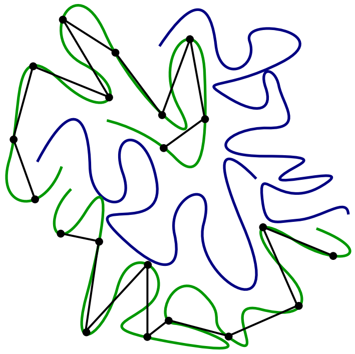


Figure 4-2: Distance constraints between neighboring beads. Green and blue curves depict two homologous pairs of chromosomes. For the green curves distances between neighboring genomic regions are shown by black lines.

satisfying equations

$$\begin{aligned} x_i + y_i = x_i^* + y_i^* \text{ and } \|x_i\|^2 + \|y_i\|^2 = \|x_i^*\|^2 + \|y_i^*\|^2 \text{ for } 1 \le i \le n, \\ \|x_i - x_{i+1}\| = \|x_i^* - x_{i+1}^*\| \text{ and } \|y_i - y_{i+1}\| = \|y_i^* - y_{i+1}^*\| \text{ for } 1 \le i \le n-1. \end{aligned} \tag{4.5}$$

We provide the proofs for Proposition 4.4.1 in Appendix C.

Despite having uniqueness in 2D, we do not have uniqueness in 3D as shown in the following proposition.

Proposition 4.4.2. *For any $n \in \mathbb{N}$, there exist $x_1^*, x_2^*, \ldots, x_n^*$ and $y_1^*, y_2^*, \ldots, y_n^*$ such that there are infinitely many points $x_1, \ldots, x_n, y_1, \ldots, y_n \in \mathbb{R}^3$ satisfying equations (4.5).*

We provide the proofs for Proposition 4.4.2 in Appendix C.

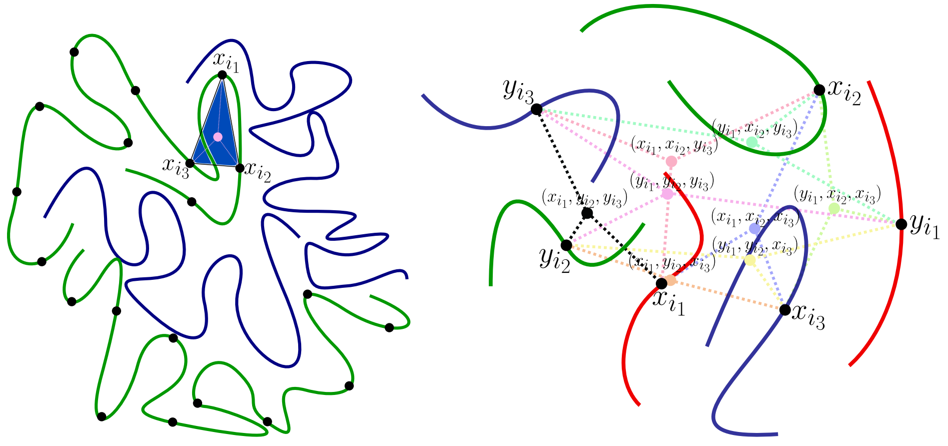
4.5 Identifiability from higher-order contact constraints

In Section 4.4, we showed that considering distances between neighboring beads only yields identifiability in 2D but not in 3D. In the following, we consider adding further constraints that are becoming widely available from experimental data, mainly higher-order contact frequencies between three or more loci as measured by experimental assays such as SPRITE (Quinodoz et al., 2018), C-walks (Olivares-Chauvet et al., 2016) and GAM (Beagrie et al., 2017). We express these constraints mathematically by letting $f \in \mathbb{R}^{m \times m \times \dots \times m}$ be a contact frequency tensor, where $f_{x_{i_1}, x_{i_2}, \dots, x_{i_k}}$ measures the contact frequency between loci $i_1, i_2, \dots, i_k$ with coordinates $x_{i_1}, x_{i_2}, \dots, x_{i_k}$. In the unphased setting we can only measure a combination of contact frequencies over the homologous loci $\{x_{i_1}, y_{i_1}\} \times \{x_{i_2}, y_{i_2}\} \times \dots \times \{x_{i_k}, y_{i_k}\}$, which we denote by $F_{i_1 i_2 \dots i_k}$. In addition, as commonly done we turn contact frequencies into distances by defining $D_{i_1 i_2 \dots i_k} := 1/F_{i_1 i_2 \dots i_k}$.

In the following we describe how to derive constraints on the 3D location of the genomic loci from such measured higher-order distances. For simplicity, we first describe the higher-order distance constraints in the phased setting and only for three loci. Since the contact frequency measures when all three loci come together, we define $D_{i_1 i_2 i_3}$ as the sum of the distances of the three loci $x_{i_1}, x_{i_2}, x_{i_3}$ to their centroid (Figure 4-3a). We next generalize this distance to the unphased setting. For three homologous loci $(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2})$ and (x_{i_3}, y_{i_3}), these contact frequencies can be formed by 8 possible triples, namely $(x_{i_1}, x_{i_2}, x_{i_3})$, $(x_{i_1}, x_{i_2}, y_{i_3})$, $(x_{i_1}, y_{i_2}, x_{i_3})$, $(y_{i_1}, x_{i_2}, x_{i_3})$, $(x_{i_1}, y_{i_2}, y_{i_3})$, $(y_{i_1}, y_{i_2}, x_{i_3})$, $(y_{i_1}, x_{i_2}, y_{i_3})$, $(y_{i_1}, y_{i_2}, y_{i_3})$. We will assume that one of the triples constitutes the majority of the observed contact frequency count and hence we define the $D_{i_1 i_2 i_3}$ as the minimum over all 8 higher-order distances. This is illustrated in Figure 4-3b. Generalizing from three to k loci, the distance constraint becomes

$$D_{i_1 i_2 \dots i_k} = \min_{z_{i_j} \in \{x_{i_j}, y_{i_j}\}} \left(\sum_{j=1}^{k} \| z_{i_j} - (z_{i_1} + \dots + z_{i_k})/k \|^2 \right).$$

In the following, we prove our main result; namely we show that the distance constraints of order 3 together with the previously considered pairwise distance constraints and distance constraints among consecutive beads results in unique identifiability of the 3D genome configuration (Theorem 4.5.1). In fact, only very few order 3 distance constraints are required for unique identifiability. As we show in Theorem it is sufficient that the first and last bead of each chromosome be contained


x_{i_1}
x_{i_3}
x_{i_2}
y_{i_3}
x_{i_2}
$(y_{i_1}, x_{i_2}, y_{i_3})$
$(x_{i_1}, x_{i_2}, y_{i_3})$
$(y_{i_1}, y_{i_2}, y_{i_3})$
$(x_{i_1}, y_{i_2}, y_{i_3})$
$(y_{i_1}, x_{i_2}, x_{i_3})$
$(x_{i_1}, x_{i_2}, x_{i_3})$
y_{i_1}
$(y_{i_1}, y_{i_2}, x_{i_3})$
y_{i_2}
$(x_{i_1}, y_{i_2}, x_{i_3})$
x_{i_1}
x_{i_3}

(a) (b)

Figure 4-3: Higher-order distance constraints. (a) Three loci $x_{i_1}, x_{i_2}, x_{i_3}$, located on the same chromosome, are depicted. In the phased setting, the higher-order distance constraint $D_{i_1 i_2 i_3}$ is defined as the sum of the distances (pink dashed lines) of the three loci $x_{i_1}, x_{i_2}, x_{i_3}$ to their centroid (pink point). Green and blue depict two different chromosomes. (b) This figure illustrates the definition of $D_{i_1 i_2 i_3}$ in the unphased setting. Green, blue and red curves depict neighborhoods around three homologous loci $(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2})$ and (x_{i_3}, y_{i_3}). From these homologous loci 8 possible higher-order distance constraints can be defined (colored dashed lines) based on the 8 centroids depicted in the figure. The chosen higher-order distance constraint $D_{i_1 i_2 i_3}$ is the minimum of the possible 8 higher-order distance constraints (achieved here by the three black dashed line segments).

in an order 3 distance constraint. This is a reasonable constraint given that methods such as SPRITE, C-walks and GAM measure higher-order interactions over the whole genome. These insights are of interest experimentally since they suggest that the methods can restrict the measurement of such higher-order constraints to first and last beads of each chromosome, known as telomeres.

Theorem 4.5.1. *Let m be the number of chromosome pairs, let $n_1, n_2, \ldots, n_m$ be the number of domains on chromosomes $1, 2, \ldots, m$ and define $n = n_1 + n_2 + \ldots + n_m$. Let $I \subseteq [n] \times [n] \times [n]$ be such that each of $1, n_1, n_1+1, n_1+n_2, \ldots, n_1+n_2+\ldots+n_{m-1}+1, n$ (labels of domains at the beginning and at the end of each chromosome) is contained in at least one triple in I. Let $x_1^*, \ldots, x_n^*, y_1^*, \ldots, y_n^* \in \mathbb{R}^3$ be fixed such that*

$$\min_{z_i^* \in \{x_i^*, y_i^*\} \textit{ for } i=k_1,k_2,k_3} \left(\sum_{j \in \{k_1,k_2,k_3\}} \|z_j^* - (z_{k_1}^* + z_{k_2}^* + z_{k_3}^*)/3\|^2 \right) = 0 \textit{ for } (k_1, k_2, k_3) \in I.$$

Consider the polynomial system:

$$x_i + y_i = x_i^* + y_i^* \textit{ and } \|x_i\|^2 + \|y_i\|^2 = \|x_i^*\|^2 + \|y_i^*\|^2 \textit{ for } 1 \leq i \leq n,$$
$$\|x_i - x_{i+1}\| = \|x_i^* - x_{i+1}^*\| \textit{ and } \|y_i - y_{i+1}\| = \|y_i^* - y_{i+1}^*\| \textit{ for } i \in [n]\backslash\{n_1, n_1 + n_2, \ldots, n\},$$
$$\min_{z_i \in \{x_i, y_i\} \textit{ for } i=k_1,k_2,k_3} \left(\sum_{j \in \{k_1,k_2,k_3\}} \|z_j - (z_{k_1} + z_{k_2} + z_{k_3})/3\|^2 \right) = 0 \textit{ for } (k_1, k_2, k_3) \in I. \tag{4.6}$$

For generic $x_1^*, \ldots, x_n^*, y_1^*, \ldots, y_n^*$, this system has a unique solution in $(\mathbb{R}^3)^{2n}$.

We provide the proof for Theorem 4.5.1 in Appendix C.

4.6 Algorithms and implementation

So far, we derived a theoretical framework to establish when we have unique and finite identifiability of the 3D configuration in the noiseless setting. However, a unique solution does not necessarily mean that we can find it efficiently, as in many cases finding the solution may be NP-hard. In addition, we have so far not yet considered the noisy setting. In this section, we show how to construct an optimization formulation to determine the 3D configuration efficiently.

We frame the 3D reconstruction problem as a Euclidean embedding problem, where the coordinates $x_1, \ldots x_n, y_1, \ldots y_n \in \mathbb{R}^3$ are inferred from distances. Similar to ChromSDE (Zhang et al., 2013), we formulate all distances in terms of entries in the Gram matrix G, which tracks the dot products between the $2n$ genomic regions. Namely, letting the column/row i of G correspond to x_i and the column/row $n + i$ correspond to its homologous locus y_i, then the distances are given by $\|x_i - x_j\|^2 = G_{i,i} + G_{j,j} - 2G_{i,j}$, $\|x_i - y_j\|^2 = G_{i,i} + G_{n+j,n+j} - 2G_{i,n+j}$ and $\|y_i - y_j\|^2 = G_{n+i,n+i} + G_{n+j,n+j} - 2G_{n+i,n+j}$. It is natural to work with the Gram matrix G, since it is rotation invariant. By imposing the constraint $\sum_{i,j} G_{i,j} = 0$ we can also fix the translational axis. Also the additional distance constraints that we introduced in the previous sections (Lemma 4.3.3, Proposition 4.4.2, Theorem 4.5.1) can be represented as linear constraints in terms of entries in G as follows:

- Pairwise distance constraints:

$$g_{ij}(G) := G_{i,i} + G_{j,j} + G_{n+i,n+i} + G_{n+j,n+j} - G_{i,j} - G_{n+i,j} - G_{i,n+j} - G_{n+i,n+j}$$

- Distances between homologous pairs:

$$g_{ii}(G) := G_{i,i} + G_{n+i,n+i} - 2G_{i,n+i}$$

- Distances between neighboring beads:

$$g_{i+}(G) := G_{i,i} + G_{i+1,i+1} - 2G_{i,i+1}$$

- Higher-order distance constraints:

$$g_{ijk}(G) := \min_{l}(g_{ijkl} : l = 1, \dots, 8)$$

  where

$$\begin{aligned}
g_{ijk1}(G) :=& G_{i,i} + G_{j,j} + G_{k,k} - G_{i,j} - G_{i,k} - G_{j,k},\\
g_{ijk2}(G) :=& G_{i,i} + G_{j,j} + G_{n+k,n+k} - G_{i,j} - G_{i,n+k} - G_{j,n+k},\\
g_{ijk3}(G) :=& G_{i,i} + G_{n+j,n+j} + G_{k,k} - G_{i,n+j} - G_{i,k} - G_{n+j,k},\\
g_{ijk4}(G) :=& G_{i,i} + G_{n+j,n+j} + G_{n+k,n+k} - G_{i,n+j} - G_{i,n+k} - G_{n+j,n+k},\\
g_{ijk5}(G) :=& G_{n+i,n+i} + G_{j,j} + G_{k,k} - G_{n+i,j} - G_{n+i,k} - G_{j,k},\\
g_{ijk6}(G) :=& G_{n+i,n+i} + G_{j,j} + G_{n+k,n+k} - G_{n+i,j} - G_{n+i,n+k} - G_{j,n+k},\\
g_{ijk7}(G) :=& G_{n+i,n+i} + G_{n+j,n+j} + G_{k,k} - G_{n+i,n+j} - G_{n+i,k} - G_{n+j,k},\\
g_{ijk8}(G) :=& G_{n+i,n+i} + G_{n+j,n+j} + G_{n+k,n+k} - G_{n+i,n+j} - G_{n+i,n+k} - G_{n+j,n+k}
\end{aligned}$$

Our objective is to determine a rank 3 solution of G, satisfying the above constraints. However, this optimization problem is non-convex due to the rank constraint, and we instead consider the standard relaxation: we minimize the trace of the Gram matrix as an approximation to matrix rank (Fazel et al., 2001). The resulting optimization problem then becomes the following semidefinite program (SDP):

$$\begin{aligned}
\underset{G}{\text{minimize}} \quad & \operatorname{tr}(G) \\
\text{subject to} \quad & g_{ii}(G) = D^*_{ii},\ 1 \leq i \leq n, \\
& g_{ij}(G) = D^*_{ij},\ 1 \leq i < j \leq n, \\
& g_{i+}(G) = D^*_{i+},\ i \in \Omega_1, \\
& g_{ijk}(G) = D^*_{ijk},\ (i,j,k) \in \Omega_2, \\
& \sum_{1 \leq i,j \leq 2n} G_{i,j} = 0, \\
& G \succeq 0.
\end{aligned} \tag{4.7}$$

Here, D^*_{ii} denote the distances between homologous pairs computed from the pairwise distances using Lemma 4.3.3, D^*_{ij} denote the pairwise distances, D^*_{i+} denote the distances between neighboring beads, and D^*_{ijk} denote the distances between three loci (while one could also consider 4 or higher-order distance constraints, in our implementation, we only used three-way distance constraints). The index set $\Omega_1 = [2n]\backslash\{n_1, n_1+n_2, \ldots, n, n+n_1, n+n_1+n_2, \ldots, 2n\}$ corresponds to all beads that are not the last bead on a chromosome. The index set $\Omega_2 \subseteq [n]^3$ corresponds to all triples of beads with non-zero contact frequencies.

In the noisy setting, which is relevant for biological data, we replace the equality constraints by penalties in the loss function. Namely, using D^* for the noiseless and D for the noisy distances, we replace the equality constraints of the form $g(G) = D^*$ by adding $(g(G) - D)^2$ to the objective function. For the higher-order distance constraints of the form $D^*_{ijk} = \min(g_{ijk1}(G), \ldots, g_{ijk8}(G))$ for $(i,j,k) \in \Omega_2$ we use slack variables and a convex relaxation using an atomic norm that combines the ℓ_2- and ℓ_1-norms. More precisely, we propose the use of the following transformation in the noisy setting,

$$D_{ijk} + \lambda_{ijkl} = g_{ijkl}(G) + s_{ijkl} \text{ for } l = 1, 2, \ldots, 8,$$

where $\lambda_{ijkl}, s_{ijkl} \geq 0$ for all i, j, k, l act as slack variables. In general, for each triple

(i, j, k) we want one of the λ_{ijkl} to be close to 0 and the sum over all s_{ijkl} to be small. Naively this can be done by placing $\sum s_{ijkl} + \sum \lambda_{ijkl}$ into the objective function. However, this would not enforce for each (i, j, k) at least one λ_{ijkl} to be close to 0. Instead we propose to use

$$\sum_{(i,j,k)\in\Omega_2, 1\le l\le 8} s_{ijkl} + \sqrt{\sum_{(i,j,k)\in\Omega_2} \left(\sum_{1\le l\le 8} \lambda_{ijkl}\right)^2}.$$

The ℓ_2-norm will push down the $\sum_l \lambda_{ijkl}$ for each (i, j, k), while the ℓ_1 norm will drive at least one of the λ_{ijkl} to zero, which is precisely the desired behavior. The quantity $\sqrt{\sum_{i,j,k} \left(\sum_l \lambda_{ijkl}\right)^2}$ is an atomic norm as defined in (Chandrasekaran et al., 2012) with the set of atoms

$$\mathcal{A} = \{(\lambda_{ijkl}) : \sum_{i,j,k} \left(\sum_l \lambda_{ijkl}\right)^2 = 1 \text{ and } \sum_{i,j,k} \lambda^2_{ijkl_{ijk}} = 1 \text{ for } l_{ijk} = 1, \ldots, 8, (i, j, k) \in \Omega_2\}.$$

Then the optimization problem in the noisy setting becomes:

$$\begin{aligned}
\underset{G,s,\lambda}{\text{minimize}} \quad & \rho \operatorname{tr}(G) + \sum_{1\le i\le n} (g_{ii}(G) - D_{ii})^2 + \sum_{1\le i<j\le n} (g_{ij}(G) - D_{ij})^2 \\
& + \sum_{i\in\Omega_1} (g_{i+}(G) - D_{i+})^2 + \sum_{(i,j,k)\in\Omega_2, 1\le l\le 8} s_{ijkl} + \sqrt{\sum_{(i,j,k)\in\Omega_2} \left(\sum_{1\le l\le 8} \lambda_{ijkl}\right)^2} \\
\text{subject to} \quad & D_{ijk} + \lambda_{ijkl} = g_{ijkl}(G) + s_{ijkl},\ (i, j, k) \in \Omega_2, 1 \le l \le 8, \\
& s_{ijkl} \ge 0,\ (i, j, k) \in \Omega_2, 1 \le l \le 8, \\
& \lambda_{ijkl} \ge 0,\ (i, j, k) \in \Omega_2, 1 \le l \le 8, \\
& \sum_{1\le i,j\le 2n} G_{i,j} = 0, \\
& G \succeq 0.
\end{aligned} \tag{4.8}$$

We use a tuning parameter ρ for the trace in the objective function, which can be used to balance low-rank versus satisfying the constraints. The tuning parameter ρ can be chosen using cross-validation or by selecting it so that the resulting solution has small $(d+1)^{th}$ eigenvalue. As shown in Appendix C (Figures C-4, C-9, C-10) we observe on synthetic and real data that the solution is robust to the choice of ρ.

The theoretical results from Lemma 4.3.3 allow us to compute the distances between homologous pairs from the pairwise distances D_{ij}. We recall that we need to compute $\|v\|^2$ such that

$$\det(T' - 8J\|v\|^2) = 0,$$

where T' is an invertible matrix constructed from the pairwise distance matrix by selecting a set of $2d+2$ indices. One step of computing $\|v\|$ involves inverting T'. Even if the error in the measurements is small, noise can propagate and severely impact this computation. In order to obtain a robust estimate of homolog-homolog distances, for each locus i, we sample 100 T' matrices and obtain 100 solutions to the equation for $\|v\|^2$. We then take the median of the solutions to be the homolog-homolog distance for locus i and use these homolog-homolog distances for the evaluation of our algorithms for synthetic and real data in the following section.

To solve the two convex optimization problems presented in this section for the noiseless and noisy setting, we make use of the solver MOSEK implemented in CVX within MATLAB. This results in the Gram matrix. In order to reconstruct the coordinates of the genomic regions from the Gram matrix, we use an eigenvector decomposition as also done in (Zhang et al., 2013), namely: letting $\gamma_1, \ldots, \gamma_d$ be the top d eigenvalues and $\nu_1, \ldots, \nu_d$ the corresponding eigenvectors of G, then

$$x_i = (\sqrt{\gamma_1} \cdot \nu_{1,i}, \ldots, \sqrt{\gamma_d} \cdot \nu_{d,i}) \text{ and } y_i = (\sqrt{\gamma_1} \cdot \nu_{1,n+i}, \ldots, \sqrt{\gamma_d} \cdot \nu_{d,n+i}) \text{ for } i = 1, \ldots, n.$$

Since we are interested in recovering the genome configuration in 3D, we use $d = 3$, thereby obtaining the desired 3D diploid configuration.

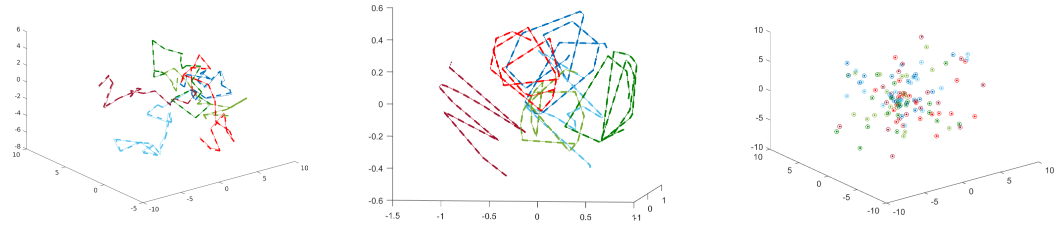
4.7 Evaluation on synthetic and real data

Synthetic data

We start by testing our method on simulated data. For this we construct three different types of 3D structures: (a) a Brownian motion model using a standard normal distribution to generate successive points; (b) points sampled uniformly along a spiral with random translations sampled uniformly within $(0, 0.5)$ range and orientations sampled uniformly within $(-\frac{\pi}{4}, \frac{\pi}{4})$; (c) points sampled uniformly in a unit sphere.

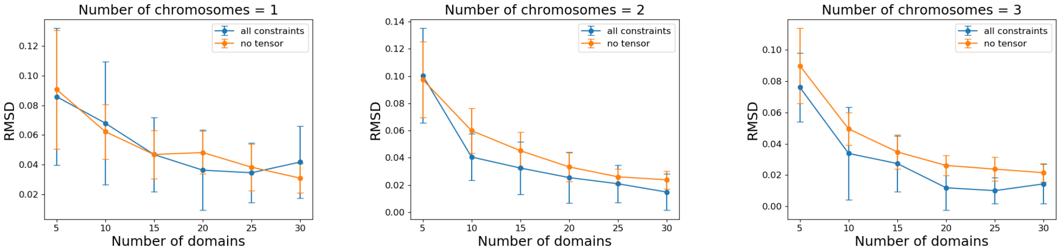
Performance of our method in the noiseless setting. For the dimension one case we deduced in Section 4.3 that the pairwise distance constraints by themselves are sufficient to identify the underlying 3D configuration. For the dimension two case we proved in Section 4.4 that knowing additionally the distances between neighboring beads leads to uniqueness. We here perform simulations in dimension 3 since this is the biologically relevant setting. These results are depicted in Figure 4-4 with additional examples in Figure C-2. The input to our algorithm are the pairwise distances (which are summed over homologs), all three-way distances, the distances between homologous loci, and the distances between neighboring beads. In the noiseless setting considered here we solve the SDP formulation in Equation (4.7). Figure 4-4 and Figure C-2 show that the true and reconstructed structures highly overlap, thereby indicating that our optimization formulation is able to recover the 3D structure of the full diploid genome in the noiseless setting. When the three-way distance constraints are removed, the reconstructions are less aligned with the true structures. This is shown in Figure 4-5, where we measure the root-mean-square deviation (RMSD) between true and reconstructed 3D coordinates over 20 trials. In line with our theoretical results, these experimental results in the noiseless setting indicate the importance of higher-order contact frequencies for recovering the 3D diploid configuration, especially when the number of chromosomes is high.

Performance of our method in the noisy setting. Next, we consider noisy distance observations $D_{ij} = D^*_{ij}(1 + \delta)$ and noisy three-way distance observations



(a) (b) (c)

Figure 4-4: Examples of true and reconstructed points on simulated data. (a) Brownian motion model. (b) Spirals. (c) Random points in a sphere. We generate six chromosomes with in total of 120 domains, corresponding to three homologous pairs with 20 domains per chromosome in the noiseless setting. Solid lines / points correspond to the true 3D coordinates and dashed lines / unfilled points to the reconstructions via our method. Each color represents a different chromosome.

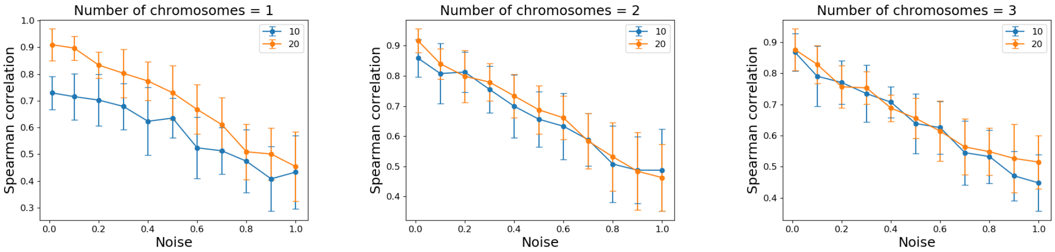


(a) (b) (c)

Figure 4-5: Performance of our method in the noiseless setting. Root-mean-square deviation (RMSD) between true and reconstructed structure computed with and without tensor constraints. Simulated data was generated using a Brownian motion model with (a) one (b) two and (c) three chromosomes. Mean and standard deviation over 20 trials are shown.

$D_{i_1 i_2 \ldots i_k} = D^*_{i_1 i_2 \ldots i_k}(1+\delta)$ by sampling δ uniformly within $(-\epsilon, \epsilon)$ as in (Zhang et al., 2013), where ϵ is a given noise level. For our simulations we sample a maximum of 1000 three-way distance constraints. As shown in Figure C-3, we observe that the number of constraints does not have a major effect on the reconstruction accuracy. While for all simulations shown in this section, we set $\rho = 0.000001$, Figure C-4 in shows that the performance is not significantly different when using different choices of ρ.

In Figure 4-6 we numerically assess the accuracy of our predicted structure for the Brownian motion model for different number of chromosomes (one, two, or three) and



(a) (b) (c)

Figure 4-6: Performance of our method in the noisy setting. Spearman correlation under different noise levels for (a) one, (b) two and (c) three chromosomes. Simulated data was generated using a Brownian motion model where each chromosome has 10 or 20 domains. Mean and standard deviation over 20 trials are shown.

different number of domains per chromosome (10 or 20) by computing the Spearman correlation between reconstructed and true pairwise distances, similar to (Zhang et al., 2013). As expected, Figure 4-6 shows that when the noise level increases, then the Spearman correlation between the original and reconstructed configuration decreases. For the simulations with one chromosome, the Spearman correlation is higher for 20 domains than 10.

Application to 3D diploid genome reconstruction

We apply our algorithm to the problem of reconstructing the diploid genome from contact frequency data derived from experiments. We obtain pairwise and three-way contact frequencies collected via SPRITE in human lymphoblastoid cells from (Quinodoz et al., 2018). Since we aim to reconstruct the whole diploid genome, which consists of approximately 6 billion base pairs, for computational reasons we bin the contact frequencies in the SPRITE dataset into 10 Mega-base pair (Mb) regions. While some previous studies considered higher resolutions, the majority of the studies (Cauer et al., 2019; Hu et al., 2013; Rousseau et al., 2011; Varoquaux et al., 2014; Zhang et al., 2013) did not attempt to reconstruct the whole diploid genome and focused only on reconstructing one chromosome, thus enabling them to consider higher resolutions.

After filtering out regions with a small number of total contacts, we obtain 514 unphased points on the chromosomes. We convert the pairwise contact frequencies to pairwise distances using the previously observed relationship $D_{ij} = F_{ij}^{-1/2}$ (Rousseau et al., 2011) and use Lemma 4.3.3 to obtain the distances between homolog pairs from this data. As in our simulations in the noisy setting, we randomly sample 1000 three-way distance constraints from all nonzero three-way contact frequencies (for the transformation from three-way contact frequencies to three-way distances, see Section 4.5). Finally, we obtain the distances between neighboring 10Mb beads by empirically evaluating 3D reconstructions under different input distances; see Appendix C and Figures C-5, C-6.

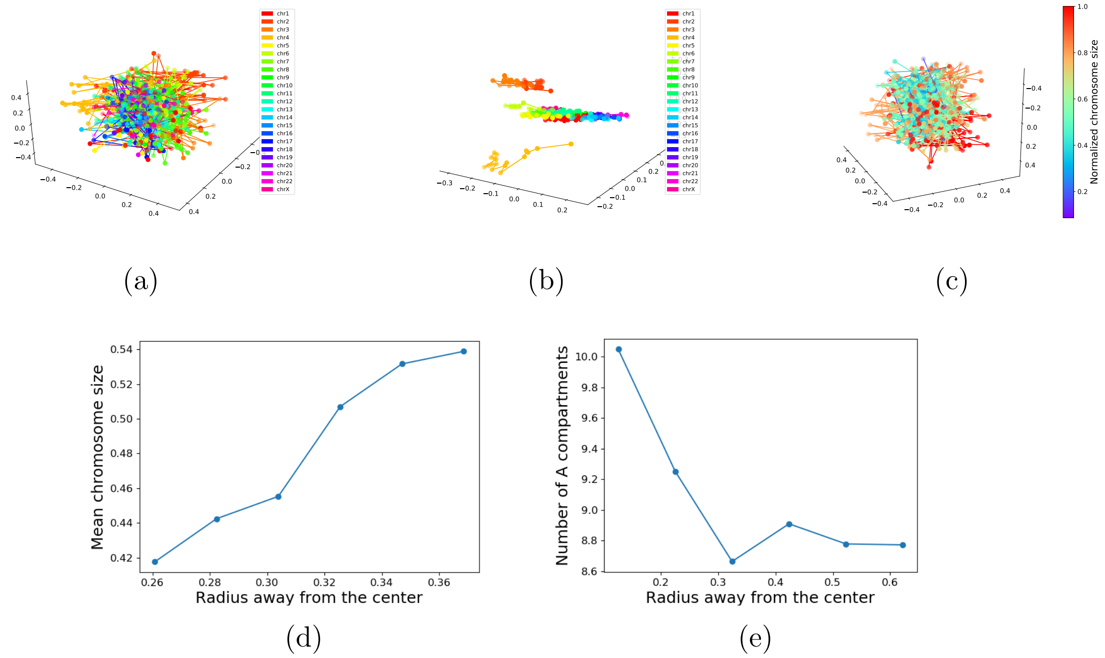
After obtaining the pairwise constraints, homolog-homolog constraints, neighboring bead constraints, and three-way distance constraints, we solve the SDP problem in Equation (4.8) for the noisy setting and analyze the corresponding 3D coordinates. Our diploid reconstruction is shown in Figure 4-7a. We compare this diploid genome reconstruction to the 3D structure computed via ChromSDE (Zhang et al., 2013), shown in Figure 4-7b obtained under the assumption that the observed contact frequencies and the corresponding distances are a sum of four equal quantities, i.e., $\|x_i - x_j\|^2$, $\|x_i - y_j\|^2$, $\|y_i - x_j\|^2$, and $\|y_i - y_j\|^2$ are equal. In Figure C-7, we show that the reconstruction obtained using ChromSDE with equal distances does not recapitulate known biology as described in the following paragraphs.

Experimental studies have shown that chromosomes are organized by size within the nucleus, with small chromosomes in the interior and larger chromosomes on the periphery (Bolzer et al., 2005). We colored each chromosome according to its size and computed the mean chromosome size versus distance away from the center. The results of the 3D configuration obtained using our method are shown in Figure 4-7c,d and recapitulate prior studies: smaller chromosomes are preferentially located in the center, whereas larger chromosomes are preferentially on the periphery; see also Figure C-8. This is especially apparent for chromosomes 2 and 4, which are some of the largest chromosomes, and in our reconstruction they are located on the periphery as expected.

Experimental studies on the spatial organization of the genome have also shown that the center of the nucleus is enriched in active compartments (known as A compartments), while the periphery contains inactive compartments (known as B compartments) (Stevens et al., 2017). From previously published data on the location of A and B compartments in human lymphoblastoid cells (Rao et al., 2014), we counted the number of A compartments per 10Mb bin. Then dividing our 3D reconstruction into concentric circles of increasing radius away from the center, we found the mean number of A compartments in each concentric circle. Figure 4-7e shows that with increasing distance away from the center, the number of A compartments decreases. Thus, our reconstruction recovers the experimentally observed trend for A compartments to be preferentially located near to the nucleus center. As shown in Figures C-9 and C-10, we note that our results are robust to the choice of the tuning parameter ρ resulting in biologically plausible configurations independent of the choice of ρ. Currently, many studies such as (Rieber and Mahony, 2017) simply ignore the fact that the genome is diploid and infer the 3D genome organization as if the data was collected from a haploid organism, assuming that the homologous loci have the same 3D structure. However, we show in Figure C-11 that the haploid distance matrices, computed by including only one copy of each of the homologous loci, are different between the two copies with a mean Spearman correlation of only 0.08. This shows that modeling the diploid aspect of the genome provides valuable information regarding the 3D structure of each of the homologs, which may be substantially different.

4.8 Discussion

In this study, we proved that for diploid organism the 3D genome structure is not identifiable from pairwise distance measurements alone. This implies that applying any algorithm for the reconstruction of the 3D genome structure from typical chromosome conformation capture data for a diploid organism can result in any of the infinitely many configurations with the same pairwise contact frequencies. We showed that unique idenfiability is obtained using distance constraints between neighboring genomic loci as well as three-way distance constraints in addition to the pairwise



(a) (b) (c)

(d) (e)

Figure 4-7: 3D diploid genome reconstruction. Estimated 3D positions of all chromosomes and their corresponding homologs at 10Mb resolution. 3D positions obtained using (a) our method and (b) using ChromSDE with chromosomes colored according to chromosome number. (c) Whole diploid organization obtained via our method, colored by chromosome size. (d) Mean chromosome size as the distance from the center increases. (e) The number of A compartments as the distance from the center increases.

distance constraints that can be obtained from typical chromosome capture data. Distances between neighboring genomic loci can be obtained from imaging studies or empirically, while three-way distance constraints can be obtained from the most recently developed sequencing-based methods for obtaining contact frequencies such SPRITE (Quinodoz et al., 2018), C-walks (Olivares-Chauvet et al., 2016) and GAM (Beagrie et al., 2017). We also presented SDP formulations for determining the 3D genome reconstruction both in the noiseless and noisy setting. Finally, we applied our algorithm to contact frequency data from human lymphoblastoid cells collected using SPRITE and showed that our results recapitulate known biological trends; in particular, in the 3D configuration identified using our method, the small chromosomes are preferentially situated in the interior of the cell nucleus, while larger chromosomes are preferentially situated at the periphery of cell nucleus. In addition, in the 3D configuration identified using our method the number of A domains is higher in the interior versus the periphery, which is in line with experimental results. Our work shows the importance of higher-order contact frequencies that can be measured using SPRITE (Quinodoz et al., 2018), C-walks (Olivares-Chauvet et al., 2016) and GAM (Beagrie et al., 2017) for obtaining the 3D organization of the genome in diploid organisms. This is particularly relevant for the reconstruction of cancer genomes, where copy number variations are frequent and hence the genome may contain even more than two copies of each locus.

4.9 Future directions

We conjecture that identifiability of the 3D genome structure can also be achieved by replacing the higher-order contact constraints by distance constraints to the center of the cell nucleus. Such constraints are also biologically relevant, since these distances can be measured via imaging experiments, or inferred by measuring whether a particular locus is in a lamin-associated domain or a telomere, both of which tend to lie at the boundary of the cell nucleus (Crabbe et al., 2012; Guelen et al., 2008b; Van Steensel and Belmont, 2017). Another future research direction is the development of specialized solvers to enable reconstruction of the genome at higher resolution. In this study we used a 10Mbp resolution due to the computational constraints imposed by SDP solvers. Furthermore, the theory in this study builds on the assumption that distances are inverses of square roots of pairwise and higher-order contact frequencies. Finally, another interesting future research direction is to develop a method for estimating the map between higher-order contact frequencies and distances, and then prove identifiability as well as build reconstruction algorithms for these different maps.

Chapter 5

Learning Causal Differences between Gene Regulatory Networks

Parts of this chapter were published as:
Wang, Y., Squires, C., Belyaeva, A., & Uhler, C. (2018). Direct estimation of differences in causal graphs. In Advances in Neural Information Processing Systems (pp. 3770-3781).
My contributions to that manuscript were to evaluate the methodology and contribute to writing the manuscript. In this chapter, we omit the proofs showing the consistency of the proposed algorithm but they can be found in the above publication (Wang et al., 2018b).
The majority of this chapter is part of a manuscript under review:
Belyaeva, A., Squires, C., & Uhler, C. (2020). DCI: Learning Causal Differences between Gene Regulatory Networks.
My contributions were to create a package for the method, design and perform method and data analysis, and write the manuscript.

5.1 Summary

Designing interventions to control gene regulation necessitates modeling a gene regulatory network by a causal graph. Currently, large-scale expression datasets from

different conditions, cell types, disease states and developmental time points are being collected. However, application of classical causal inference algorithms to infer gene regulatory networks based on such data is still challenging, requiring high sample sizes and computational resources. Here, we propose an algorithm that efficiently learns the differences in gene regulatory mechanisms between different conditions. Our difference causal inference (DCI) algorithm infers changes (i.e., edges that appeared, disappeared or changed weight) between two causal graphs given gene expression data from the two conditions. This algorithm is efficient in its use of samples and computation since it infers the differences between causal graphs directly without estimating each possibly large causal graph separately. We provide a user-friendly Python implementation of DCI and also enable the user to learn the most robust difference causal graph across different tuning parameters via stability selection. Finally, we evaluate DCI on synthetic data and show how to apply DCI to bulk and single-cell RNA-seq data from different conditions and cell states, and we also validate our algorithm by predicting the effects of interventions.

5.2 Introduction

Biological processes from differentiation to disease progression are governed by gene regulatory networks. Various methods have been developed for inferring such networks from gene expression data (Wang and Huang, 2014), the majority by learning *undirected* graphs using correlations (Langfelder and Horvath, 2008), Gaussian graphical models to capture partial correlations (Friedman et al., 2008), or mutual information (Reshef et al., 2011). However, the ultimate goal is often to use gene regulatory networks to predict the effect of an intervention (small molecule, overexpression of a transcription factor, knock-out of a gene, etc.). This cannot be done using an undirected graph and necessitates modeling a gene regulatory network by a *causal* (*directed*) graph.

Causal relationships are commonly represented by a directed acyclic graph (DAG) and a variety of methods have been developed for learning causal graphs from observational data (Glymour et al., 2019). These methods have been successfully applied to learning (directed) gene regulatory networks on a small number of genes, starting with the pioneering study by (Friedman et al., 2000). However, applying these methods at the whole genome-level is still challenging due to high sample size and computational requirements of the algorithms.

We address this problem by noting that it is often of interest to learn *changes* in causal (regulatory) relationships between two related gene regulatory networks corresponding to different conditions, disease states, cell types or developmental time points, as opposed to learning the full gene regulatory network for each condition. This can reduce the high sample and computational requirements of current causal inference algorithms, since while the full regulatory network is often large and dense, the difference between two related regulatory networks is often small and sparse. As of now, this problem has only been addressed in the undirected setting, namely by KLIEP (Liu et al., 2017), DPM (Zhao et al., 2014) and others (Fukushima, 2013; Lichtblau et al., 2017) that estimate differences between undirected graphs; for a recent review see (Shojaie, 2020). In the following, we describe the *difference causal inference* (*DCI*) algorithm and present an easy to use Python package for the direct estimation of the difference between two causal graphs based on observational data from two conditions (for the theoretical properties of this algorithm see (Wang et al., 2018b)). In particular, we show how to apply DCI to gene expression data from different conditions and demonstrate the algorithm's performance on synthetic data and real data. Importantly, our DCI implementation also allows selecting the most robust difference gene regulatory network based on a collection of tuning parameters via stability selection. To seamlessly integrate DCI with other causal inference methods, it is incorporated in the `causaldag` package at `https://github.com/uhlerlab/causaldag`.

5.3 Preliminaries

Let $\mathcal{G}^{(k)} = ([p], E^{(k)})$ for $k \in \{1, 2\}$ be a directed acyclic graph (DAG) with nodes $[p] := \{1, \ldots, p\}$ and directed edges $E^{(k)}$. The DAGs $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ model the gene regulatory networks in the two conditions of interest. We assume that the two DAGs are consistent with the same ordering, meaning that there cannot be an edge $i \to j$ in $\mathcal{G}^{(1)}$ if there is a directed path $j \to \cdots \to i$ in $\mathcal{G}^{(2)}$ and vice-versa. This assumption is reasonable in gene regulatory networks, since genetic interactions may appear or disappear or change edge weights, but generally do not change directions. For each graph we associate a random variable $X_i^{(k)}$ to each node $i \in [p]$. We consider the setting where we have data from two conditions and this data is generated by a linear structural equation model

$$X^{(k)} = B^{(k)T} X^{(k)} + \epsilon^{(k)} \qquad \text{for} \;\; k \in \{1, 2\}, \tag{5.1}$$

where $X = (X_1, \cdots, X_p)^T$ is a random vector, $B^{(k)}$ denotes the weighted adjacency matrix of the DAG $\mathcal{G}^{(k)}$ and $\epsilon^{(k)} \sim \mathcal{N}(0, \Omega^{(k)})$ denotes Gaussian noise with covariance matrix $\Omega^{(k)} := \mathrm{diag}(\sigma_1^{2(k)}, \cdots, \sigma_p^{2(k)})$. Given samples $\hat{X}^{(1)} \in \mathbb{R}^{n_1 \times p}$ and $\hat{X}^{(2)} \in \mathbb{R}^{n_2 \times p}$ from the two models (where n_1 and n_2 denote the sample size under each condition), our goal is to estimate the difference-DAG across the two conditions. The difference-DAG is denoted by $\Delta = ([p], E)$ and contains an edge $i \to j \in E$ if and only if $B_{ij}^{(1)} \neq B_{ij}^{(2)}$.

5.4 Difference Causal Inference (DCI) algorithm

DCI takes as input two matrices $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ of size $n_1 \times p$ and $n_2 \times p$, where n_1, n_2 are the number of samples in each dataset and p is the number of genes. These matrices contain the RNA-seq values corresponding to two different conditions. DCI outputs the difference causal graph between the two conditions, i.e. the edges in the gene regulatory networks that appeared, disappeared or changed weight between the two conditions (Fig. 5-1). DCI consists of three steps described below (also described in Algorithm 1) for computing the difference-DAG. These steps are implemented in the `dci` function of the `causaldag` package found at `https://github.com/uhlerlab/causaldag`. Briefly, DCI is initialized with the difference undirected graph,

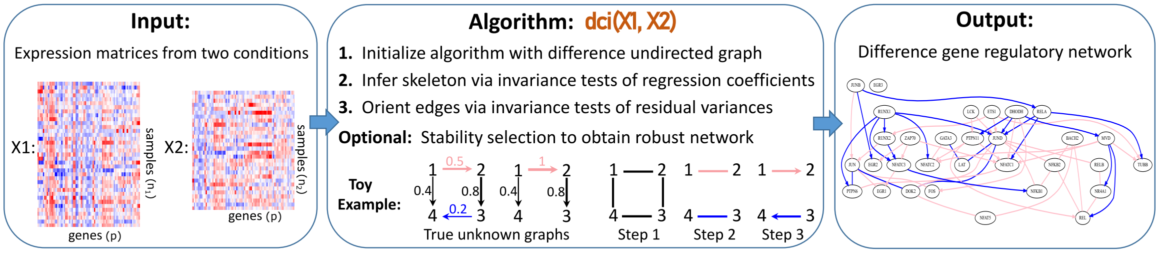

Input:
Expression matrices from two conditions
X1:
samples (n_1)
genes (p)
X2:
samples (n_2)
genes (p)
Algorithm: dci(X1, X2)
1. Initialize algorithm with difference undirected graph
2. Infer skeleton via invariance tests of regression coefficients
3. Orient edges via invariance tests of residual variances
Optional: Stability selection to obtain robust network
Toy Example:
0.5
0.4
0.8
0.2
1
True unknown graphs
Step 1
Step 2
Step 3
Output:
Difference gene regulatory network

Figure 5-1: Overview of DCI algorithm: DCI takes as input two gene expression matrices $X1$ and $X2$, representing two different conditions of interest. The function `dci(X1,X2)` outputs the difference gene regulatory network consisting of the causal relationships that appeared, disappeared or changed weight between the two conditions.

```
Algorithm 1 Difference Causal Inference (DCI) algorithm (dci function)
  Input: Sample data $\hat{X}^{(1)}$, $\hat{X}^{(2)}$.
  Output: Estimated difference-DAG $\hat{\Delta}$.
  Initialize with difference undirected graph $\bar{\Delta}$ and conditioning set $\mathcal{C}$.
  Estimate the skeleton of the difference-DAG $\tilde{\Delta}$ using Algorithm 2.
  Direct edges in $\tilde{\Delta}$ using Algorithm 3 to obtain $\hat{\Delta}$.
```

which can be obtained using prior methods, our constraint-based method, based on prior knowledge or by simply taking the complete graph. Second, the skeleton of the difference causal graph is determined by testing for invariance of regression coefficients estimated from data. Finally, edges are oriented by testing for invariance of residual variances, also estimated from the data.

Step 1: Initialization with a difference undirected graph. In the first step, the algorithm is initialized with a difference undirected graph (representing changes of conditional dependencies among genes between the two conditions), which we denote by $\bar{\Delta}$, with edge $i - j$ if and only if $\Theta^{(1)}_{ij} \neq \Theta^{(2)}_{ij}$ for $i \neq j$, where $\Theta^{(1)}$ and $\Theta^{(2)}$ are the precision matrices corresponding to the DAGs $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. This is done to remove some edges to reduce the downstream computational burden. The difference undirected graph can be determined either using our constraint-based method outlined below, previous methods such as KLIEP (Liu et al., 2014, 2017; Zhao et al., 2014; Fukushima, 2013; Lichtblau et al., 2017), based on prior biological knowledge, or simply with the complete graph when the number of considered genes is small. In addition, to reduce the number of downstream hypothesis tests, the nodes to be considered as conditioning sets can be reduced to the nodes in the difference undirected graph as well as nodes whose conditional distribution changes between the two conditions, namely $\mathcal{C} = \left\{ i \mid \exists j \in [p] \text{ such that } \Theta^{(1)}_{i,j} \neq \Theta^{(2)}_{i,j} \right\}$. The reduced node set can be determined from the output of methods such as our constraint-based method, KLIEP (Liu et al., 2014, 2017; Zhao et al., 2014; Fukushima, 2013; Lichtblau et al., 2017), prior biological knowledge, or taken as the set of all nodes when the number of genes to be considered is small.

Our constraint-based method estimates the precision matrices corresponding to each dataset, $\hat{\Theta}^{(1)}$ and $\hat{\Theta}^{(2)}$, from data and computes the following test statistic for each entry (i, j) to quantify the difference:

$$\hat{Q_{ij}} := \left(\hat{\Theta}^{(1)}_{ij} - \hat{\Theta}^{(2)}_{ij}\right)^2 \cdot \left(\frac{\hat{\Theta}^{(1)}_{ii}\hat{\Theta}^{(1)}_{jj} + (\hat{\Theta}^{(1)}_{ij})^2}{n_1} + \frac{\hat{\Theta}^{(2)}_{ii}\hat{\Theta}^{(2)}_{jj} + (\hat{\Theta}^{(2)}_{ij})^2}{n_2}\right)^{-1}.$$

In order to determine whether a particular edge should remain as part of the undirected difference graph, $\hat{Q_{ij}}$ is tested for fit to the F-distribution with parameters $F(1, n_1 + n_2 - 2p + 2)$ and the edge remains if the null hypothesis is rejected. As described in (Wang et al., 2018b), this hypothesis testing framework comes from the facts that (1) the entry $\hat{\Theta}_{ij}$ converges asymptotically to a multivariate normal centered at the true parameter, with variance $\Theta_{ii}\Theta_{jj} + \Theta^2_{ij}$, (2) the difference between two independent standard normal random variables follows a χ^2 distribution, and (3) the F-distribution with the suggested parameters converges asymptotically to a χ^2 distribution, but the fatter tails of the F-distribution better match the finite sample distribution of the test statistic (Lütkepohl, 2005).

Step 2: Estimation of the skeleton of the difference causal graph. In the second step, edges are removed from the difference undirected graph by testing for invariance of regression coefficients using an F-test. Since each entry B_{ij} corresponds to a regression coefficient $\beta_{ij|S}$ obtained by regressing X_j on X_i given the parents of node j in $\mathcal{G}$, testing whether $B^{(1)}_{ij} = B^{(2)}_{ij}$, is equivalent to testing whether there exists a set of nodes S such that $\beta^{(1)}_{i,j|S} = \beta^{(2)}_{i,j|S}$. More precisely, Algorithm 2 describes the estimation of the skeleton of the difference-DAG, denoted by $\tilde{\Delta}$. Given $i, j \in [p]$

Algorithm 2 Estimating skeleton of the difference-DAG (`dci_skeleton` function)

```
**Input:** Sample data $\hat{X}^{(1)}$, $\hat{X}^{(2)}$, estimated difference undirected graph $\bar{\Delta}$ and conditioning set $\mathcal{C}$, maximum conditioning set size $r$.
**Output:** Estimated skeleton $\tilde{\Delta}$.
Set $\tilde{\Delta} := \bar{\Delta}$;
**for** each edge $i - j$ in $\tilde{\Delta}$ **do**
    If $\exists S \subseteq \mathcal{C} \setminus \{i, j\}$, with $|S| \leq r$, such that $\beta^{(k)}_{i,j|S}$ is invariant across $k = \{1, 2\}$, delete $i - j$ in $\tilde{\Delta}$ and continue to the next edge. Otherwise, continue.
**end for**
```

and $S \subseteq [p] \setminus \{i, j\}$, the regression coefficient $\beta^{(k)}_{i,j|S}$ is defined as the entry in $\beta^{(k)}_M$ corresponding to i, where $\beta^{(k)}_M$ is the best linear predictor of $X^{(k)}_j$ given $X^{(k)}_M$, i.e., the minimizer of $\mathbb{E}[(X^{(k)}_j - (\beta^{(k)}_M)^T X^{(k)}_M)^2]$ and $M := \{i\} \cup S$. Hence, $\beta^{(k)}_{i,j|S}$ can be computed in closed form. Note that $B^{(k)}_{ij}$ corresponds to a particular regression coefficient, namely when $S = \mathrm{Pa}^{(k)}(j) \setminus \{i\}$, where $\mathrm{Pa}^{(k)}(j)$ denotes the parents of node j in $\mathcal{G}^{(k)}$. This means that we can determine whether $B^{(1)}_{ij} = B^{(2)}_{ij}$ without learning each graph $\mathcal{G}^{(k)}$, namely by testing subsets S: if there exists a subset S such that $\beta^{(1)}_{i,j|S} = \beta^{(2)}_{i,j|S}$, then $B^{(1)}_{ij} = B^{(2)}_{ij}$ and hence the edge $(i, j) \notin \tilde{\Delta}$. In fact, it turns out that it is sufficient to consider conditioning sets $S \subseteq \mathcal{C}$ (Wang et al., 2018b).

Step 3: Orienting edges in the difference causal graph. All edge directions that are identifiable from observational data are obtained by testing for invariance of residual variances. More precisely, we direct edges in the skeleton of the difference-DAG $\tilde{\Delta}$ using Algorithm 3. Similar to many prominent causal inference algorithms such as the PC (Spirtes et al., 2000) and GES (Meek, 1997) algorithms, we may not be able to determine the directions of all edges in $\tilde{\Delta}$, since in general, the difference-DAG Δ is not completely identifiable. In fact, we are able to identify the direction of all edges adjacent to nodes whose internal node variances are unchanged across the two conditions, i.e. for which $\sigma^{(1)}_i = \sigma^{(2)}_i$ (Wang et al., 2018b). Hence the output of the DCI algorithm is a partially directed acyclic graph, which contains both directed and undirected edges. Edge directions in the difference-DAG are determined by calculating residual variances $(\sigma^{(k)}_{j|S})^2$ and testing whether they are invariant, i.e. whether $(\sigma^{(1)}_{j|S})^2 = (\sigma^{(2)}_{j|S})^2$, again using an F-test. Given $j \in [p]$ and $S \subseteq [p] \setminus \{j\}$, the residual

Algorithm 3 Directing edges in the difference-DAG (`dci_orient` function)

```
Input: Sample data $\hat{X}^{(1)}$, $\hat{X}^{(2)}$, estimated skeleton $\tilde{\Delta}$ and conditioning set $\mathcal{C}$, maximum conditioning set size $r$.
Output: Estimated difference-DAG $\hat{\Delta}$.
Set $\hat{\Delta} := \emptyset$;
for conditioning set size $k = 1, \ldots, r$ do
  Set $\mathcal{V}$ to all nodes $j$ incident to at least one undirected edge in $\tilde{\Delta}$
  For each $j \in \mathcal{V}$, let $p_j = \max_{S \subseteq \mathcal{C} \setminus \{j\}: |S| = k} \mathrm{pval}(\sigma^{(1)}_{j|S} = \sigma^{(2)}_{j|S})$
  while $\mathcal{V} \neq \emptyset$ do
    Let $j = \arg\max_{j' \in \mathcal{V}} p_j$
    If $p_j > \alpha$, set the corresponding $S$ as the parent set for $j$ in $\hat{\Delta}$, and the remaining adjacent nodes to $j$ as its children in $\hat{\Delta}$, as long as this does not create any cycles or contradict any existing edges.
    Let $\mathcal{V} = \mathcal{V} \setminus \{j\}$
  end while
end for
Orient as many undirected edges as possible via graph traversal using the following rule:
    Orient $i - j$ as $i \to j$ whenever there is a chain $i \to \ell_1 \to \cdots \to \ell_t \to j$.
```

variance $(\sigma^{(k)}_{j|S})^2$ is defined as the variance of the regression residual when regressing $X^{(k)}_j$ onto the random vector $X^{(k)}_S$. In fact it holds that $\sigma^{(1)}_i = \sigma^{(2)}_i$ if and only if there exists a subset $S \subseteq \mathcal{C} \setminus \{i\}$ such that $\sigma^{(1)}_{i|S} = \sigma^{(2)}_{i|S}$ and if $i \to j$ in Δ then $j \notin S$, whereas if $j \to i$ in Δ then $j \in S$ (Wang et al., 2018b). Hence determining conditioning sets that lead to the invariance of residual variances can be used to orient some of the edges in the difference-DAG Δ. In order to ensure that the results are not dependent on the order in which one iterates over nodes, Algorithm 3 simultaneously considers all nodes at each level of conditioning set size. In Algorithm 3 we note that $\mathrm{pval}(\sigma^{(1)}_{j|S} = \sigma^{(2)}_{j|S})$ refers to the p-value obtained from the F-test to determine the invariance of residual variances.

Example. We end the description of the main three steps of the DCI algorithm with an example, shown in Fig. 5-2. Suppose that $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ are two true but unknown gene regulatory networks governing genes 1, 2, and 3 corresponding to two different biological conditions. Let $B^{(1)}$ and $B^{(2)}$ be the autoregressive matrices defined by the edge weights of $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ and let $\epsilon^{(k)} \sim \mathcal{N}(0,1)$. Given enough

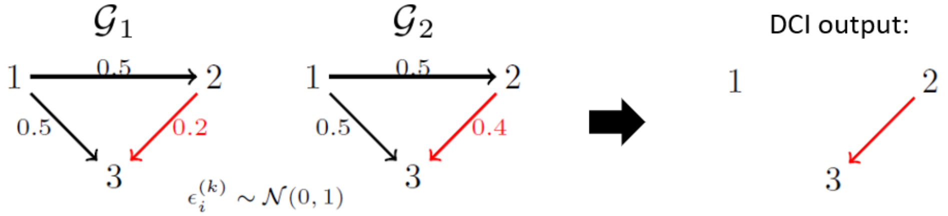


Figure 5-2: Example of the DCI algorithm for 3 nodes.

samples from the model, we expect that DCI will find the true difference between $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$, i.e. that there is an edge $2 \rightarrow 3$ with a changed weight. First, we can initialize DCI with the complete graph as the difference undirected graph and let the conditioning set $\mathcal{C}$ be the set of all nodes. Second, the skeleton of the difference causal graph is determined. For the 3-node example in Fig. 5-2, for the changed edge $2-3$ observe that given enough samples the regression coefficients would be estimated as $\beta^{(1)}_{2,3|\{1\}} = 0.2$ and $\beta^{(2)}_{2,3|\{1\}} = 0.4$, so $\beta^{(1)}_{2,3|\{1\}} \neq \beta^{(2)}_{2,3|\{1\}}$ and in fact, for all sets of nodes $S \subseteq \mathcal{C}$, $\beta^{(1)}_{2,3|S} \neq \beta^{(2)}_{2,3|S}$. While, for example, for unchanged edge $1-3$, there exists a set $S = \{2\}$ such that $\beta^{(1)}_{1,3|\{2\}} = \beta^{(2)}_{1,3|\{2\}} = 0.5$. This observation motivates testing invariance of regression coefficients to infer the skeleton of the difference graph and deleting edges with invariant regression coefficients. Third, the edges in the skeleton of the difference causal graph are oriented. In the 3-node example in Fig. 5-2, $2 \rightarrow 3$ is the correct orientation for the edge $2-3$. Observe that the residual variances such as $\sigma^{(1)}_{3|\{1,2\}} = \sigma^{(2)}_{3|\{1,2\}} = 1$ are invariant when node 2 (a parent of node 3) $\in S$, while $\sigma^{(1)}_{3|S} \neq \sigma^{(2)}_{3|S}$ when $2 \notin S$, which motivates using invariance between residual variances to determine edge orientations.

Stability selection to obtain robust difference gene regulatory network. Running DCI requires choosing several hyperparameters, namely the ℓ_1-regularizer for estimating the difference undirected graph via KLIEP (Liu et al., 2017) or a significance level for the constraint-based method as well as the significance levels for hypothesis testing of invariance of regression coefficients as well as residual variances. We implemented DCI with stability selection to address the issue of choosing the correct hyperparameters. Stability selection was introduced by (Meinshausen and Bühlmann, 2010) and has been successfully applied in tandem with other causal

Algorithm 4 DCI with stability selection (`dci_stability_selection` function)

```
Input: Sample data $\hat{X}^{(1)}$, $\hat{X}^{(2)}$, set of tuning parameters $\Lambda$, number of subsamples $N$ of size given by the fraction $f$ of all samples, and threshold $\pi_{\text{thr}}$ for choosing stable variables.
Output: Stable estimate of difference-DAG $\hat{\Delta}^{\text{stable}}$.
for each $\lambda$ in $\Lambda$ do
  for each $i$ in $1, \dots, N$ do
    Generate subsamples of the two datasets, $\hat{X}^{(1)}_{(i)}$ and $\hat{X}^{(2)}_{(i)}$ (without replacement) of size defined by the fraction $f$ of the full samples size.
    Run Algorithm 1 on $\hat{X}^{(1)}_{(i)}$ and $\hat{X}^{(2)}_{(i)}$ with hyperparameters $\lambda$ to obtain $\hat{\Delta}^{\lambda}_{(i)}$.
  end for
  Calculate selection probability for each edge $k$ by $\hat{\Pi}^{\lambda}_{k} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\{k \in \hat{\Delta}^{\lambda}_{(i)}\}$.
end for
Construct stable estimate of difference-DAG $\hat{\Delta}^{\text{stable}} = \{k : \max_{\lambda \in \Lambda} \hat{\Pi}^{\lambda}_{k} \geq \pi_{\text{thr}}\}$.
```

inference methods (Meinshausen et al., 2016). The idea behind stability selection is to choose the most stable estimate across different hyperparameters as opposed to focusing on choosing the right value for the hyperparameters.

Algorithm 4 outlines the methodology for running DCI with stability selection. Let Λ denote the set of considered hyperparameter values consisting of ℓ_1 regularizers for KLIEP or significance levels for the constraint-based method, significance levels for hypothesis testing of invariance of regression coefficients and significance levels for hypothesis testing of invariance of residual variances. Given a particular $\lambda \in \Lambda$, we can run DCI (Algorithm 1) and obtain the corresponding estimated difference causal graph $\hat{\Delta}^{\lambda}$. Stability selection relies on estimating the probability of selection of each edge $\hat{\Pi}^{\lambda}_{k}$ by running the DCI algorithm on subsamples of the data. Aggregating selection probabilities across different tuning parameters $\lambda \in \Lambda$, we keep edges with high selection probability as the stable set of estimated edges in the difference-DAG $\hat{\Delta}^{\text{stable}}$.

Stability selection alleviates the need for the user to rely on the results from a single hyperparameter in DCI. Instead, the user can see how the selection probability of edges vary in comparison to each other across a range of hyperparameters. Such a plot is shown in Figure 5-3, where the significance level for hypothesis tests in the difference skeleton discovery phase is varied from $\alpha = 10^{-5}$ to $\alpha = 0.1$, and the probability that each edge is included in the difference skeleton is given on the vertical axis. Additionally, for each edge, we may use a heatmap to indicate the probability that it appears in either one or the other orientation in the difference-DAG. This provides additional information on how much the user should trust the presence of an edge: a true edge should consistently be oriented in its correct orientation, and thus inconsistent orientation of an edge across different subsamples of data indicates that it is a false positive. In Figure 5-3, true positive edges are colored according to the probability that they appear in their correct orientation, while false positive edges are colored according to the proportion of times that they occur in an arbitrary fixed direction. It is apparent from Figure 5-3 that the three edges (a-c) are significantly more likely than the others to belong to the difference skeleton, and that all of these edges have a consistent orientation, which suggests that these are true positives, and indeed they are. Meanwhile, the fourth and fifth most likely edges (d, e) have inconsistent orientations, and they indeed do not correspond to edges in the true DAG, while the sixth most likely edge (f) has a more consistent orientation and is indeed a true positive. This information allows the user to select a trade-off between false positives and true positives that is suitable for their application, e.g. a conservative rule which excludes any edges which themselves have inconsistent orientations or lay below an edge with inconsistent orientations would still get three of the four true positives correct, while more liberal rules would pick up all four true positives while only returning a small number of false positives.

DCI with stability selection is optimized to run in parallel on multiple cores across the different bootstrap subsamples. In addition, since the DCI skeleton learning phase has a monotonicity property, i.e. if an edge is absent in the difference skeleton for some α, then it is absent in the difference skeleton for all $\alpha' < \alpha$, the DCI skeleton discovery phase is run simultaneously for all significance levels to speed up computation.

An issue that can arise in practice when applying the original DCI method to gene expression data is due to the need to compute test statistics that depend on the inverse of the sample covariance matrix (Wang et al., 2018b). This inverse may not exist, since gene expression data is often high-dimensional with more genes than samples, in particular when the data is subsampled for stability selection, and the matrix can have many zeros (possibly leading to variance zero for a node) due to dropout. In this case, we use the pseudoinverse instead of the inverse to compute the test statistics.

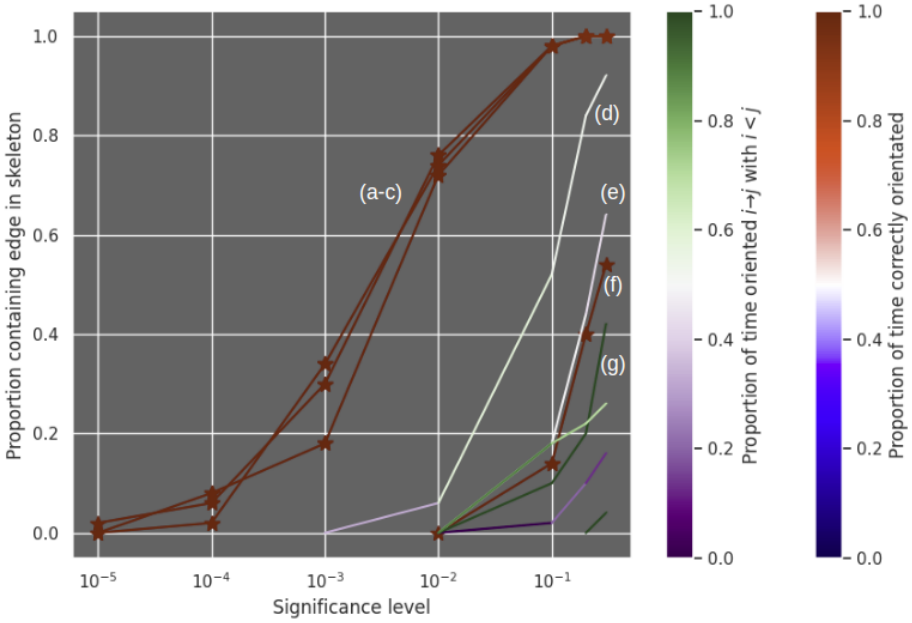


Figure 5-3: Stability selection paths for the difference between two simulated DAGs, with $p = 40$ nodes, $n = 2,000$ samples, and 2 additions/deletions between either DAG, where the skeleton of one of the DAGs was generated from an Erdös-Renyi model with 2 expected neighbors per node. Each curve represents an edge in the difference skeleton. The horizontal axis displays the significance level used for hypothesis tests in the difference skeleton discovery phase of the method, and the vertical axis displays the proportion of bootstrap iterations (out of 50) for which an edge was picked to be present. For the 4 edges which belong to the true difference skeleton (curves a-c and f), their color (varying from blue to red) indicates the probability that the edge was oriented in the correct direction, given that it was included in the graph. For the edges which do not belong to the true difference skeleton (such as curves d, e and g), their color (varying from purple to green) indicates the probability that the edge was oriented such that the node with the smaller index pointed to the larger one.

5.5 Evaluation

In this section, we compare the performance of DCI to the naive approach of running classical causal inference algorithms such as PC (Spirtes et al., 2000) or GES (Meek, 1997) on each dataset $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ separately, obtaining two causal graphs and then taking the difference. We evaluate DCI on both synthetic and real data.

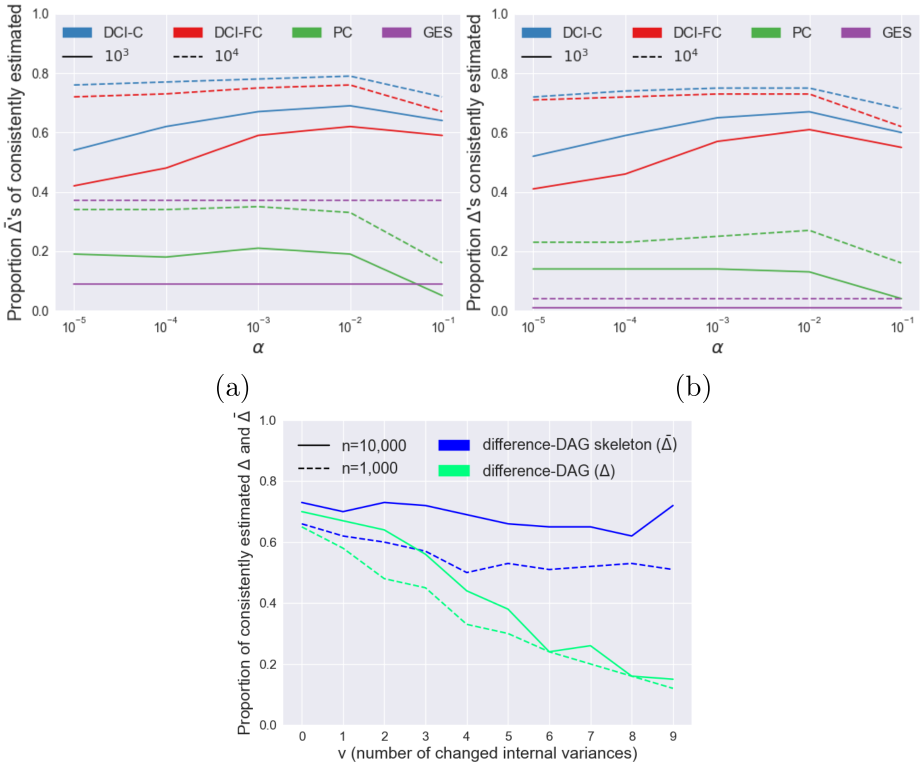
Synthetic data

We analyze the performance of our algorithm in both, the low- and high-dimensional setting. For both settings we generated 100 realizations of pairs of upper-triangular SEMs $(B^{(1)}, \epsilon^{(1)})$ and $(B^{(2)}, \epsilon^{(2)})$. For $B^{(1)}$, the graphical structure was generated using an Erdös-Renyi model with expected neighbourhood size s, on p nodes and n samples. The edge weights were uniformly drawn from $[-1, -0.25] \cup [0.25, 1]$ to ensure that they were bounded away from zero. $B^{(2)}$ was then generated from $B^{(1)}$ by adding and removing edges with probability 0.1, i.e.,

$$\begin{aligned} B_{ij}^{(2)} &\overset{\text{i.i.d.}}{\sim} \text{Ber}(0.9) \cdot B_{ij}^{(1)} \text{ if } B_{ij}^{(1)} \neq 0, \\ B_{ij}^{(2)} &\overset{\text{i.i.d.}}{\sim} \text{Ber}(0.1) \cdot \text{Unif}([-1, -.25] \cup [.25, 1]) \text{ if } B_{ij}^{(1)} = 0. \end{aligned}$$

Note that while the DCI algorithm is able to identify changes in edge weights, we only generated DAG models that differ by edge insertions and deletions. This is to provide a fair comparison to the naive approach, where we separately estimate the two DAGs $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ and then take their difference, since this approach can only identify insertions and deletions of edges.

In Figure 5-4 we analyzed how the performance of the DCI algorithm changes over different choices of significance levels α. The simulations were performed on graphs with $p = 10$ nodes, neighborhood size of $s = 3$ and sample size $n \in \{10^3, 10^4\}$. For Figure 5-4a and b we set $\epsilon^{(1)}, \epsilon^{(2)} \sim \mathcal{N}(0, \mathbf{1}_p)$. We compared the performance of DCI to the naive approach, where we separately estimated the two DAGs $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ and then took their difference. For separate estimation we used the prominent PC and GES algorithms tailored to the Gaussian setting. Since KLIEP requires an additional tuning parameter, to understand how α influences the performance of the DCI algorithm, we here only analyzed initializations in the fully connected graph and using the constraint-based method. Both initializations provide a provably consistent algorithm (Wang et al., 2018b). Figures 5-4a and b show the proportion of consistently



(a)

(b)

(c)

Figure 5-4: Proportion of consistently estimated difference-DAGs for 100 realizations per setting with $p = 10$ nodes and sample size n. PC and GES is compared with DCI initialized with the fully connected graph (DCI-FC) and using the constraint-based method (DCI-C). (a) and (b) show the proportion of consistently estimated difference-DAGs when considering (a) just the skeleton and (b) both skeleton and edge orientations. α is the significance level used for the hypothesis tests in the algorithms. (c) shows the proportion of consistent estimates with respect to the number of changes in internal node variances v.

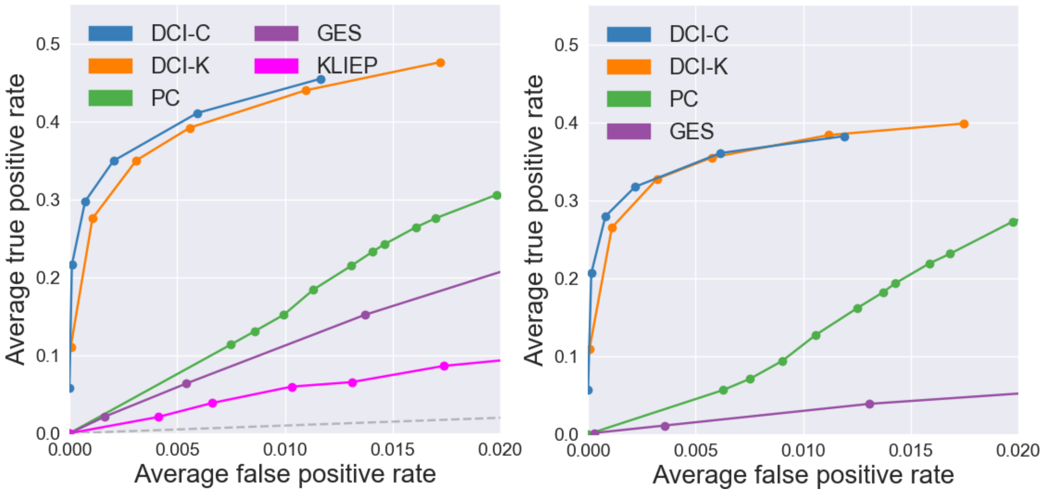
estimated difference-DAGs by just considering the skeleton and both skeleton and orientations, respectively. For PC and GES, we considered the set of edges that appeared in one estimated skeleton but disappeared in the other as the estimated skeleton of the difference-DAG $\tilde{\Delta}$. In determining orientations, we considered the arrows that were directed in one estimated CPDAG but disappeared in the other as the estimated set of directed arrows. We used the exact recovery rate as the evaluation criterion. Both variants of the DCI algorithm outperformed taking differences after separate estimation. Figure 5-4a and b also show that the PC algorithm outperformed GES, which is unexpected given previous results showing that GES usually has a higher exact recovery rate than the PC algorithm for estimating a single DAG. This is due to the fact that while the PC algorithm usually estimates less DAGs correctly, the incorrectly estimated DAGs tend to look more similar to the true model than the incorrect estimates of GES (as also reported in (Solus et al., 2017)) and can still lead to a correct estimate of the difference-DAG.

In Figure 5-4c we analyzed the effect of changes in the noise variances on estimation performance. We set $\epsilon^{(1)} \sim \mathcal{N}(0, \mathbf{1}_p)$, while for $\epsilon^{(2)}$ we randomly picked v nodes and uniformly sampled their variances from $[1.25, 2]$. We used $\alpha = .05$ as significance level based on the evaluation from Figure 5-4. As we increase the number of nodes i such that $\epsilon_i^{(1)} \neq \epsilon_i^{(2)}$, the number of edges whose orientations can be determined decreases. This is because Algorithm 3 can only determine an edge's orientation when the variance of at least one of its nodes is invariant. Moreover, Figure 5-4c shows that the accuracy of Algorithm 2 is not impacted by changes in the noise variances.

Finally, Figure 5-5 shows the performance (using ROC curves) of the DCI algorithm in the high-dimensional setting when initiated using KLIEP and the constraint-based method. The simulations were performed on graphs with $p = 100$ nodes, expected neighborhood size of $s = 10$, sample size $n = 300$, and $\epsilon^{(1)}, \epsilon^{(2)} \sim \mathcal{N}(0, \mathbf{1}_p)$. $B^{(2)}$ was derived from $B^{(1)}$ so that the total number of changes was 5% of the total number of edges in $B^{(1)}$, with an equal amount of insertions and deletions. Figure 5-5 shows that DCI with both initializations performs similarly well and outperform separate estimation using GES and the PC algorithms. The respective plots for 10% change between $B^{(1)}$ and $B^{(2)}$ are given in Supplementary Figure D-1.

Real data analysis

We evaluate DCI on real gene expression data. First, we evaluate DCI for learning the causal difference gene regulatory network on single-cell gene expression data and quantify its performance in predicting the effects of gene perturbations. Note that a major advantage of our work is the ability to learn a causal as opposed to an undirected graph, which enables us to predict the effects of interventions on genes



(a)

(b)

Figure 5-5: High-dimensional evaluation of the DCI algorithm on synthetic data; $(a) - (b)$ are the ROC curves for estimating the (a) skeleton of the difference-DAG and (b) the difference-DAG with $p = 100$ nodes, expected neighbourhood size $s = 10$, $n = 300$ samples, and 5% change between DAGs.

and evaluate them against true effects of interventions, measured experimentally. In the following, we assess the performance of DCI on two datasets collected via CROP-seq (Datlinger et al., 2017) and Perturb-seq (Dixit et al., 2016). Both of these experimental techniques collect, in a pooled fashion, single-cell gene expression data with no interventions (observational data) as well as single-cell gene expression data where some genes were knocked out via CRISPR/Cas9 (interventional data). We use the observational data to learn a causal difference gene regulatory network via DCI and evaluate this graph against the held-out CRISPR/Cas9 gene knockouts, similar in spirit to prior evaluations of causal inference methods (Wang et al., 2017).

As a final evaluation, we apply DCI to bulk gene expression data of patients with ovarian cancer with different survival rates. We provide a qualitative evaluation of our method by considering prior literature.

CROP-seq: Naive versus activated T cells

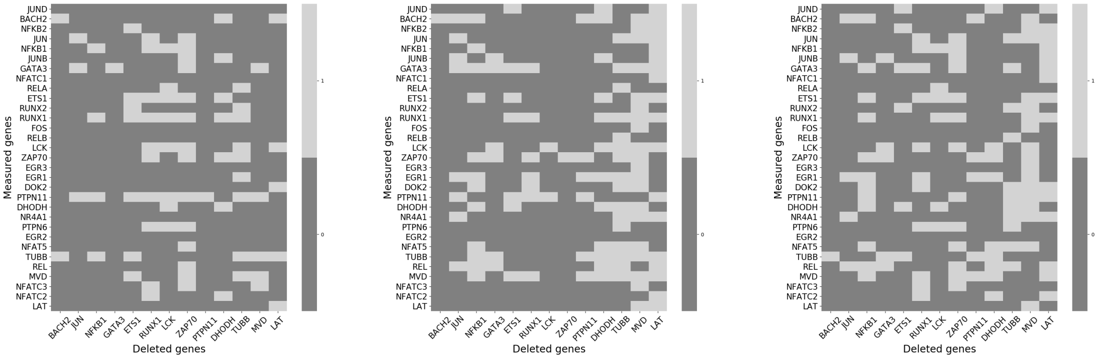
We test our method on gene expression data collected via CROP-seq for naive and activated Jurkat T cells. In particular, we use DCI to learn the differences in the gene regulatory networks as a result of T-cell activation. The CROP-seq data includes 615 observational naive Jurkat T cells and 1320 observational activated Jurkat T cells. As in the original CROP-seq study (Datlinger et al., 2017), we normalize the gene expression of each cell by the total number of reads corresponding to the cell, scale expression by 10^4 and apply a $\log_2(x+1)$ transformation to the data. The data is mean-centered prior to applying our algorithm. We follow (Datlinger et al., 2017) in focusing on genes most relevant to T-cell activation and keep genes that have non-zero variance, resulting in 31 genes.

We apply DCI on the observational naive and activated gene expression data to directly obtain the causal difference gene regulatory network (difference-DAG), which contains edges that appeared, disappeared or changed weight between the two cell states. We report the performance of DCI when initialized in the complete graph, initialized with the difference undirected graph estimated via KLIEP (ℓ_1 regularization set to 0.005) and with the constraint-based method (significance threshold set to 0.005). Additionally, we compared the performance of DCI to the naive approach of running classical causal inference algorithms such as PC (Spirtes et al., 2000) or GES (Meek, 1997) on each dataset (naive and activated) separately, obtaining two causal graphs and then taking the difference. We consider an edge to be in the difference-DAG if the edge was directed in one causal graph and absent in the other causal graph.

As previously mentioned, we can use gene knockouts, collected as part of the CROP-seq study for evaluation of the causal difference gene regulatory network. Note that if perturbing a gene affected the gene expression distribution of another gene, this means that the perturbed gene is upstream of the affected gene in the gene regulatory network. In the following we describe how we estimate the differences in the effects of CRISPR/Cas9 perturbations on genes between the two states (naive and activated) to construct an ROC curve for evaluating the DCI algorithm versus naive applications of PC and GES.

First, for each condition (naive and activated), we separately obtain a matrix that describes which gene knockouts had an effect on which genes (Figures 5-6a and 5-6b). Then, we take the difference between these matrices to determine the differences in the effects of perturbations (Figure 5-6c). In order to construct the matrices in Figures 5-6a and 5-6b, for each condition, we estimate the impact of each gene deletion $j \in \{1, \ldots, d\}$ on each of the measured genes $i \in \{1, \ldots, p\}$ by testing whether the observational distribution (no intervention) of the measured gene i is significantly different from the interventional distribution of the measured gene i when gene j was deleted using a Wilcoxon rank-sum test. We form a $p \times d$ matrix of p-values, Q, from the Wilcoxon rank-sum tests. Next, each column j in Q is thresholded using the entry q_{jj}, which is the p-value obtained by comparing the distribution of the gene expression level of a deleted gene versus its distribution without intervention. The rationale is that knocking out a particular gene should result in a change in its own gene expression distribution and can be used as a baseline to threshold the other entries in the column. In particular, we conclude that q_{ij} is significant if and only if $q_{ij} \leq q_{jj}$. After thresholding the matrix Q in this manner, we obtain the binary matrices in Figures 5-6a and 5-6b, which summarize the effects of the interventions. By forming the difference of these binary matrices we obtain the binary matrix Q^{Δ} in Figure 5-6c. Since not all CRISPR/Cas9 knockouts were effective, here we focused our analysis on the top most effective interventions, which were prioritized based on the maximum q_{jj} p-value (taken over two conditions), using the mean p-value as the cutoff to filter interventions.

We use the matrix of differences in the effects of interventions to evaluate DCI, PC and GES by constructing an ROC curve. If the predicted difference-DAG has a directed edge from j to i, we count this edge as a true positive if $Q^{\Delta}_{ij} = 1$, i.e. there was a difference in the effect of knocking out gene j on gene i between the two conditions. If the predicted difference-DAG has a directed edge from $j \rightarrow i$ but $Q^{\Delta}_{ij} = 0$, the edge is counted as a false positive. Note that this definition of a false positive is overly conservative, since we may have $Q^{\Delta}_{ij} = 0$ if q_{ij} is significant in both matrices, but the magnitude of the effect changes. In other words, $Q^{\Delta}_{ij} = 1$ only captures additions/deletions of edges, but does not capture changes in edge weights. We construct an ROC curve by varying the parameters of DCI, PC and GES. The ROC curve in Figure 5-7 shows that DCI outperforms PC and GES in predicting the



(a) (b) (c)

Figure 5-6: Effects of gene deletions estimated from CROP-seq data; (a) naive T cells, (b) activated T cells, and (c) the difference between the binary matrices in (a) and (b), i.e., the difference in the effects of each gene deletion on the measured genes between naive and activated T cells; this binary matrix is taken to be the ground truth for constructing ROC curves.

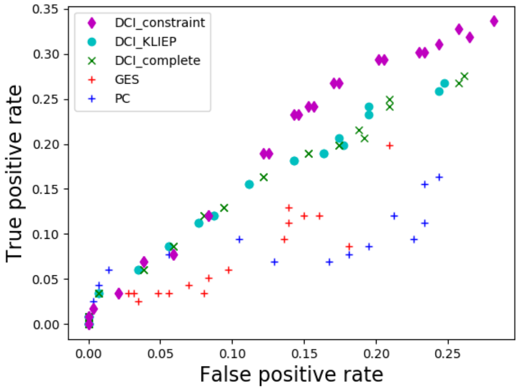
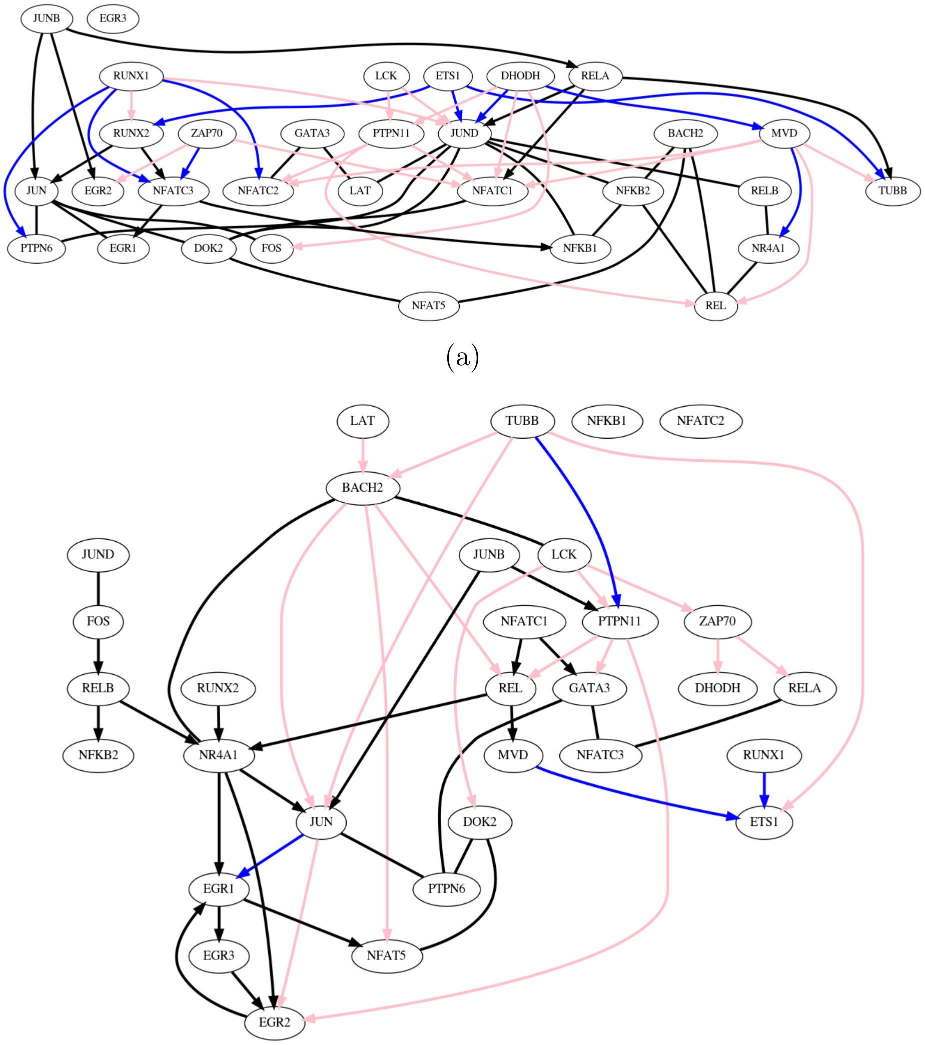


Figure 5-7: ROC plot evaluating DCI (initialized in the undirected difference graph estimated via the constraint-based method, KLIEP as well as in the complete graph), GES and PC on the CROP-seq data for predicting the differences in the effects of gene knockouts. Each point in the ROC curve represents a run with different tuning parameters for DCI, PC and GES.

effects of interventions on this single-cell gene expression dataset. In order to quantify the improvement of the DCI algorithm over the naive approaches, we report a p-value quantifying the difference from random guessing. On the CROP-seq dataset, the PC algorithm achieved a p-value of 0.84, GES a p-value of 0.99, DCI_complete a p-value of 1.1×10^{-7}, DCI_KLIEP a p-value of 3.0×10^{-9}, and DCI_constraint a p-value of 1.5×10^{-10}. The p-value is calculated by sampling causal graphs from an Erdös-Renyi model and quantifying the number of true and false positives. For each false positive level, we created a distribution over true positives based on the sampled random causal graphs and calculated the p-value for the number of true positives obtained from the PC, GES and DCI algorithms. The p-values were combined using Fisher's method and this combined p-value was used for evaluating the causal algorithms. In Figure 5-8, we include examples of the estimated difference gene regulatory networks inferred via DCI (our algorithm) and GES (the best performing baseline). The p-value analysis indicates that PC and GES do not produce good results - this is expected because both methods need to infer the full (potentially dense) gene regulatory networks for each condition as opposed to a likely sparse difference gene regulatory network. One of the main motivations for the DCI algorithm is the fact that high-degree nodes pose a challenge even for state-of-the-art causal inference algorithms; the computational complexity and statistical guarantees of these algorithms depend exponentially on the maximum in-degree of the graph (Chickering and Meek, 2015) and the detrimental effect of large neighborhood size has also been observed empirically (Solus et al., 2017). Since gene regulatory networks are believed to have many high-degree nodes, this may explain the poor performance of PC and GES on such data. On the other hand, the difference gene regulatory networks may likely contain less high-degree nodes, which could explain the improved performance of DCI as compared to PC and GES.

Perturb-seq: Dendritic cells at 0 versus 3 hours post-stimulation

We perform a similar evaluation of DCI on gene expression data collected as part of the Perturb-seq dataset (Dixit et al., 2016). Gene expression data was collected

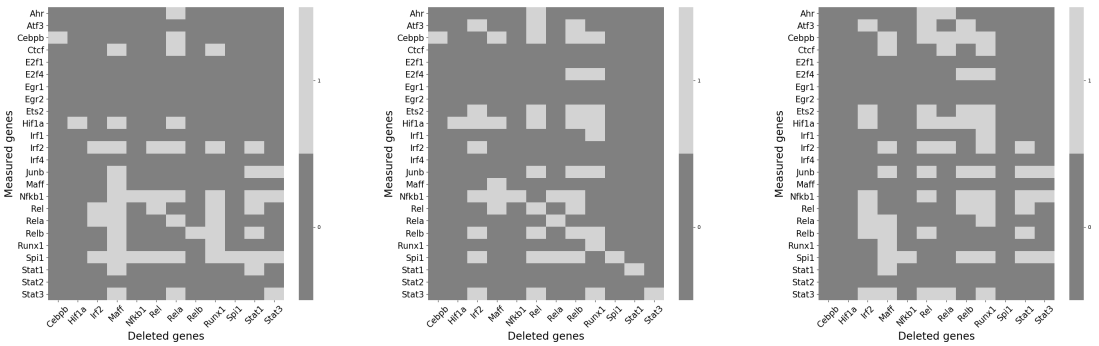


(a)

(b)

Figure 5-8: Examples of difference gene regulatory networks between naive and activated Jurkat T cells, estimated from the CROP-seq data. Difference gene regulatory network inferred via (a) our algorithm, DCI, initialized with KLIEP, which directly learns the difference causal graph from two datasets and (b) baseline causal structure discovery algorithm, GES, which estimates two gene regulatory networks separately and then takes the difference. Blue edges indicate true positives and pink edges indicate false positives. Black edges are the edges inferred to be in the difference gene regulatory network for which ground truth is not available. Graphs were chosen such that the number of false positives is the same across the two methods (16 false positives).

from bone-marrow derived dendritic cells (BMDCs) pre-stimulation (0 hours) and after stimulation with LPS (3 hours). We applied DCI to learn the difference gene regulatory network between these two time points. We used the same procedure for pre-processing Perturb-seq data as we used for CROP-seq. Additionally, we filtered cells for quality, only keeping cells with at least two nonzero counts (CROP-seq dataset already satisfied this filtering constraint). The filtered Perturb-seq data includes 940 observational cells collected at 0 hours and 990 observational cells collected at 3 hours. We followed (Dixit et al., 2016) in focusing on 24 transcription factors that are important for dendritic cell regulation.



(a) (b) (c)

Figure 5-9: Effects of gene deletions estimated from Perturb-seq data; (a) before stimulation with LPS, (b) after stimulation with LPS, and (c) the difference between the binary matrices in (a) and (b), i.e., the difference in the effects of each gene deletion on the measured genes before and after stimulation with LPS; this binary matrix is taken to be the ground truth for constructing ROC curves.

Using the same procedure as performed on the CROP-seq dataset, we constructed the binary matrices describing the effects of gene deletions on measured genes for the two time points (0 and 3 hours) separately, shown in Figures 5-9a and 5-9b, and then determined the difference in the effects of the interventions between the two time points in Figure 5-9c. As above, we constructed an ROC curve, taking the differences in the effects of interventions as the ground truth. The ROC curve (Figure 5-10) shows that in the majority of settings, DCI outperforms the naive approach of estimating two causal graphs separately via PC or GES and taking the difference of the output graph. On the Perturb-seq dataset, the PC algorithm achieved a p-value of 0.91, GES a p-value of 0.5, DCI_complete a p-value of 5.1×10^{-9}, DCI_KLIEP a p-value of 3.1×10^{-9}, and DCI_constraint a p-value of 2.9×10^{-7}. Figure 5-11 shows examples of the estimated difference gene regulatory networks inferred via DCI (our algorithm) and GES (best performing baseline).

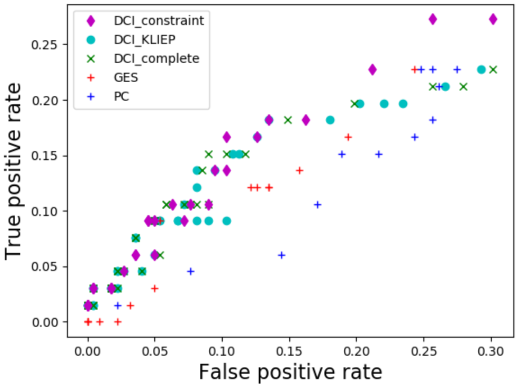
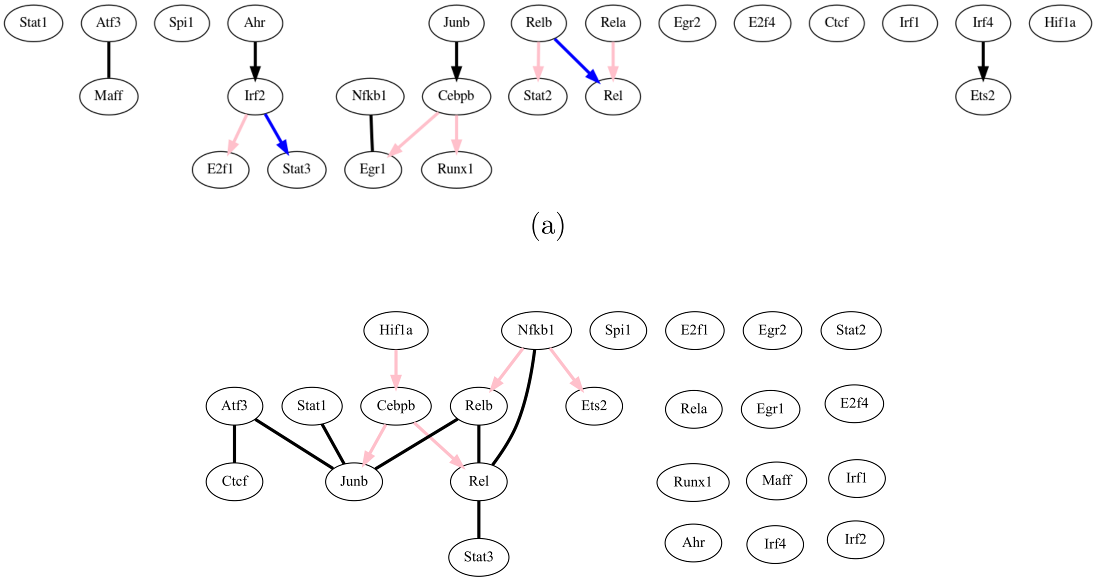


Figure 5-10: ROC plot evaluating DCI (initialized in the undirected difference graph estimated via the constraint-based method, KLIEP as well as in the complete graph), GES and PC on the Perturb-seq data for predicting the differences in the effects of gene knockouts. Each point in the ROC curve represents a run with different tuning parameters for DCI, PC and GES.

Ovarian cancer: patients with different survival rates

We tested our method on an ovarian cancer data set (Tothill et al., 2008) that contains two groups of patients with different survival rates and was previously analyzed using the DPM algorithm in the undirected setting (Zhao et al., 2014). We followed the analysis of (Zhao et al., 2014) and applied the DCI algorithm to gene expression data from the apoptosis and TGF-β pathways. Figure 5-12 shows the estimated difference causal graphs. In the apoptosis pathway we identified two hub nodes: BIRC3, also discovered by DPM, is an inhibitor of apoptosis (Johnstone et al., 2008) and one of the main disregulated genes in ovarian cancer (Jönsson et al., 2014); PRKAR2B, not identified by DPM, has been shown to be important in disease progression in ovarian cancer cells (Cheadle et al., 2008) and an important regulatory unit for cancer cell growth (Chiaradonna et al., 2008). In addition, the RII-β protein encoded by



(a)

(b)

Figure 5-11: Examples of difference gene regulatory networks of dendritic cells before and after stimulation with LPS, estimated from the Perturb-seq data. Difference gene regulatory network inferred via (a) our algorithm, DCI, initialized with KLIEP, which directly learns the difference causal graph from two datasets and (b) baseline causal structure discovery algorithm, GES, which estimates two gene regulatory networks separately and then takes the difference. Blue edges indicate true positives and pink edges indicate false positives. Black edges are the edges inferred to be in the difference gene regulatory network for which ground truth is not available. Graphs were chosen such that the number of false positives is the same across the two methods (5 false positives).

PRKAR2B has been considered as a therapeutic target for cancer therapy (Cho-Chung, 1999; Mikalsen et al., 2006), thereby confirming the relevance of our findings. With respect to the TGF-β pathway, the DCI method identified THBS2 and COMP as hub nodes. Both of these genes have been implicated in resistance to chemotherapy in epithelial ovarian cancer (Marchini et al., 2013) and were also recovered by DPM. Overall, the difference undirected graph discovered by DPM is comparable to the difference-DAG found by our method. More details on this analysis are given in Appendix D along with graphs resulting from naively applying PC and GES and computing differences in Fig. D-2.

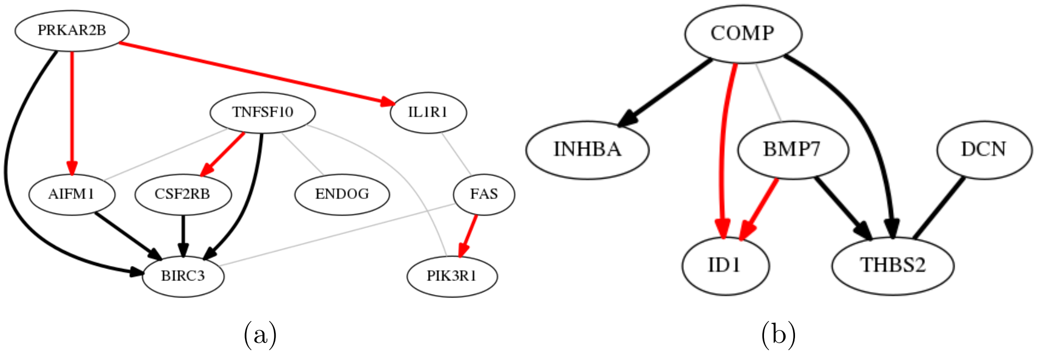


Figure 5-12: Estimate of the difference-DAG between the two groups of ovarian cancer patients for (a) the apoptosis and (b) TGF-β pathways. The black lines represent the edges discovered by both our method and DPM, the red lines represent the edges discovered only by our method, and the grey lines represent the undirected edges discovered only by DPM.

Time complexity comparison

We assess the run time of DCI as compared to the naive approach of estimating each graph separately via PC or GES and then taking the difference on simulated data. For this, we generate 10 different pairs of ground truth causal graphs and sample 10 pairs of datasets from these graphs. For the generation of causal graphs, we sample a weighted adjacency matrix $B^{(1)}$ using an Erdös-Renyi model with expected neighbourhood size of 10, on p nodes. The weights are uniformly drawn from $[-1, -0.25] \cup [0.25, 1]$ to ensure that they are bounded away from zero. The second weighted adjacency matrix $B^{(2)}$ is constructed from $B^{(1)}$ by adding and removing 5 edges (10 changes in total). We sample datasets $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ from the distribution induced by the Gaussian DAG models with $n = 100,000$ samples. Next, we run DCI ($\alpha_{\text{undirected}} = 0, \alpha_{\text{skeleton}} = 0.1, \alpha_{\text{orient}} = 0.1$), PC ($\alpha = 1 \times 10^{-6}$) and GES ($\lambda = 1000$) on $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ and evaluate the CPU time (in seconds) as well as the number of true and false positives averaged over the 10 simulations. As shown in Figure 5-13, DCI is much faster than PC and GES (in terms of mean CPU time) and significantly more accurate as indicated by the average number of true and false positives for each setting. For example, with 500 nodes, on average, DCI runs in 45 *seconds* and results in 9 true positives (out of 10 possible) and 1 false positive, while GES runs in 0.67 *hours* and results in 9.67 true positives and 1529.56 false positives, and PC runs in 4 *hours* and results in 2.33 true positives and 187.56 false positives. Figure 5-14 shows the CPU time of DCI for varied parameter settings, which control the sparsity of the output given by the different steps in the DCI algorithm.

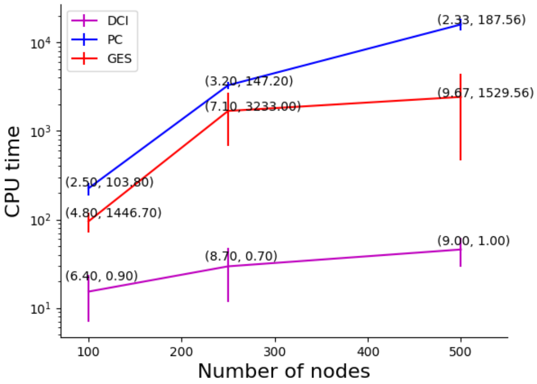
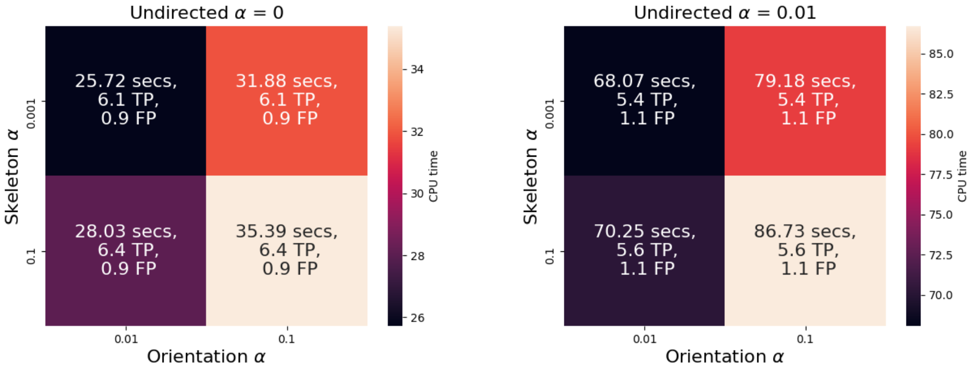


Figure 5-13: CPU time, in seconds, averaged over 10 simulations for variable input size. Each simulation consisted of $p \in \{100, 250, 500\}$ nodes, 10 expected neighbors, 10 changed edges between the graphs and 100,000 samples. DCI was run with $\alpha = 0$ for undirected graph estimation via the constraint-based method, $\alpha = 0.1$ for skeleton estimation, and $\alpha = 0.1$ for inferring the edge orientations. PC was run with $\alpha = 1 \times 10^{-6}$ and GES was run with $\lambda = 1000$. Each point is annotated with a tuple consisting of the average number of true and false positives.



(a) (b)

Figure 5-14: CPU time, in seconds, averaged over 10 simulations for varying tuning parameters (sparsity). Each simulation consisted of $p = 100$ nodes, 10 expected neighbors, 10 changed edges between the graphs and 100,000 samples. DCI was run with (a) $\alpha = 0$ and (b) $\alpha = 0.01$ for undirected graph estimation via the constraint-based method, $\alpha \in \{0.001, 0.1\}$ for skeleton estimation, and $\alpha \in \{0.01, 0.1\}$ for inferring the edge orientations. Each square is annotated with the average CPU time as well as the number of true and false positives.

5.6 Discussion

We presented an algorithm with an accompanying Python package for directly estimating the difference between two causal DAG models given i.i.d. samples from each model. To our knowledge this is the first such algorithm and is of particular interest for learning differences between related gene regulatory networks, where each network might be large and complex, while the difference is sparse. We evaluated DCI on synthetic and real gene expression data, and showed that it outperforms the naive approach of separately estimating two DAG models and taking their difference. We applied our algorithm to gene expression data in bulk and from single cells, showing that DCI is able to predict the effects of interventions and identify biologically relevant genes. This purports DCI as a promising method for identifying intervention targets that are causal for a particular phenotype for subsequent experimental validation.

5.7 Future directions

In order to make DCI scale to networks with thousands of nodes, an important challenge is to reduce the number of hypothesis tests. Currently the time complexity (given by the number of hypothesis tests) of DCI scales exponentially with respect to the size of $\mathcal{C}$ (Wang et al., 2018b). The PC algorithm overcomes this problem by dynamically updating the list of CI tests given the current estimate of the graph. It is an open problem whether one can similarly reduce the number of hypothesis tests for DCI. Furthermore, in many applications (e.g., when comparing normal to diseased states), there is an imbalance of data/prior knowledge for the two models and it is of interest to develop methods that can make use of this for learning the differences between the two models.

Next, while there has been some work on learning causal graphs with cycles (see e.g. Richardson, 2019), the majority of causal inference algorithms, including DCI, assume acyclicity. This assumption stems from the fact that a cause always precedes its effect and thus with sufficient time resolution there would be no feedback loops. However, it is well known that feedback loops are an important feature of biological networks. Therefore, the causal graphs obtained from the methods used in this work (DCI, PC and GES) do not represent the true gene regulatory network and should merely be viewed as useful models for downstream tasks. In particular, learning a DAG allows predicting the effect of interventions, which is one of our main motivations and also how we evaluated the learned networks. While some algorithms have been developed that allow for cycles, this comes at a significant computational cost. Extending our method to allow for feedback loops is an interesting area for future work.

DCI is preferable to separate estimation methods like PC and GES since it can infer not only edges that appear or disappear, but also edges with changed edge weights. However, unlike separate estimation methods, DCI relies on the assumption that the two DAGs share a topological order. Especially in the presence of feedback loops, this assumption may be violated in real biological systems. For example, while in one condition, gene A might control gene B, in a different condition, the other part of the cycle may be activated (gene B controls gene A). Developing methods to directly estimate the difference of two DAGs that do not share a topological order is of great interest for future work.

Another limitation of our method is the assumption of a linear-Gaussian model for gene expression data, which may exhibit complicated nonlinear relationships. Indeed, prior work (Wang et al., 2017) has demonstrated the utility of removing the linear Gaussian assumption when working with gene expression data. While the present work does not investigate nonlinear models, it would be straightforward to extend the current algorithm to the nonlinear setting. For instance, by allowing each function in the structural causal model to be a generalized additive model (GAM), we can associate a vector to each edge, representing the coefficients of each basis function in the model. Then, we can define the difference-DAG by including an edge whenever at least one of these coefficients changes between two settings. Finally, we could modify the algorithm to use GAM regression and hypothesis tests for the invariance of this whole vector. For more complicated models with interaction terms, the difference-DAG would need to be defined slightly differently, but the same ideas still carry through. As for the assumption of Gaussian noise, our test statistic remains valid even for non-Gaussian noise, but we may no longer be able to compute confidence intervals, in which case stability selection may be even more important to obtain robust results.

Furthermore, while we described our framework in the setting with no latent variables, our methodology extends to the setting where the edge weights and noise terms of all latent variables remain invariant across the two DAGs. However, for real world problems, latent variables might have different effects across the two conditions - extending our framework to this situation is of considerable interest for future work.

Finally, since interventional data from CRISPR/Cas9 is becoming more commonplace in biology, it would be valuable to extend DCI to take advantage of interventional data in addition to the observational data.

Chapter 6

Causal Network Models of SARS-CoV-2 Expression and Aging to Identify Candidates for Drug Repurposing

Parts of this chapter have been accepted for publication to Nature Communications: Belyaeva, A.*, Cammarata, L.*, Radhakrishnan, A.*, Squires, C., Yang, K. D., Shivashankar, G. V., & Uhler, C. (2020). Causal Network Models of SARS-CoV-2 Expression and Aging to Identify Candidates for Drug Repurposing.
* These authors contributed equally. My contributions were to design and implement algorithms, perform method and data analysis, and write the manuscript.

6.1 Summary

Given the severity of the SARS-CoV-2 pandemic, a major challenge is to rapidly repurpose existing approved drugs for clinical interventions. While a number of data-driven and experimental approaches have been suggested in the context of drug repurposing, a platform that systematically integrates available transcriptomic, proteomic and structural data is missing. More importantly, given that SARS-CoV-2

pathogenicity is highly age-dependent, it is critical to integrate aging signatures into drug discovery platforms. We here take advantage of large-scale transcriptional drug screens combined with RNA-seq data of the lung epithelium with SARS-CoV-2 infection as well as the aging lung. To identify robust druggable protein targets, we propose a principled causal framework that makes use of multiple data modalities. Our analysis highlights the importance of serine/threonine and tyrosine kinases as potential targets that intersect the SARS-CoV-2 and aging pathways. By integrating transcriptomic, proteomic and structural data that is available for many diseases, our drug discovery platform is broadly applicable. Rigorous in vitro experiments as well as clinical trials are needed to validate the identified candidate drugs.

6.2 Introduction

Candidates for drug repurposing have mainly been identified based on an understanding of their pharmacology or based on retrospective analyses of their clinical effects. Recently, also more systematic computational methods combined with large-scale experimental screens have been employed (Pushpakom et al., 2019). The Connectivity Map (CMap) containing gene expression profiles generated by dosing thousands of small molecules, including many FDA approved compounds, in a number of human cell lines has been particularly valuable in this regard (Subramanian et al., 2017). Common computational approaches include signature matching, where the signature of a drug is determined for example using CMap and compared to the reverse signature of a disease to identify drugs with high correlation (Dudley et al., 2011). In addition, approaches to identify drug or disease networks based on known pathways, protein-protein interactions, gene expression or genome-wide association studies have also been employed (Greene and Voight, 2016; Smith et al., 2012; Gordon et al., 2020). To capitalize on the abundance of data, it is critical to develop computational platforms that can integrate different data modalities including gene expression, drug targets and signatures, as well as protein-protein interactions. In addition, a drug represents an intervention in the system and only a causal framework allows predicting the effect of an intervention. It is therefore critical to capitalize on recent advances in causal inference (Pearl, 2009; Spirtes et al., 2000) in particular with respect to the use of interventional data (Eberhardt, 2007; Meinshausen et al., 2016; Wang et al., 2017; Yang et al., 2017b).

Given the current coronavirus disease 2019 (COVID-19) crisis, there is an urgent need for the development of robust drug repurposing methods. Coronaviruses belong to the family of positive-strand RNA-viruses. While most coronaviruses infect the upper respiratory tract and cause mild illness, they can have serious effects as exemplified by the severe acute respiratory syndrome coronavirus (SARS-CoV) epidemic and now the SARS-CoV-2 pandemic (de Wit et al., 2016). Recent studies have shown that coronaviruses use canonical inflammatory pathways (e.g. NF-κB) of the host cell for their replication, while simultaneously dampening their outward inflammatory signaling (Fung and Liu, 2019; Poppe et al., 2017). This delicate partial up and down-regulation of inflammatory pathways by coronaviruses has represented major challenges for therapeutic interventions (Yang et al., 2017a). While the infection rates for these viruses are similar among different age groups, the morbidity and fatality rates are significantly higher in the aging population (Wu et al., 2020; Onder et al., 2020). The respiratory system of aging individuals is characterized by alterations of tissue stiffness (Sicard et al., 2018). Notably, recent micropatterning experiments have shown that cells subjected to substrates of different stiffness stimulated with the same cytokine (TNF-α) exhibit different downstream NF-κB signaling (Mitra et al., 2017). In a recent commentary, we outlined that the cross-talk between coronavirus infection and cellular aging could play a critical role in the replication of the virus in host cells by differentially intersecting with NF-κB signaling (Uhler and Shivashankar, 2020b). This suggests that efforts for drug repurposing should analyze SARS-CoV-2 infected host cell expression programs in conjunction with aging-dependent programs. While a number of studies are underway that investigate viral integration/replication and interactions with the host cell (Gordon et al., 2020; Zhou et al., 2020), to our knowledge the interplay of SARS-CoV-2 host response and aging has not been explored in the context of drug development and repurposing.

In this study, we propose a novel computational platform for drug repurposing, which integrates transcriptomic, proteomic and structural data with a principled causal framework, and we apply it in the context of SARS-CoV-2 (Fig. 6-1, Supplementary Fig. E-1). Given the age-dependent pathogenicity of SARS-CoV-2, we first identify genes that are differentially regulated by SARS-CoV-2 infection and aging based on bulk RNA-seq data from (Blanco-Melo et al., 2020; Carithers et al., 2015). We then use an autoencoder, a type of artificial neural network used to learn data representations in an unsupervised manner (Baldi, 2012; LeCun et al., 2015), to embed the CMap data together with the SARS-CoV-2 expression data for signature matching to obtain an ordered list of FDA approved drugs. In particular, we show that overparameterized autoencoders align drug signatures from different cell types and thus allow constructing synthetic interventions (Agarwal et al., 2019; Abadie et al., 2010) by translating the effect of a drug from one cell type to another similar in spirit but different from (Hodos et al., 2018), where a tensor-based approach was used. We then construct a combined SARS-CoV-2 and aging interactome using a Steiner tree analysis to connect the differentially expressed genes within a protein-protein interaction network (De Las Rivas and Fontanillo, 2010; Huang and Fraenkel, 2009). By intersecting the resulting combined SARS-CoV-2 and aging interactome with the targets of the top ranked FDA approved drugs from the previous analysis, we identify serine/threonine and tyrosine kinases as potential drug targets for therapeutic interventions. Causal structure discovery methods applied to the combined SARS-CoV-2 and aging interactome show that the identified protein kinase inhibitors such as axitinib, dasatinib, pazopanib and sunitinib target proteins that are upstream from genes that are differentially expressed in SARS-CoV-2 infection and aging, thereby validating these drugs as being of particular interest for the repurposing against COVID-19, post-infection. While we apply our computational platform in the context of SARS-CoV-2, our algorithms integrate data modalities that are available for many diseases, thereby making them broadly applicable.

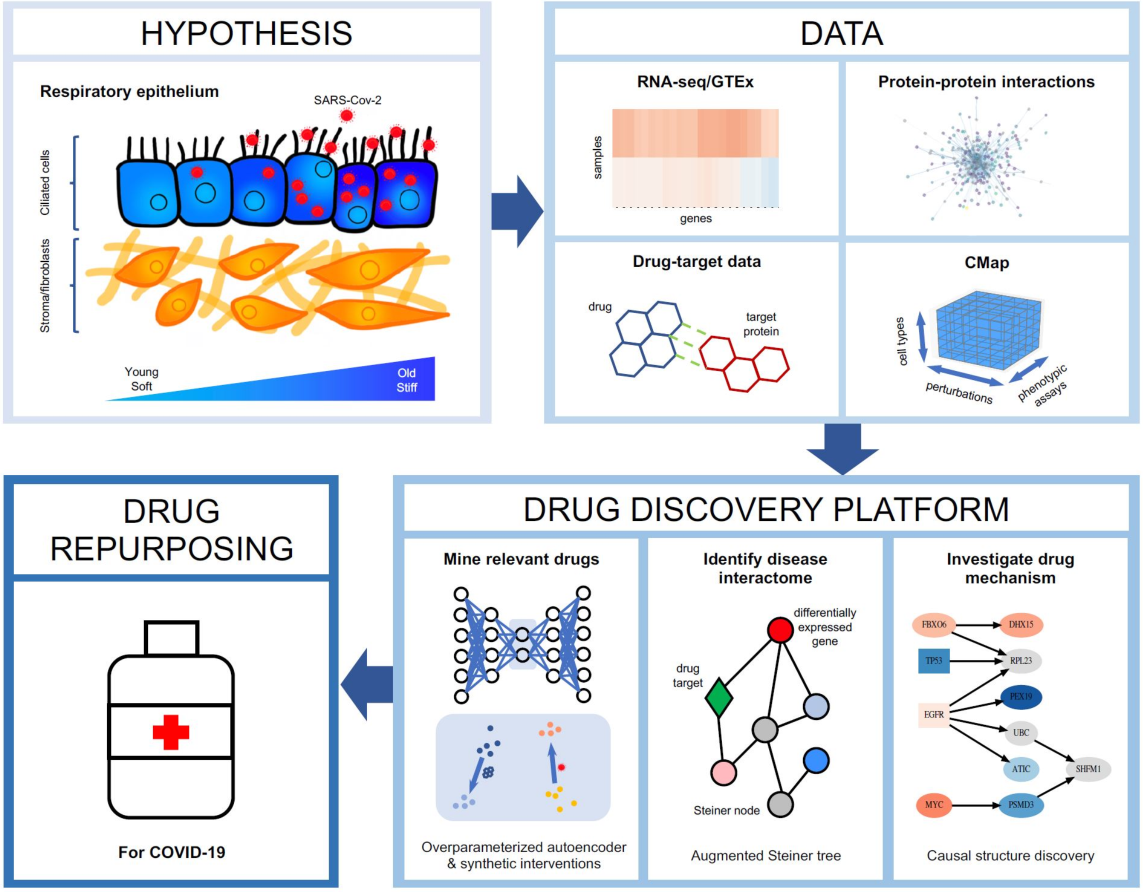


Figure 6-1: Overview of computational drug repurposing platform for COVID-19. (a) COVID-19 is associated with more severe outcomes in older individuals, suggesting that gene expression programs associated with SARS-CoV-2 and aging must be analyzed in tandem. A potential hypothesis regarding the cross-talk between SARS-CoV-2 and aging relies on changes in tissue stiffness in older individuals, outlined in (Uhler and Shivashankar, 2020b). (b) In order to identify potential drug candidates for COVID-19, we integrated RNA-seq data from SARS-CoV-2 infected cells (obtained from (Blanco-Melo et al., 2020)) and RNA-seq data from the lung tissue of young and old individuals (collected as part of the GTEx project (Carithers et al., 2015)) with protein-protein interaction data (from (Razick et al., 2008)), drug-target data (from DrugCentral (Ursu et al., 2019)) and the large-scale transcriptional drug screen CMap (Subramanian et al., 2017). (c) Based on this data, we develop a novel drug repurposing pipeline, which consists of first, mining relevant drugs by matching their signatures with the disease signature in the latent embedding obtained by an overparameterized autoencoder and sharing data across cell types to obtain missing drug signatures via synthetic interventions. Second, we identify a disease interactome within the protein-protein interaction network by identifying a minimal subnetwork that connects the genes differentially expressed by SARS-CoV-2 infection and aging using a Steiner tree analysis. Third, we validate the drugs identified in the first step that have targets in the interactome by identifying the potential drug mechanism using causal structure discovery.

6.3 Results

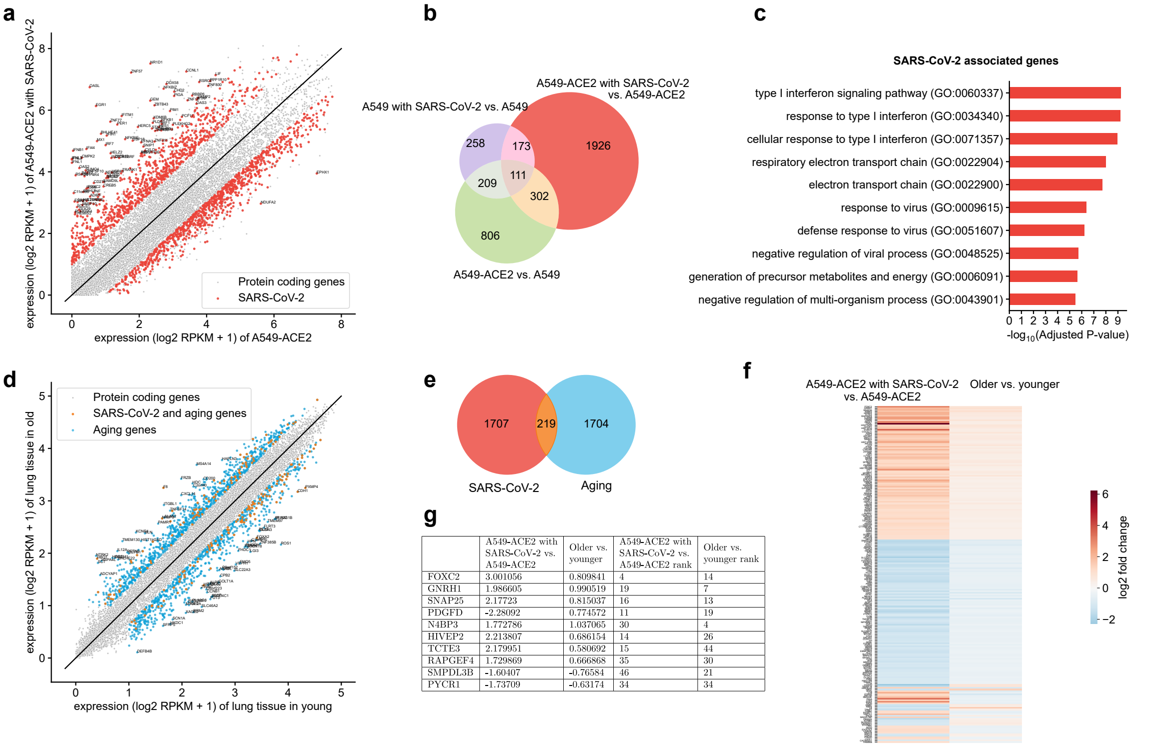
Differential expression analysis identifies genes that intersect the SARS-CoV-2 host response and aging pathways

Since age is strongly associated with severe outcomes in patients with COVID-19, we sought to analyze genes differentially expressed in normal versus SARS-CoV-2 infected cells as well as genes differentially expressed in young versus old individuals. Used as model system for lung epithelial cells and the effect of SARS-CoV-2 infection, we obtained from (Blanco-Melo et al., 2020) RNA-seq samples from normal and SARS-CoV-2 infected A549 lung alveolar cells as well as A549 cells supplemented with ACE2 (A549-ACE2), a receptor that has been shown to be critical for SARS-CoV-2 cell entry (Hoffmann et al., 2020). Fig. 6-2a shows the expression of A549-ACE2 cells infected with SARS-CoV-2 in comparison to normal A549-ACE2 cells, with many genes upregulated as a result of the infection, as expected. Given the availability of A549 data with/without ACE2 and with/without SARS-CoV-2 infection, we removed genes from this initial list of differentially expressed genes that were just ACE2-specific or just SARS-CoV-2 infection-specific to extract a more refined expression pattern of ACE2-mediated SARS-CoV-2 infection (Methods, Fig. 6-2b). The rationale was to remove genes linked to the response of the ACE2 receptor to signals other than SARS-CoV-2 infection or genes involved in the entry of SARS-CoV-2 into the cell through means other than the ACE2 receptor, which has been shown to be the critical mode of entry in humans (Hoffmann et al., 2020). Gene ontology (GO) enrichment analysis revealed enrichment in mitotic cell cycle as the top term, further supporting removal of these genes (Supplementary Fig. E-2). The remaining 1926 genes are denoted in red in Fig. 6-2a,b and used for the subsequent analysis. GO enrichment analysis of these genes revealed that they are significantly enriched in the type I interferon signaling pathway and defense response to virus in addition to other GO terms (Fig. 6-2c). Next, in order to analyze the link between SARS-CoV-2 infection and aging, we analyzed RNA-seq samples from the lung of different aged individuals collected as part of the Genome Tissue Expression (GTEx) study (Carithers et al., 2015). Given the stark increase in case fatality rates of COVID-19 after age 70 (Wu et al., 2020; Onder et al., 2020), we performed a differential expression analysis comparing the youngest group (20-29 years old) and oldest group (70-79 years old), thereby identifying 1923 genes differentially regulated in aging (Fig. 6-2d, Supplementary Fig. E-3). As shown in Fig. 6-2e, these genes show a significant overlap with the 1926 genes found to be differentially regulated by SARS-CoV-2 (p-value= 0.01999, Fisher's exact test), thereby confirming results obtained using a different analysis in (Chow and Chen, 2020). Interestingly, these 219 genes that we found to intersect the SARS-CoV-2 infection and aging pathways (Fig. 6-2e) display concordant changes in gene expression (i.e. the majority of genes is either upregulated or downregulated with SARS-CoV-2 infection and aging) as shown by the $\log_2$-fold changes in Fig. 6-2f and Supplementary Fig. E-4a. The association in the directionality of regulation between SARS-CoV-2 infection and aging is statistically significant (p-value $< 2.2 \times 10^{-16}$, Fisher's exact test), thereby providing further evidence for the interplay of SARS-CoV-2 host response and aging as hypothesized in (Uhler and Shivashankar, 2020b). Fig. 6-2g shows the $\log_2$-fold changes of the 10 most differentially expressed genes across aging and SARS-CoV-2 infection (based on the sum of their ranks with Supplementary Fig. E-4b showing the distribution of the ranks).

Identification of SARS-CoV-2 infection signature in reduced L1000 gene expression space

Next, we focused our analysis on identifying the SARS-CoV-2 transcriptional signature, which we then correlated with the transcriptional signatures of FDA approved drugs in CMap to identify drugs that could revert the effect of SARS-CoV-2 infection. While this analysis resulting in an initial ranking of FDA approved drugs did not take the transcriptional signature of aging into account, aging was a critical component in the final selection of FDA approved drugs described below.

Since gene expression in CMap was quantified using L1000 reduced representation



	A549-ACE2 with SARS-CoV-2 vs. A549-ACE2	Older vs. younger	A549-ACE2 with SARS-CoV-2 vs. A549-ACE2 rank	Older vs. younger rank
FOXC2	3.001056	0.809841	4	14
GNRH1	1.986605	0.990519	19	7
SNAP25	2.17723	0.815037	16	13
PDGFD	-2.28092	0.774572	11	19
N4BP3	1.772786	1.037065	30	4
HIVEP2	2.213807	0.686154	14	26
TCTE3	2.179951	0.580692	15	44
RAPGEF4	1.729869	0.666868	35	30
SMPDL3B	-1.60407	-0.76584	46	21
PYCR1	-1.73709	-0.63174	34	34

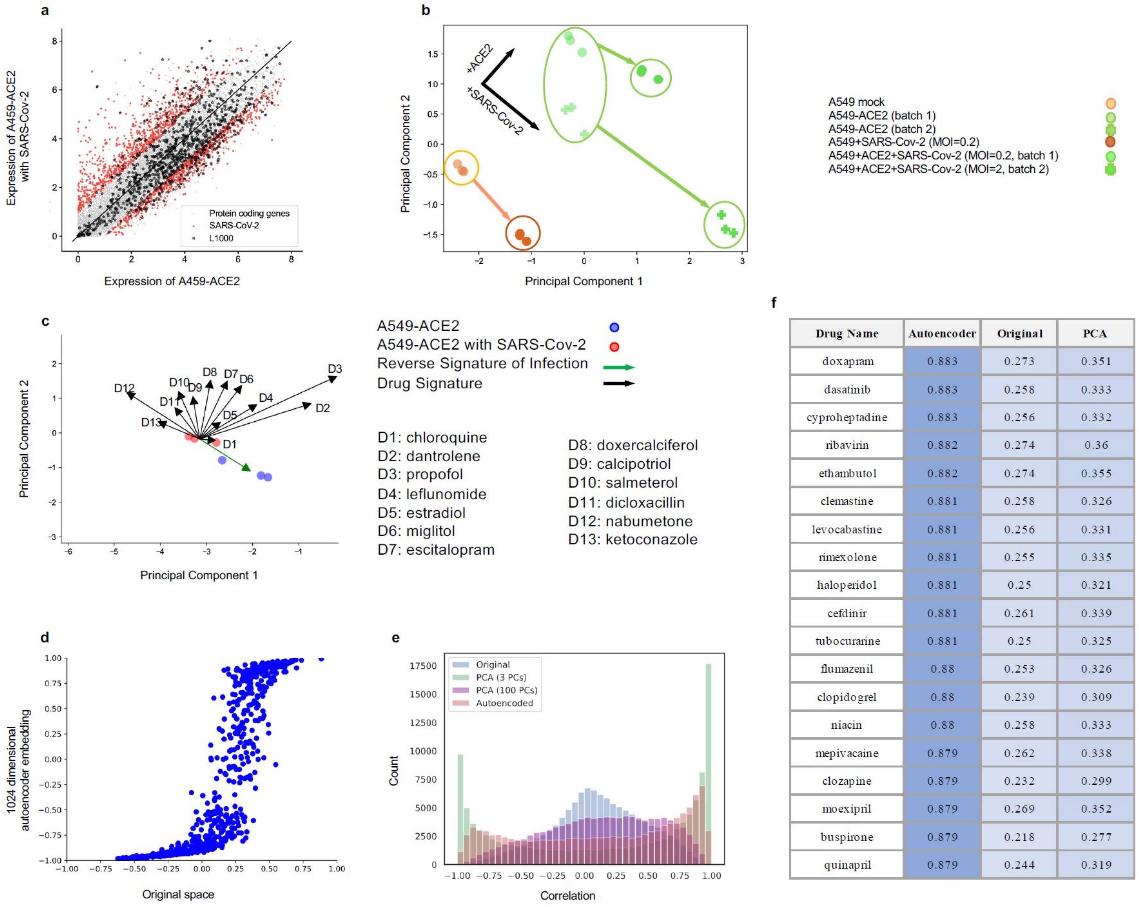
Figure 6-2: Identification of differentially regulated genes in SARS-CoV-2 infection and aging. (a) Gene expression ($\log_2$ RPKM + 1) of A549-ACE2 cells infected with SARS-CoV-2 versus normal A549-ACE2 cells. Genes associated with ACE2-mediated SARS-CoV-2 infection after removing just ACE2-specific or just SARS-CoV-2 infection-specific genes are shown in red. (b) Venn diagram, showing the number of genes in sets considered for obtaining the 1926 genes in the red subset and shown in red in (a) associated with ACE-2 mediated SARS-Cov-2 infection. (c) Top 10 gene ontology terms associated with SARS-CoV-2 infection (adjusted p-value < 0.05). (d) Gene expression ($\log_2$ RPKM + 1) of cells collected from lung tissue of older (70-79 years old) versus younger (20-29 years old) individuals. Differentially expressed genes associated with aging are shown in blue and genes that are associated with both aging and SARS-CoV-2 are shown in orange. (e) Venn diagram of genes associated with SARS-CoV-2 and aging; intersection is significant (p-value $= 0.01999$, Fisher's exact test). (f) Heatmap of $\log_2$-fold changes of differentially expressed genes shared by SARS-CoV-2 and aging; most genes show concordant expression, i.e., they are both upregulated or both downregulated with SARS-CoV-2 infection and ageing. (g) Table of the top 10 most differentially expressed genes across aging and SARS-CoV-2, based on the sum of their ranks with $\log_2$-fold changes for each gene.

expression profiling (Subramanian et al., 2017), which measures gene expression of 1000 representative genes, we first sought to analyze whether these genes sufficiently capture the transcriptional signature of SARS-CoV-2 infection. For this, we intersected the genes measured both by Blanco et al. (Blanco-Melo et al., 2020) and CMap (Subramanian et al., 2017), resulting in 911 genes. We found a statistically significant overlap between the genes identified as differentially expressed by SARS-CoV-2 infection in Fig. 6-2 and the L1000 genes (p-value=7.94×10^{-16}, Fisher's exact test), thereby providing a rational for using the CMap database for drug identification in this disease context (Fig. 6-3a). We thus proceeded to obtain the signature of SARS-CoV-2 infection in the reduced L1000 gene expression space by projecting the RNA-seq data of A549 cells with and without ACE2-receptor and SARS-CoV-2 infection onto the shared 911 genes. The resulting signatures of SARS-CoV-2 infection and ACE2-receptor are visualized using the first two principal components in Fig. 6-3b. Interestingly, the signature of SARS-CoV-2 infection (indicated by arrows) was aligned across both A549 and A549-ACE2 cells as well as across different levels of infection (MOI of 0.2 and 2), suggesting that the SARS-CoV-2 transcriptional signature was captured robustly by the L1000 genes, thus providing further rational for using CMap to identify drugs that could reverse the effect of SARS-CoV-2 infection.

Combined autoencoder and synthetic interventions framework to identify drug signatures and rank FDA approved drugs for SARS-CoV-2

Next, we sought to determine transcriptional drug signatures using the CMap database, which includes among other cell lines A549. The data was visualized using Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) in Supplementary Fig. E-5a, showing that the perturbations clustered by cell type and hence the drug signatures were small relative to the differences between cell types. We intersected the perturbations from CMap with a list of FDA approved drugs using Slinky (Kort and Jovinge, 2019), resulting in 759 drugs of which 605 were available for A549. After removing batch effects using k-means clustering (see Methods and Supplementary Fig. E-5b), we computed initial signatures of these drugs based on the mean before and after drug perturbation in A549 cells. Fig. 6-3c shows a selection of drug signatures in relation to the signature of SARS-CoV-2 infection visualized using the top two principal components.

Since the effect of a drug can be cell-type specific (Niepel et al., 2017), this standard approach to computing drug signatures may not allow extrapolating the obtained signatures beyond A549 cells. In order to determine robust drug signatures and consider also FDA approved drugs that have been dosed on cell lines other than A549 in CMap, we employed an autoencoder framework. Autoencoders, a particular class of neural networks where an input is mapped through a latent space to itself, have been widely used for representation learning (Hinton and Salakhutdinov, 2006; Baldi, 2012; LeCun et al., 2015) and more recently also in genomics and single-cell biology (Yang et al., 2020d,c; Lotfollahi et al., 2019). We trained an autoencoder (architecture described in Supplementary Fig. E-6) to minimize reconstruction error on CMap data and data from Blanco et al. (Blanco-Melo et al., 2020) in the L1000 gene expression space. We then computed the disease and drug signatures based on the embedding of the data in the latent space. Interestingly, by comparing the correlations between drug signatures obtained from A549 cells and MCF7 cells (Fig. 6-3d) as well as HCC515 cells (Fig. 6-4), cell lines with many perturbations in CMap, it is apparent that the autoencoder aligned the drug signatures across different cell types. While autoencoders and other generative models have been used for computing signatures of perturbations also in other works (Lotfollahi et al., 2019; Ghahramani et al., 2018), these works have used autoencoders in the standard way to obtain a *lower*-dimensional embedding of the data. Motivated by our recent work which, quite counter-intuitively, described various benefits of using autoencoders to learn a latent representation of the data that is *higher*-dimensional than the original space (Radhakrishnan et al., 2019), we found that overparameterized autoencoders not only led to better reconstruction of the data than standardly used autoencoders (Supplementary Fig. E-7 and architectures described in Supplementary Fig. E-6), but also to a better alignment of drug signatures between different cell types (Fig. 6-4). Interestingly, overparameterized autoencoders provided about the same alignment of drug signatures as using the top three principal components (Fig. 6-3e), while at the same time allowing a near perfect reconstruction of the original gene expression vectors from the embedding. We thus used this latent space embedding to rank the drugs based on their correlation with the reverse disease signature in A549 cells. Since overparameterized autoencoders aligned drug signatures across cell types, this embedding also allowed constructing synthetic interventions (Agarwal et al., 2019; Abadie et al., 2010), i.e., to predict the effect of a drug on A549 cells without measuring it, by linearly transferring the corresponding drug signature in the latent space from a cell type where it has been measured. In this way, we obtained an enlarged list of drug signatures, which we correlated in the latent space with the reverse disease signature to obtain further candidates of FDA approved drugs for SARS-CoV-2. To compare the correlations obtained with the different embeddings, a list of the top ranked drugs is shown in Fig. 6-3f and the similarity between drug lists is quantitatively assessed by an analysis akin to a receiver operating characteristic (ROC) plot (Supplementary Note and Supplementary Fig. E-8), showing that the drug lists obtained using an embedding in the PCA or the original space are similar but not identical to the autoencoder embedding (AUC of 0.901 and 0.904, respectively). Interestingly, these drug lists contain various drugs that were identified also in (Gordon et al., 2020) using a different analysis (clemastine, haloperidol, ribavirin) or are currently in clinical trials (ribavirin, quinapril). To put these AUC values into perspective and assess the robustness of the identified drug list using the autoencoder embedding, we repeated the analysis on two other SARS-CoV-2 datasets from (Blanco-Melo et al., 2020), namely infected A549 cells without ACE-2 supplement as well as samples collected at a lower MOI (0.2 instead of 2). This resulted in very similar drug lists (Supplementary Fig. E-9); in fact the drug lists from A549 cells with and without ACE-2 supplement in the autoencoder embedding were more similar than the drug lists obtained from the PCA and the original space embedding.



Drug Name	Autoencoder	Original	PCA
doxapram	0.883	0.273	0.351
dasatinib	0.883	0.258	0.333
cyproheptadine	0.883	0.256	0.332
ribavirin	0.882	0.274	0.36
ethambutol	0.882	0.274	0.355
clemastine	0.881	0.258	0.326
levocabastine	0.881	0.256	0.331
rimexolone	0.881	0.255	0.335
haloperidol	0.881	0.25	0.321
cefdinir	0.881	0.261	0.339
tubocurarine	0.881	0.25	0.325
flumazenil	0.88	0.253	0.326
clopidogrel	0.88	0.239	0.309
niacin	0.88	0.258	0.333
mepivacaine	0.879	0.262	0.338
clozapine	0.879	0.232	0.299
moexipril	0.879	0.269	0.352
buspirone	0.879	0.218	0.277
quinapril	0.879	0.244	0.319

Figure 6-3: Mining FDA approved drugs by correlating disease and drug signatures using an overparameterized autoencoder embedding. (a) Gene expression ($\log_2$ RPKM $+1$) of A549-ACE2 cells infected with SARS-CoV-2 versus normal A549-ACE2 cells with genes collected as part of the CMap study using L1000 reduced representation expression profiling method highlighted as stars, showing that L1000 genes significantly overlap with SARS-CoV-2 associated genes (p-value$=7.94\times10^{-16}$, Fisher's exact test). (b) Signature of SARS-CoV-2 infection on A549 and A549-ACE2 cells visualized using the first two principal components based on RNA-seq data from (Blanco-Melo et al., 2020). Signature of SARS-CoV-2 infection is aligned across normal A549 and A549-ACE2 cells as well as across different levels of infection. (c) Comparison of the signatures of a selection of 13 representative FDA approved drugs as compared to the signature of SARS-CoV-2 infection based on A549-ACE2 cells visualized using the first two principal components. Drugs whose signatures maximally align with the direction from SARS-CoV-2 infection to normal are considered candidates for treatment. As expected, drugs have varying signatures of varying magnitudes. (d) Correlation between drug signatures in A549 and MCF7 cells when using the original L1000 expression space versus the embedding obtained from an overparameterized autoencoder. The overparameterized autoencoder aligns the drug signatures in A549 and MCF7 cells by shifting the correlations towards -1 or 1 while maintaining the sign of the correlation in the original space. (e) Histogram of correlations between cell types for a given drug using original L1000 gene expression vectors, overparameterized autoencoder embedding, top 100 principal components, and top 3 principal components. The overparameterized autoencoder achieves about the same alignment of drug signatures as using the top 3 principal components, while at the same time faithfully reconstructing the data (10^{-7} training error). (f) A list of drugs whose signatures maximally align with the direction from SARS-CoV-2 infection to normal in A549-ACE2 cells (MOI 2) with respect to correlations using the overparameterized autoencoder embedding, the original L1000 gene expression space, and the top 100 principal components.

Steiner tree analysis identifies candidate drug targets by constructing combined SARS-CoV-2 and aging interactome

Our differential expression analysis revealed relevant genes to investigate in the context of SARS-CoV-2infection and aging, while the combined autoencoder and synthetic interventions analysis provided candidate FDA approved drugs for reverting the effect of SARS-CoV-2 infection. Next, we integrated these two separate analyses to obtain a final list of FDA approved drugs by constructing a combined SARS-CoV-2 infection and aging protein-protein interactome and intersecting it with the targets

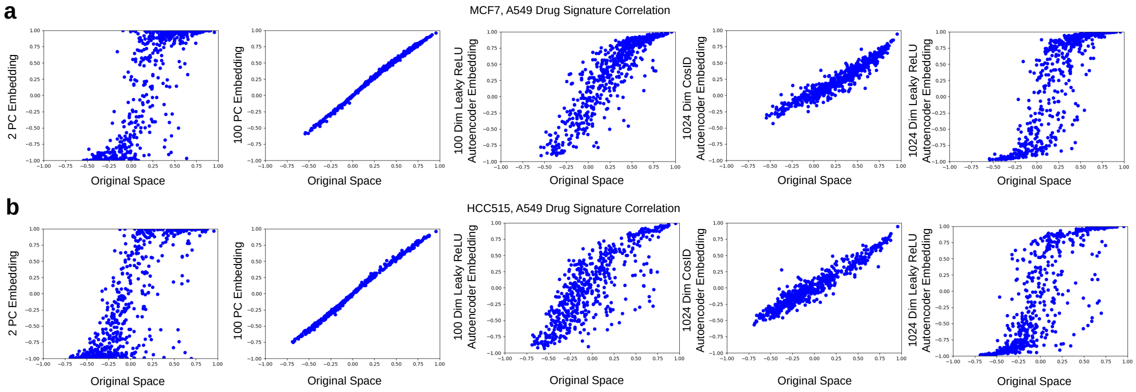


Figure 6-4: Comparison of drug signature alignment between A549 and MCF7 (top) and A549 and HCC515 (bottom) cell types upon using an embedding verus the original space. Embeddings provided include (from left to right) top 2 PCs, top 100 PCs, underparameterized leaky ReLU autoencoder, overparameterized cosid autoencoder, overparameterized leaky ReLU autoencoder. Embeddings from the overparameterized autoencoder with leaky ReLU activation better align drug signatures between these two pairs of cell types than any other embedding considered while still providing near perfect reconstruction of the original data.

of the candidate drugs (Fig. 6-5a). For this, we selected the differentially expressed genes identified in Fig. 6-2f that showed concordant regulation between aging and SARS-CoV-2 infection and intersected them with the nodes of the human protein-protein interaction (PPI) network (IRefIndex Version 14 (Razick et al., 2008)), which contains 182,002 interactions between 15,759 human proteins along with a confidence measure for each interaction. This resulted in 162 protein-coding genes, which we call *terminals* (Supplementary Fig. E-10 and Methods). To gain a better understanding of the molecular pathways connecting these terminal genes, we used a Steiner tree algorithm (Huang and Fraenkel, 2009; Tuncbag et al., 2012) to determine a "minimal" subnetwork or *interactome* within the PPI network that connects these genes (see Methods). A Steiner tree is minimal in that it is a minimum weight subnetwork that connects the terminals. As edge weights in the PPI network we used 1 minus the confidence in the corresponding interactions so as to favor high-confidence edges. After a careful sensitivity analysis to select the various tuning parameters (Methods and Supplementary Fig. E-11), this resulted in an interactome containing 252 nodes and 1,003 edges (Fig. 6-5b and Supplementary Fig. E-12). Interestingly, the interactome contained five genes whose corresponding proteins have been found in (Gordon et al., 2020) to interact with SARS-CoV-2 proteins (EXOSC5, FOXRED2, LOX, RBX1, RIPK1). The 2-nearest-neighborhoods of these proteins are shown in Fig. 6-5c. Another Steiner tree analysis revealed that two additional SARS-CoV-2 interaction partners (CUL2 and HDAC2) were connected to the identified interactome via few high-confidence edges (Supplementary Fig. E-13 to E-15).

Next, we intersected the interactome with the targets of the candidate drugs identified in the previous analysis. A compound was considered if its signature matched the reverse SARS-CoV-2 signature with at least a correlation of 0.86, resulting in 142 FDA approved drugs (see Methods). The targets of these drugs were determined using DrugCentral (Ursu et al., 2016, 2019) and filtered for high affinity (activity constants lower than $10\mu M$, a common threshold used in the field for K_i, K_d, $IC50$ or $EC50$). Interestingly, the resulting drugs, shown in Fig. 6-5d, consisted (with few exceptions) of protein kinase inhibitors (e.g. axitinib, dasatinib, pazopanib, sunitinib). To analyze the specificity of our findings to SARS-CoV-2 infection in aged individuals, we repeated the above analysis without using the GTEx data. This resulted in an interactome containing 1,052 edges across 270 nodes, 42 of which (15%) were also present in the interactome taking age into consideration (Supplementary Fig. E-16). This pure SARS-CoV-2 interactome contained 6 SARS-CoV-2 interaction partners (ETFA, GNB1, NUP62, RBX1, RIPK1, SNIP1). Drugs targeting proteins in this interactome belonged to several families including serotonin inhibitors (clozapine, cyproheptadine, desipramine, methysergide), histamine H1 blockers (clemastine, cyproheptadine, ketotifen), protein kinase inhibitors (including axitinib, dasatinib, pazopanib, sunitinib) and HDAC inhibitors (vorinostat, belinostat). This analysis shows that taking aging into account acted as a valuable filter for the identification of drugs.

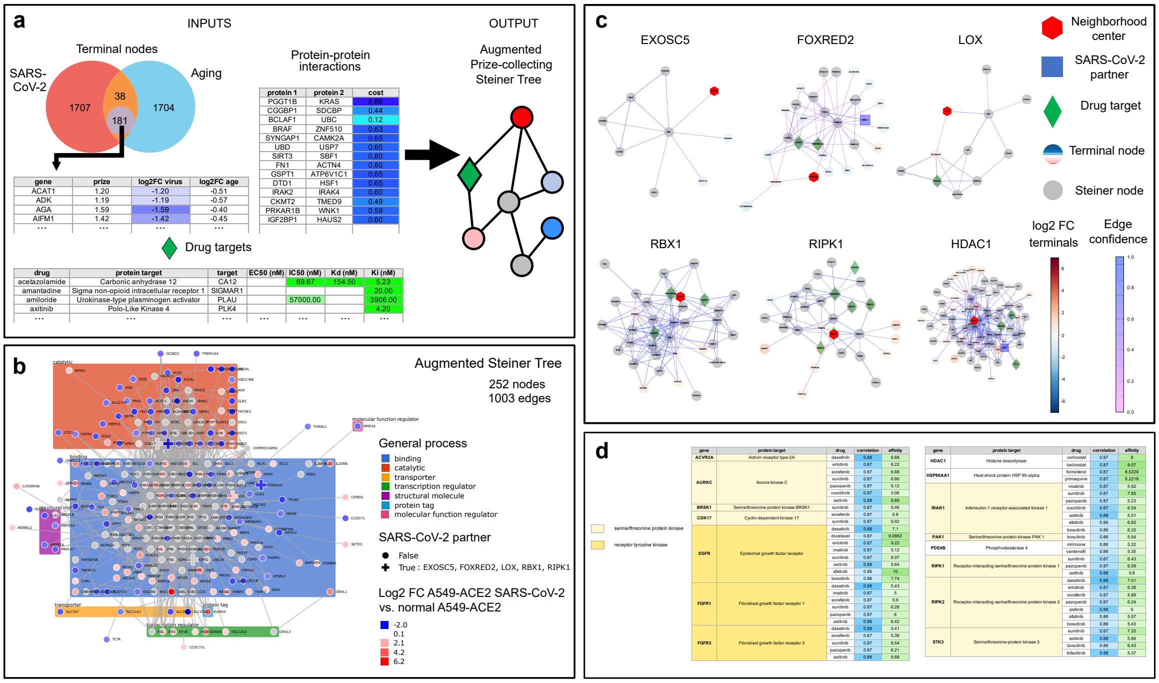


Figure 6-5: Drug target discovery via Steiner tree analysis to identify putative molecular pathways linking differentially expressed genes in SARS-CoV-2 infection and aging. (a) The general procedure takes as input a list of genes of interest (terminal nodes) with prizes indicating their respective importance, a protein-protein interaction (PPI) network with edge cost/confidence information (e.g. from IRefIndex v14), and a list of drugs of interest along with their protein targets and available affinity constants (e.g. from DrugCentral). In this study we consider 181 terminal nodes (of which 162 are present in the IREF network) corresponding to genes differentially expressed in SARS-CoV-2 infection and aging from Fig. 6-2 that are either up-regulated in both SARS-CoV-2 infection and aging or down-regulated in both SARS-CoV-2 infection and aging. The prize of a terminal node equals the absolute value of its $\log_2$-fold change in SARS-CoV-2-infected A549-ACE2 cells versus normal A549-ACE2 cells based on the data from (Blanco-Melo et al., 2020). Terminals and PPI data are processed using OmicsIntegrator2 (Huang and Fraenkel, 2009) to output the disease *interactome*, i.e., the subnetwork induced by a Steiner tree, with drug targets indicated by green diamonds and terminal nodes colored according to their prizes. (b) Interactome obtained using this procedure. Genes are grouped by general function and marked with a cross if known to interact with SARS-CoV-2 proteins based on data from (Gordon et al., 2020). (c) 2-Nearest-Neighborhoods of nodes of interest (denoted by a red hexagon) in the interactome. A threshold was applied on the edge confidence to improve readability. Proteins known to interact with SARS-CoV-2 are denoted by blue squares, drug targets are denoted as green diamonds, terminal nodes are colored according to their $\log_2$-fold change in SARS-CoV-2-infected A549-ACE2 cells versus normal A549-ACE2 cells, Steiner nodes appear in grey. (d) Table of drug targets and corresponding drugs in the interactome. Selected drugs are FDA approved, high affinity (at least one of the activity constants K_i, K_d, $IC50$ or $EC50$ is below $10\mu M$), and match the SARS-CoV-2 signature well (correlation > 0.86). The affinity column displays $-\log_{10}(\text{activity})$. Protein name corresponding to each gene is included.

Causal structure discovery methods validate serine/threonine and tyrosine kinases as critical targets in SARS-CoV-2 infection in the elderly.

Finally, in order to suggest putative causal drug mechanisms and validate the predicted drugs for COVID-19, we supplemented the PPI analysis with causal structure discovery. Since the edges in the PPI network and hence in the SARS-CoV-2 and aging interactome are undirected, it is a-priori not clear whether a drug that targets a node in the interactome has any effect on the differentially expressed terminal nodes, since the target may be downstream of these nodes (Fig. 6-6a). To understand which genes can be modulated by a drug, it is therefore critical to obtain a causal (directed) network. We obtained single-cell RNA-seq data for A549 cells from (Li et al., 2017) and intersected it with the genes present in the combined SARS-CoV-2 and aging interactome. To learn the (causal) regulatory network among these genes, we took advantage of recently developed causal structure discovery algorithms, in particular the greedy sparsest permutation (GSP) algorithm: it performs a greedy search over orderings of the genes to find the sparsest causal network that best fits the data, and it has been successfully applied to single-cell gene expression data before (Solus et al., 2017; Wang et al., 2017; Yang et al., 2017b). To validate the obtained causal model and benchmark the performance of GSP to other prominent causal structure discovery algorithms including PC and GES (Glymour et al., 2019), we took advantage of the gene knockout and overexpression data available from CMap. A causal model should allow predicting the effect of such interventions. Thus, for each such gene knockout and overexpression experiment in CMap that targeted a gene in the interactome, we inferred the genes whose expression changed as a result of the intervention, when compared to control samples (Methods and Supplementary Fig. E-17a). We then constructed receiver operating characteristic (ROC) curves to evaluate GSP, PC and GES by varying their tuning parameters and counting an edge $i \to j$ as a true positive if intervening on gene i resulted in a change in the expression of gene j and a false positive otherwise, thereby showing that GSP exceeded random guessing based on the PPI network (p-value=0.0177, see Methods) and outperformed the other methods (Supplementary Fig. E-17b).

Having established that the causal network obtained by GSP can be used to predict the effect of an intervention, we turned to analyzing the regulatory effects of the identified candidate drugs on the SARS-CoV-2 and aging interactome in A549 cells. The main connected component of the corresponding causal graph is shown in Fig. 6-6a (see also Supplementary Fig. E-18a) highlighting the drug targets and the genes that were found to be differentially expressed by SARS-CoV-2 infection and aging. We then traced the possible downstream effects for each identified drug, thereby finding that the protein kinase inhibitors and HDAC inhibitors could target the majority of differentially expressed genes in this connected component (Supplementary Table E.1). Similarly, we traced the downstream effects for each gene in the interactome that can be targeted by one of the identified drugs, thereby finding that EGFR, FGFR3, HDAC1, HSP90AA1, IRAK1, PAK1, RIPK1, RIPK2, STK3 all have downstream nodes in the interactome with RIPK1 having the largest number of them (127).

To validate these results in a broader context, we obtained single-cell RNA-seq data from (Reyfman et al., 2019) and repeated the analysis in AT2 cells, which have been shown to be critically affected by SARS-CoV-2 in humans (Hoffmann et al., 2020). The resulting causal network for AT2 cells (Supplementary Fig. E-18b) is similar to the one for A549 cells, intersecting it in 55.3% of the edges, with EGFR, HDAC1, HSP90AA1, IRAK1, RIPK1 and RIPK2 all having descendants in the interactome, and targets of protein kinase inhibitors and HDAC inhibitors being particularly central (Supplementary Table E.1). To analyze the most critical targets for the crosstalk between SARS-CoV-2 and aging, we repeated the analysis in the interactome obtained without taking aging into account (Supplementary Fig. E-18c). Interestingly, while HDAC1 and HSP90AA1 continued to have widespread effect, the number of genes downstream of RIPK1 changed drastically to just 1, suggesting that RIPK1 plays a critical role in the SARS-CoV-2 and aging cross-talk. In line with this, while the effect of HDAC inhibitors remained similar in the analysis without ageing, the effect of protein kinase inhibitors changed drastically (Supplementary Table E.1). Collectively, our combined analysis points to protein kinase inhibitors, and it in particular highlights RIPK1, a serine/threonine-protein kinase, as one of the main targets against SARS-CoV-2 infections with a highly age-dependent role and the largest number of downstream differentially expressed genes in the combined SARS-CoV-2 and ageing interactome.

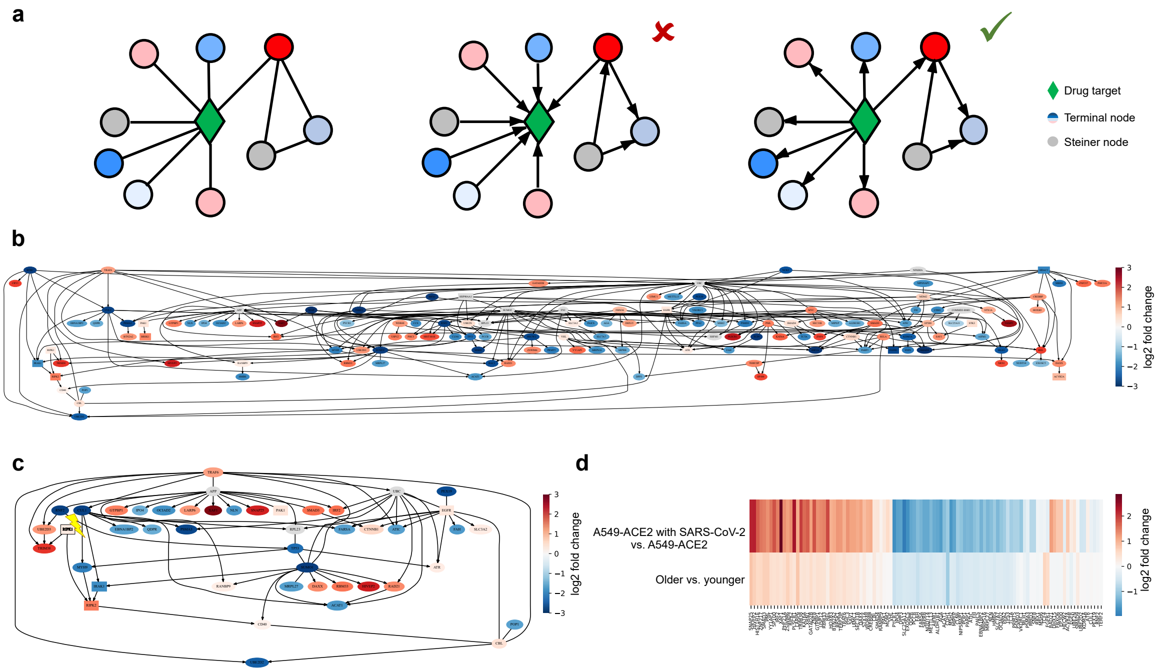


Figure 6-6: Causal mechanism discovery of potential drug targets. (a) In an undirected PPI network (left), edge directions for a particular drug target (green diamond) are unknown. Establishing causal directions is important since it is of interest to avoid drug targets that do not have many downstream nodes in the disease interactome (middle) and instead choose drug targets that have a causal effect on many downstream nodes in the disease interactome (right). (b) Causal network underlying the combined SARS-CoV-2 and aging interactome in A549 cells with gene targets of selected drugs in boxes (largest connected component shown). (c) Causal subnetwork of A549 cells corresponding to nodes within 5 nearest neighbors of RIPK1. The node color corresponds to the $\log_2$-fold change of A549-ACE2 with versus without SARS-CoV-2. (d) Heatmap of $\log_2$-fold change of genes that are downstream of RIPK1.

6.4 Discussion

The repurposing of drugs for SARS-CoV-2 has been a major challenge given the many pathways involved in host-pathogen interactions and the intricate interplay of SARS-CoV-2 with inflammatory pathways (Fung and Liu, 2019; de Wit et al., 2016; Poppe et al., 2017; Yang et al., 2017a). Interestingly, while both young and old individuals are susceptible to SARS-CoV-2 infection, the virus' pathogenicity is significantly more pronounced in the elderly (Wu et al., 2020; Onder et al., 2020). Since the mechanical properties of the lung tissue change with aging (Sicard et al., 2018), this led us to hypothesize an interplay between viral infection/replication and tissue aging (Uhler and Shivashankar, 2020b), suggesting that this could play an important role in drug discovery programs. While ongoing drug repurposing efforts have analyzed host-pathogen interactions and the associated gene expression programs (Gordon et al., 2020; Blanco-Melo et al., 2020), they have lacked an integration with aging. More generally, while a number of data-driven and experimental approaches have been proposed for drug identification and repurposing (Pushpakom et al., 2019), a platform that systematically integrates different data modalities including transcriptomic, proteomic and structural data into a principled causal framework to predict the effect of different drugs has been missing.

By combining bulk RNA-seq data from GTEx (Carithers et al., 2015) and Blanco et al. (Blanco-Melo et al., 2020), we identified a critical group of genes that were differentially expressed by aging and by SARS-CoV-2 infection. While previous analysis relied primarily on contrasting the expression in cells with and without SARS-CoV-2 infection (Chow and Chen, 2020), we made an attempt to separate the effect of the ACE2 receptor alone and the effect of SARS-CoV-2 in cells without ACE2 receptor to extract a more refined differential expression pattern of ACE2-mediated SARS-CoV-2 infection. While previous computational efforts to repurpose drugs have mainly considered two approaches: (1) identifying drug targets by analyzing disease networks based for example on PPI or transcriptomic data (Smith et al., 2012; Greene and Voight, 2016; Gordon et al., 2020), and (2) identifying drugs by matching their signature (for example obtained from the CMap project (Subramanian et al., 2017)) to the reverse disease signature (Dudley et al., 2011), we developed a principled causal framework that encompasses these two approaches. First, in order to ensure that the CMap database, which measures expression using 1000 representative genes, would be useful in the context of SARS-CoV-2, we validated that the intersection of these genes with the SARS-CoV-2 differentially expressed genes was significant. Second, to establish drug signatures based on the CMap database, we employed a particular autoencoder framework (Radhakrishnan et al., 2019). Rather unintuitively, we showed that using an overparameterized autoencoder, i.e. by using an autoencoder not to perform dimension reduction as usual but to instead embed the data into a higher-dimensional space, aligned the drug signatures across different cell types. This allowed constructing synthetic interventions, i.e., to predict the effect of a drug on a cell type without measuring it by using other cell types to infer it. Third, to identify drug targets in the pathways intersecting SARS-CoV-2 and aging, we connected the differentially expressed genes in the PPI network using a Steiner tree analysis (Huang and Fraenkel, 2009) and intersected the resulting interactome with high-affinity targets of the drugs obtained using the overparameterized autoencoder framework. Finally, while computational drug discovery programs have been largely correlative (Pushpakom et al., 2019), we made use of recent causal structure discovery algorithms (Glymour et al., 2019; Solus et al., 2017; Wang et al., 2017) to validate the identified drug targets and their downstream effects, thereby identifying protein kinase inhibitors such as axitinib, dasatinib, pazopanib, and sunitinib as drugs of particular interest for the repurposing against COVID-19.

Among the various protein kinases, in particular from the family of serine/threonine-protein kinases, identified by our drug repurposing pipeline, RIPK1 was singled out by our causal analysis as being upstream of the largest number of genes that were differentially expressed by SARS-CoV-2 infection and aging, while losing its central role in the corresponding gene regulatory network without taking aging into account. Notably, RIPK1 has been shown to bind to SARS-CoV-2 proteins (Gordon et al., 2020) and has also been found to be in an age-dependent module (Chow and Chen, 2020). RIPK1 belongs to an interesting family of proteins comprising of a kinase domain on the N terminus and a death domain on the C terminus; activation of the kinase domain has been associated with epithelial cell homeostasis, while activation of the death domain leads to triggering necroptotic or apoptotic pathways (Festjens et al., 2007; Dannappel et al., 2014), the death pathways potentially triggering tissue fibrosis (Sauler et al., 2019). Interestingly, our differential expression analysis found RIPK1 to be upregulated with SARS-COV-2 infection. We hypothesize that upon SARS-CoV-2 infection in older individuals the death pathways may be favored, thereby leading to fibrosis and increased blood clotting. Consistent with this, recent post-mortem lung tissue biopsies of SARS-CoV-2 human patients revealed a fibrotic epithelium and increased blood clotting (Jose and Manuel, 2020; Spagnolo et al., 2020).

In order to test how specific our findings are to SARS-CoV-2 and demonstrate the broad applicability of our pipeline, we repeated the analysis on gene expression data available from (Blanco-Melo et al., 2020) for respiratory syncytial virus (RSV) and influenza A virus (IAV); see Supplementary Note for a detailed description of the analysis. Differential gene expression analysis showed that the intersection of the identified genes with RSV and IAV was only 3.19% and 19.6%, respectively (Supplementary Fig. E-19). Comparing the drug lists resulting from the overparameterized autoencoder analysis for IAV and RSV to SARS-CoV-2 shows that the drug rankings for SARS-CoV-2 and RSV are significantly different, while the rankings for SARS-CoV-2 and IAV are more similar, but less so than between different SARS-CoV-2 datasets (Supplementary Fig. E-20 and E-9). The Steiner tree analysis further enforced these findings (Supplementary Fig. E-21), which is in line with SARS-CoV-2 and IAV having more similar clinical symptoms with higher morbidity and fatality rates in the ageing population, while RSV is riskier for young children.

Collectively, our results highlight the importance of RIPK1 in the interplay between SARS-CoV-2 infection and aging as a potential target for drug repurposing programs to be administered post-infection. There are various drugs currently approved that non-specifically target RIPK1 (such as pazopanib and sunitinib) as well as under investigation that are highly specific to RIPK1 (Martens et al., 2020; Degterev et al., 2019). Given the distinct pathways elicited by RIPK1, there is a need to develop appropriate cell culture models that can differentiate between young and aging tissues to validate our findings experimentally and allow for highly specific and targeted drug discovery programs. While our method is broadly applicable, we note several limitations. First, our drug repurposing pipeline relies on the availability of RNA-seq data from normal and infected/diseased cells in the cell type of interest and therefore the availability of such data is necessary for the application of our platform. Second, since our autoencoder is trained on CMap data, which only contains the expression of 1000 genes (L1000 genes), it is possible that the signal of the infection may not be captured by these 1000 genes. However, this can be checked by assessing whether there is a statistically significant overlap between the L1000 genes and the differentially expressed genes in the disease of interest, which we performed in our analysis for SARS-CoV-2. Finally, since the CMap data contains a limited set of drugs, it is possible that none of the drugs are anticorrelated with the disease signature, thus preventing the user from identifying drug candidates. While our work identified particular drugs and drug targets in the context of COVID-19, our computational platform is applicable well beyond SARS-CoV-2, and we believe that the integration of transcriptional, proteomic and structural data with network models into a causal framework is an important addition to current drug discovery pipelines.

6.5 Future directions

We proposed a computational platform for drug repurposing and applied it in the context of SARS-CoV-2. An important future direction would be to validate our predictions experimentally. Since COVID-19 is age-dependent and since an important part of our analysis was the inclusion of age, it would be important to develop realistic organoid systems that mimic aged tissue, its microenvironment and its mechanical properties forvalidation of the identified drugs. Given that cells in younger versus older populations are known to display different phenotypes (Angelidis et al., 2019), e.g. it has been hypothesized that subpopulations of epithelial cells in older tissues undergo epithelial-to-mesenchymal transitions (Uhler and Shivashankar, 2020a), for the same viral signal, cells in younger versus older tissues could exhibit different downstream gene expression signals. These age-dependent differences have direct implications for drug discovery and thus should be taken into account during validation. Another interesting future direction would be to apply our computational platform for drug repurposing against other infections or diseases. Currently, large-scale single-cell RNA-seq data is being collected on normal and diseased cells in various contexts. In this work, we used bulk RNA-seq collected in a cell line that was also part of the CMap dataset. Going forward, it would be important to analyze how well our framework generalizes for predicting the effects of drugs for cell types not present in CMap such as those collected from single-cell RNA-seq on real human tissues.

6.6 Methods

Bulk gene expression data

The RNA-seq gene expression data related to SARS-CoV-2 infection in A549 and A549-ACE2 cells was obtained from (Blanco-Melo et al., 2020) under accession code GSE147507. The RNA-seq data of lung tissues for the aging analysis was downloaded from the GTEx Portal (https://gtexportal.org/home/index.html) along with metadata containing the age of the individual from whom the RNA-seq sample was obtained. The RNA-seq raw read counts were transformed into quantile normalized, $\log_2(x + 1)$ scaled RPKM values, following the normalization performed in (Subramanian et al., 2017).

Differential expression analysis

For differential expression analysis, we focused on genes that were highly expressed, filtering out any genes with $\log_2(\text{RPKM} + 1) < 1$ for all considered datasets. In order to determine the ACE2-mediated SARS-CoV-2 genes, we computed three different

$\log_2$-fold changes based on the data from (Blanco-Melo et al., 2020). Namely, we defined as ACE2-mediated SARS-CoV-2 genes all genes that had an absolute $\log_2$-fold change between A549-ACE2 cells infected with SARS-CoV-2 and A549-ACE2 cells above threshold, excluding genes that had an absolute $\log_2$-fold change above the same threshold in A549-ACE2 cells versus A549 cells and also excluding genes that had an absolute $\log_2$-fold change above the same threshold in A549 cells infected with SARS-CoV-2 versus normal A549 cells. In other words, the ACE2-mediated SARS-CoV-2 genes were defined as the genes denoted in red in the Venn diagram in Fig. 6-2b (with pink, brown and yellow subsets removed). The absolute $\log_2$-fold change threshold was determined such that the number of ACE2-mediated SARS-CoV-2 genes was 10% of the protein coding genes.

In order to determine the age associated genes, we analyzed lung tissue samples obtained from the GTEx portal (https://gtexportal.org/home/index.html) from individuals of varying ages. We computed the absolute $\log_2$-fold change between samples of the lung tissue from older (70-79 years old) and younger (20-29 years old) individuals, defining the age associated genes as the top 10% of protein coding genes with highest absolute $\log_2$-fold change. We also considered defining age-associated genes based on the absolute $\log_2$-fold change comparing individuals who are 20-29 years old versus 60-79 years old, which yielded similar age-associated genes, with 1339 out of the 1923 genes in common between the two sets as shown in Supplementary Fig. E-3b.

Gene ontology enrichment analysis

Gene ontology analysis was performed on a given gene set using GSEApy, keeping the top 10 gene ontology biological process terms with lowest p-values. All reported terms had p-values ≤ 0.05, after adjusting for multiple hypothesis testing using the Benjamini–Hochberg procedure.

L1000 gene expression data from CMap

The CMap data measured via L1000 high-throughput reduced representation expression profiling, which quantifies the expression of 1000 landmark genes, was obtained from (Subramanian et al., 2017) under accession code GSE92742. We chose level 2 data, truncated to only the genes that were also measured by (Blanco-Melo et al., 2020), and then performed $\log_2(x+1)$ scaling and min-max scaling on each of the resulting 911-dimensional expression vectors.

Combined autoencoder and synthetic interventions framework

We first describe our training procedures for the autoencoder framework. CMap contains a total of 1,269,922 gene expression vectors and we performed a 90-10 training-test split resulting in 1,142,929 training examples and 126,993 test examples. We selected the best model by applying early stopping with an upper bound on the number of total epochs being 150. Note that this is well past the usual early stopping method of applying a patience strategy with a patience of at most 10 epochs (Goodfellow et al., 2016). All hyperparameter settings, optimizer details, and architecture details are presented in Supplementary Fig. E-6c. To summarize, we considered a range of fully connected autoencoders with varying depth, width, and nonlinearity, and we used Adam with a learning rate of 10^{-4} for optimization. To compute the drug signatures via the trained autoencoder, we used as embeddings the output of the first hidden layer prior to application of the activation function.

Drug signatures for the A549 cells (and similarly for the MCF7 and HCC515 cells) in CMap were computed by taking the difference between the mean embedding for the A549 samples with drug and the mean embedding for the A549 control (DMSO) samples. To remove batch effects, we performed k-means clustering of the control samples in the embedding space and removed all points falling in the smaller of the two clusters (see Supplementary Fig. E-5b). Subsequent analysis of the removed cluster revealed that it consisted of samples with a minimum gene expression value of 1 (after $\log_2(x+1)$ scaling), while all other gene expression values fell in the range of [5, 13], thereby providing further reason for removal of this cluster.

Next, we briefly describe the synthetic interventions framework and how the embedding from our trained overparameterized autoencoder is used for this. The traditional application of synthetic interventions (Agarwal et al., 2019; Abadie et al., 2010) in the context of drug repurposing would proceed as follows: when a drug signature is unavailable on a given cell type but is available on other cell types, we would express the cell type as a linear combination of the other cell types and use this linear combination to predict the signature on the cell type for which data is unavailable. Since we demonstrated that over-parameterized autoencoders align drug signatures between different cell types (Fig. 6-4), instead of using a linear combination of drug signatures across cell types, we can simply use one of the available drug signatures as the synthetic intervention. In particular, in this work, we used drug signatures on MCF7 cells to construct synthetic interventions for A549 cells. We also considered drug signatures on HCC515 cells; however, there was only one FDA approved drug that was applied to HCC515 cells which was not also applied to A549 cells in CMap. While this analysis did not help to increase the number of considered drugs, we used the data on HCC515 cells in conjunction with the data on A549 and MCF7 cells to validate that the overparameterized autoencoder aligns the signatures of drugs between different cell types (Fig. 6-3d and Fig. 6-4).

Cosine similarity between perturbations

For each cell type and perturbation, we computed a cell-type specific “perturbation signature", which is defined as the difference between the average gene expression of a cell type under that perturbation and under the control perturbation, DMSO. Then, for each perturbation, we computed the cosine similarity $\left(\frac{a \cdot b}{\|a\| \|b\|}\right)$ between the perturbation vectors for all pairs of cell types which received that perturbation in CMap. For example, daunorubicin was applied to 14 cell types in cMap, resulting in $\binom{14}{2} = 91$ cosine similarities associated with daunorubicin. All cosine similarities were plotted (Fig. 6-3e).

Steiner tree analysis

Human protein-protein interaction (PPI) network

A weighted version of the publicly available IRefIndex v14 human PPI network (Razick et al., 2008) was retrieved from the OmicsIntegrator2 GitHub repository (http://github.com/fraenkel-lab/OmicsIntegrator2). The interactome contains 182,002 interactions between 15,759 proteins. Each interaction e has an associated cost $c(e) = 1 - m(e)$ where the score $m(e)$ is obtained using the MIScore algorithm (Kedaigle, 2018), which quantifies confidence in the interaction e based on several evidence criteria (e.g. number of publications reporting the interaction and corresponding detection methods).

Human-SARS-CoV-2 PPI network

A high-confidence host-pathogen interaction map of 27 SARS-CoV-2 viral proteins with HEK293T proteins (Gordon et al., 2020) was retrieved from NDEx (http://www.ndexbio.org/#/network/5d97a04a-6fab-11ea-bfdc-0ac135e8bacf), which reports interactions with 332 human proteins.

Drug-target interaction data

Data on the targets of drugs was obtained from DrugCentral (http://drugcentral.org/download), an online drug information resource, which includes drug-target interaction data extracted from the literature along with metrics (such as inhibition constant K_i, dissociation constant K_d, effective concentration $EC50$, and inhibitory concentration $IC50$) measuring the affinity of the drug for its target (Ursu et al., 2016, 2019). Drugs in the database are approved by the FDA and may also be approved by other regulatory agencies (such as the EMA). From this database, we filtered out compounds targeting non-human proteins. We also discarded drug-target pairs with affinity metrics (K_i, K_d, $EC50$ or $IC50$) higher than $10\mu M$, a commonly used threshold in the field. Based on this filtering we obtained a data set containing 12,949 high affinity drug-target pairs involving

1,457 unique human protein targets and 2,095 unique compounds. This dataset was further restricted to drugs predicted to reverse the SARS-CoV-2 signature (correlation greater than 0.86 in the overparameterized autoencoder embedding). This correlation threshold was chosen to be the point at which the proportion of drugs decreases the most rapidly (Supplementary Fig. E-22). As a result, the final drug-target data set included information on 2,296 drug-target pairs involving 652 unique human gene targets and 117 unique FDA approved drugs.

Prize-collecting Steiner forest algorithm

The Prize-Collecting Steiner Forest (PCSF) problem is an extension of the classical Steiner tree problem: Given a connected undirected network with non-negative edge weights (costs) and a subset of nodes, the *terminals*, find a subnetwork of minimum weight that contains all terminals. The resulting subnetwork is always a tree, which in general contains more nodes than the terminals; these are known as *Steiner nodes*. In the special case when there are only 2 terminals, this boils down to finding the shortest path between these nodes. The Steiner tree problem in general is known to be NP-complete, but various approximations are available. The PCSF problem generalizes this problem by introducing prices for the terminals (in addition to the edge costs already present in the Steiner tree problem) and a dummy node connected to all terminals. The problem is then to find a connected subnetwork that minimizes an objective function involving the cost of selected edges and the prizes of terminals that are missing from the subnetwork as detailed below; we used OmicsIntegrator2 to solve this optimization problem (Huang and Fraenkel, 2009).

To formally introduce the objective function, let $G = (V, E, c(\cdot), p(\cdot))$ denote the undirected PPI network with protein set V (containing N proteins), interaction set E, edge cost function $c(\cdot)$, set of terminals $S \subset V$ (containing N proteins) and attributed prizes $p(\cdot)$. The version of the PCSF problem solved by OmicsIntegrator2 (Huang and Fraenkel, 2009) and used in this study consists of finding a connected subnetwork $T = (V_T, E_T)$ of the modified graph $G^* = (V \cup \{r\}, E \cup \{\{r, s\} : s \in S\})$ that minimizes the objective function

$$\psi(T) = b \sum_{v \notin V_T} p(v) + \sum_{e \in E_T} c^*(e)$$

The node r is a dummy root node connecting all terminals in the network. The parameter $b \in \mathbb{R}^+$ linearly scales the node prizes (which are non-zero for terminal nodes exclusively), and the modified edge cost function $c^*(\cdot)$ can be expressed as follows. For any edge $e = \{x, y\}$

$$c^*(e) = \begin{cases} c(e) + \frac{d_x d_y}{d_x d_y + (N - d_x - 1)(N - d_y - 1)} 10^g & \text{if } e \in E \\ w & \text{if } e \in \{\{r, s\} : s \in S\} \end{cases} \tag{6.1}$$

where d_x denotes the degree of node x in G and $g, w \in \mathbb{R}^+$ are tuning parameters. If the resulting tree contains the root node r, r is removed from the tree, and the output is an ensemble of trees, a *forest*. The final output, the *interactome*, is the subnetwork in the PPI network induced by the nodes of this forest.

Selection of terminal nodes

Results from the differential expression analysis yielded 219 protein-coding genes that were associated with both aging and SARS-CoV-2 infection. Of particular interest among these genes were 181 genes that showed concordant regulation, i.e. they were either upregulated in both SARS-CoV-2 infection and aging or downregulated in both SARS-CoV-2 infection and aging. Intersecting the proteins corresponding to these 181 genes with proteins in the IREF interactome resulted in 162 proteins. These 162 proteins were selected as terminal nodes for the PCSF algorithm and prized according to their absolute $\log_2$-fold change between SARS-CoV-2-infected A549-ACE2 cells and normal A549-ACE2 cells (Supplementary Fig. E-10).

Parameter sensitivity analysis

Running the PCSF algorithm in the OmicsIntegrator2 required specifying three tuning parameters: g, w and b. In order to guarantee the robustness of the resulting network with respect to moderate changes in these parameters, we selected the parameters based on a sensitivity analysis.

The parameter g modifies the background PPI network by imposing an additive penalty on each edge based on the degrees of the corresponding vertices. It reduces the propensity of the algorithm to select hub nodes connecting many proteins in the interactome. While this feature may be relevant in certain biological applications, it was not necessarily the case in our work since high degree nodes may be of interest for the purpose of drug target identification. In the cost function in Equation (6.1), the absence of penalty corresponds to $g = -\infty$. However the OmicsIntegrator2 implementation only allows for $g \in \mathbb{R}^+$. In Supplementary Fig. E-11a1, we reported boxplots of penalized edge costs in the IREF interactome for different values of g. These boxplots suggest that the hub penalty parameter $g = 0$ yields similar edge costs to the desired setting where $g = -\infty$. For this reason we chose the value $g = 0$ in all OmicsIntegrator2 runs in this work.

The parameter w corresponds to the cost of edges connecting terminal nodes to the dummy root r. This parameter influences the number of trees in the Steiner forest. If w is chosen too low compared to the typical shortest path cost between two terminals, a trivial solution will connect all terminal nodes via r, leading to fully isolated terminals in the final forest. For high values of w the PCSF algorithm will not include the root r and output a connected network. Based on the histogram of the cost of the shortest path between any two terminals in the IREF interactome reported in Supplementary Fig. E-11a2, we ran a sensitivity analysis for w in the range $[0.2, 2]$.

The parameter b linearly inflates the prizes of terminal nodes in the objective function. Higher values of b result in more terminal nodes in the final PCSF. We analyzed edge costs in the network to determine a suitable range for b so as to include many terminal nodes in the resulting interactome. Supplementary Fig. E-11a1 shows that the maximum edge cost in the network for $g = 0$ was lower than 1, which meant that making b of order greater than 1 was necessary to ensure that trading off cost of edges added and prizes collected in the solution would rarely require discarding a terminal node. For this reason we ran a sensitivity analysis for b in the range $[5, 50]$.

Based on the previous considerations we fixed $g = 0$ and ran a sensitivity analysis as described in Supplementary Fig. E-11b with $w \in \{0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2\}$ and $b \in \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$. We obtained 100 PCSFs, each corresponding to a particular choice of (w, b). All of them included the entire terminal set S, a desired property resulting from the chosen range of the values of b. To analyze the robustness of the resulting networks to changes in the parameters, we analyzed the matrix $M \in [0, 1]^{100\times100}$ defined by

$$M_{ij} = \frac{\left|\left\{\substack{\text{nodes in}\\ \text{network } i}\right\} \cap \left\{\substack{\text{nodes in}\\ \text{network } j}\right\} \cap \mathcal{C}\right|}{\left|\left(\left\{\substack{\text{nodes in}\\ \text{network } i}\right\} \cup \left\{\substack{\text{nodes in}\\ \text{network } j}\right\}\right) \cap \mathcal{C}\right|}$$

for every pair of PCSFs i and j corresponding to parameters (w_i, b_i) and (w_j, b_j), respectively. Supplementary Fig. E-11c displays heatmaps of this matrix. We considered three different node sets $\mathcal{C}$, namely the set of all nodes in the input PPI network (Supplementary Fig. E-11c1), the subset of terminal nodes ($\mathcal{C} = S$, Supplementary Fig. E-11c2) and the subset of SARS-CoV-2 interaction partners (Supplementary Fig. E-11c3). Supplementary Fig. E-11c1, E-11c2, E-11c3 illustrate that choosing any $(w, b) \in [1.2, 2] \times [5, 50]$ led to the same connected PCSF with 252 nodes and 1,003 edges. This network is robust to moderate parameter changes for w and b. Collectively, this sensitivity analysis motivated the choice of $g = 0$, $w = 1.4$ and $b = 40$ used to obtain the interactome in Fig. 6-5b, where nodes are grouped by general function. The same interactome is presented in Supplementary Fig. E-12 with nodes grouped by general process. Note that since this interactome included all terminals and did not include the root node, it is equivalent to the solution of the classical Steiner tree problem.

Neighborhood analysis

For the interactomes obtained in this work, we reported 2-nearest-neighborhoods of genes of interest in Fig. 6-5c for the interactome of Fig. 6-5b, in Supplementary Fig. E-15 for the interactome of Supplementary Fig. E-14, and in Supplementary

Fig. E-16d for the interactome in Supplementary Fig. E-16c. Depending on the interactome, genes of interest include SARS-CoV-2 interaction partners (e.g. EXOSC5, FOXRED2, LOX, RBX1, RIPK1) as well as genes of potential therapeutic interest (e.g. HDAC1, EGFR). Neighborhood plots were enriched with information such as SARS-CoV-2 interaction partners and FDA approved, high affinity (based on data from DrugCentral) drugs with high correlation to the reverse SARS-CoV-2 infection signature. To improve legibility of the neighborhood networks, we discarded the highly connected hub node UBC (connected to 62% of proteins in the IREF network). To further improve legibility, we applied an upper threshold on edge cost (i.e., only visualizing high confidence edges) when the neighborhood networks were too densely connected. We generally chose this threshold at 0.53, with the exception of the LOX neighborhood (0.58) and the FOXRED2, ETFA and GNB1 neighborhoods (no thresholding). For each edge e in a given neighborhood, we defined the min-max scaled edge confidence $C(e)$ as

$$C(e) = \frac{\max_{e' \in \mathcal{E}} c(e') - c(e)}{\max_{e' \in \mathcal{E}} c(e') - \min_{e' \in \mathcal{E}} c(e')} \in [0, 1]$$

where $\mathcal{E}$ denotes the edge set of the corresponding interactome and $c(e)$ denotes the cost of edge e in the PPI network. This confidence metric was used to color edges in the neighborhood plots.

Addition of SARS-CoV-2 interaction partners to the terminal node list

In order to understand which other SARS-CoV-2 protein interaction partners were in the neighborhood of the identified interactome, we also ran the PCSF algorithm on the IREF PPI network using the SARS-CoV-2 and aging terminal list augmented with all known SARS-CoV-2 interaction partners. All SARS-CoV-2 interaction partners (with the exception of EXOSC5, FOXRED2 and LOX which were already present in the original terminal gene list) were given a small prize p. This prize was chosen by sensitivity analysis over a range of possible values from $p = 0$ (5 SARS-CoV-2 interaction partners initially selected by the method: EXOSC5, FOXRED2, LOX, RBXL1, RIPK1) to $p = 0.02$, beyond which all 332 known SARS-CoV-2 interaction partners belonged to the computed interactome. Fine-grained analysis revealed that choosing $p \in [4 \times 10^{-4}, 10^{-3}]$ leads to interactomes which include a stable set of 7 SARS-CoV-2 interaction partners, the 5 present initially plus CUL2 and HDAC2 (Supplementary Fig. E-13a). Supplementary Fig. E-13b-E-13c display heatmaps of the matrix $M \in [0,1]^{16\times 16}$ defined as

$$M_{ij} = \frac{\left|\left(\left\{\begin{smallmatrix}\text{nodes in}\\ \text{network } i\end{smallmatrix}\right\} \setminus \left\{\begin{smallmatrix}\text{nodes in}\\ \text{network } j\end{smallmatrix}\right\}\right) \cap \mathcal{C}\right|}{\left|\left\{\begin{smallmatrix}\text{nodes in}\\ \text{network } i\end{smallmatrix}\right\} \cap \mathcal{C}\right|}$$

for every pair of PCSFs i and j corresponding to parameters p_i and p_j, respectively. For the sensitivity analysis, we considered two different node sets $\mathcal{C}$, namely the set of all nodes in the input PPI network (Supplementary Fig. E-13b) as well as the subset of SARS-CoV-2 interaction partners (Supplementary Fig. E-13c). Supplementary Fig. E-13b shows that the obtained interactome was stable over the range $p \in [7 \times 10^{-4}, 10^{-3}]$. Supplementary Fig. E-13c shows that all SARS-CoV-2 interaction partners collected in the interactome when $p \in [7\times 10^{-4}, 10^{-3}]$ were also collected for higher values of p, which is a consequence of the observation from Supplementary Fig. E-13b. We used the value $p = 8 \times 10^{-4}$ for all subsequent analyses and figures, including Supplementary Fig. E-14 and Supplementary Fig. E-15.

Randomization and robustness analysis

We conducted several randomization assessments to understand the importance of each step in the pipeline, analyzing the impact of changes in the RNA-seq expression data, the underlying protein-protein interaction network, the CMap drug signatures, as well as the list of terminal genes on the final selection of drug targets and corresponding drugs. This was quantified by the frequency of appearance of each drug in the final drug list after 1000 randomization runs, for both drugs that were and that were not selected in the original non-randomized analysis. Results from this analysis suggest that the choice of terminal genes is the most critical step of the Steiner tree procedure; see below and Supplementary Table E.2.

(1) Randomization of PPI network: Randomization of the iREF protein-protein interaction network was performed via randomly permuting the vertex labels. Such randomization affects a gene's neighborhood while preserving basic network properties such as number of edges and degree distribution. The prize-collecting Steiner tree analysis pipeline was then applied to this new network. Drugs targeting terminal nodes were systematically selected in all randomization runs, as expected given that the prize-collecting Steiner tree algorithm parameters were set so that all terminal nodes are included in the solution. Other drugs identified by the non-randomized analysis that did not target any terminal node appeared with frequencies varying from 56% (primaquine, which has 5 targets in the network) to 97% (imatinib, which has 69 targets in the network). Only two drugs (mifepristone and palbociclib) that were not selected by the non-randomized analysis appeared more frequently (80% of runs) than the least frequently selected drug from the non-randomized analysis (primaquine, 56% of runs).

(2) Permuting expression data: Randomizing gene labels in the RNA-seq expression data set from (Blanco-Melo et al., 2020) while preserving gene labels of the GTEx aging data set is an implicit approach to randomizing the list of terminal genes used as input for the prize-collecting Steiner tree algorithm. After applying the Steiner tree analysis pipeline, the drugs selected in the non-randomized analysis appeared between 18% (milrinone) and 100% (sunitinib) of the runs. Generally, the more proteins a drug targeted in the iREF network, the more frequently it appeared in the solution (sunitinib, with 260 targets, is the drug with highest number of targets in the PPI network). 16 drugs that were not selected in the non-randomized analysis (this represents 1% of the set of non-selected drugs) appeared more frequently than the least frequently selected drug from the non-randomized analysis (milrinone).

(3) Randomization of CMap signatures: We also ran the Steiner tree analysis after randomly permuting the SARS-CoV-2-anticorrelation scores of the 605 CMap drugs and selecting the drugs with anticorrelation above 0.86 (resulting in 142 drugs as in the original non-randomized analysis). After applying the Steiner tree analysis pipeline, drugs that were selected in the non-randomized analysis appeared in the final list with a frequency between 22% and 26%, as expected (since 142/605 $\approx$ 23.5%). More interestingly, 17 drugs which were not selected in the non-randomized analysis (representing 1% of the overall set of non-selected drugs) appeared at a similar 22-29% frequency in the solution. These are drugs that target one of the network nodes yet have a true SARS-CoV-2-anticorrelation score lower than 0.86.

(4) Randomization of terminal nodes: Finally, we directly randomized the list of terminal nodes, by randomly selecting 162 genes from the RNA-seq expression dataset and prizing them with their corresponding absolute $\log_2$ fold change after SARS-CoV-2 infection in A549-ACE2 cells. The drugs selected in the non-randomized analysis appeared between 3% (milrinone) and 100% (sunitinib) of the runs. In this analysis, 41 drugs that were not selected in the non-randomized analysis (this represents 2.5% of the set of non-selected drugs) appeared more frequently than the least frequently selected drug from the non-randomized analysis (milrinone).

These results show that while the output of our Steiner tree analysis pipeline is quite robust to changes in the underlying PPI network, the selection of the terminal nodes has a critical effect on the final drug list.

To ensure robustness of our results to different ways of mitigating batch effects in the CMap dataset, we repeated the analysis by dropping all genes for which there was at least one sample containing a 1 in the expression value (reducing the total number of genes from 911 to 867 for the A549 cell line). As with the original batch correction approach, the resulting drugs consist mainly of protein kinase inhibitors (7 out of 9) and the drug targets are highly overlapping with the drug targets obtained from the original analysis (Supplementary Fig. E-23).

Single-cell RNA-seq analysis

Single-cell RNA-seq for A549 cells was obtained from GSE81861 (Li et al., 2017), where each entry in the matrix represents the gene expression (FPKM) of gene i in cell j. We preprocessed the data, keeping only genes that had a nonzero gene expression value in more than 10% of the cells, followed by $\log_2(x + 1)$ transformation of the data. Single-cell RNA-seq data for AT2 cells was obtained from http://www.nupulmonary.org/resources associated with (Reyfman et al., 2019). In order to avoid batch effects, we subset the data to include cells only from Donor 7 since that donor had the largest number of AT2 cells collected (4002 cells). We preprocessed the data using the same threshold as for A549 cells for filtering out genes across cells. Since single-cell RNA-seq data for AT2 cells was not yet normalized, we normalized the expression values across genes for each cell by the total RNA count for that cell, followed by $\log_2(x+1)$ transformation of the data as for A549 cells.

Evaluation of causal structure discovery algorithms

Prior to reporting the results of learning gene regulatory networks on A549 and AT2 cells, we benchmarked several causal structure discovery methods on the task of predicting the effects of interventions using gene knockout and overexpression data collected on A549 cells as part of the CMap project (Subramanian et al., 2017), similar to prior evaluations of causal methods (Wang et al., 2017; Yang et al., 2017b). We estimated the gene regulatory network underlying the identified interactome in A549 cells using the prominent causal structure discovery methods PC, GES and GSP (Spirtes et al., 2000; Glymour et al., 2019; Solus et al., 2017). Since not all edge directions are identifiable from purely observational data, these methods output a causal graph containing both directed and undirected edges. Since the advantage of causal networks is their ability to predict the effects of interventions on downstream genes, we evaluated these methods using interventions collected in CMap. In the following, we first describe how we estimated the effects of interventions based on the CMap data to use as ground truth for evaluating causal structure discovery methods. We focused our evaluation on genes and interventions that are shared between the combined SARS-CoV-2 and aging interactome and CMap knockout and overexpression experiments, resulting in 32 genes and 41 interventions (note that the number of interventions is larger than the number of genes, since in CMap interventions have been performed on genes that are not part of the L1000 landmark genes, but are contained in the interactome). We formed a matrix of genes by interventions, where each (i, j)-entry in the matrix represents the $\log_2$-fold change in expression of gene i when gene j was intervened on in comparison to the expression of gene i without intervention. We denoted by Q the binary matrix of intervention effects with $Q_{ij} = 1$ if the sign of the $\log_2$-fold change for the (i, j) entry was opposite for knockout and overexpression interventions to filter out unsuccessful interventions, the rational being that knockout and overexpression should have opposite downstream effects. Thus $Q_{ij} = 1$ denotes that perturbing gene j effects gene i and hence that gene i is downstream of gene j (Supplementary Fig. E-17a). Taking this matrix of interventional effects, Q, as the ground truth, we estimated the causal graph using the PC, GES and GSP algorithms and determined the corresponding ROC curve, counting and edge from $j \to i$ as a true positive if $Q_{ij} = 1$ and a false positive otherwise (Supplementary Fig. E-17b). In order to statistically evaluate whether the different algorithms performed better than random guessing, we sampled causal graphs (from an Erdös-Renyi model, where the edges were directed based on a uniformly sampled permutation) with different edge probabilities from the PPI network and calculated the corresponding number of true and false positives. For each false positive level, we created a distribution over true positives based on the sampled random causal graphs and calculated the p-value for the number of true positives obtained from the PC, GES and GSP algorithms. We combined the p-values across different numbers of false positives using Fisher's method and used this combined p-value for evaluating whether the PC, GES and GSP algorithms were significantly different from random guessing.

Causal structure discovery for learning gene regulatory networks

In order to learn the gene regulatory networks governing A549 and AT2 cells, we used the recent structure discovery method GSP (Solus et al., 2017; Wang et al., 2017; Yang et al., 2017b) on single-cell RNA-seq data from A549 cells as well as AT2 cells with the PPI network on 252 nodes as a prior. We used GSP since based on the previous analysis it outperformed the PC and GES algorithms in terms of ROC analysis on predicting the effect of gene knockout and overexpression experiments in A549 cells

(p-value = 0.0177 for GSP, p-value = 0.0694 for GSP and p-value = 0.5867 for GES); in addition, GSP is also preferable from a theoretical standpoint, since it is consistent under strictly weaker assumptions than the PC and GES algorithms (Solus et al., 2017). To obtain an estimate of the causal graph that is robust across hyperparameters and data subsampling, we used stability selection (Meinshausen and Bühlmann, 2010). In short, stability selection estimates the probability of selection of each edge by running GSP on subsamples of the data. Aggregating selection probabilities across algorithm hyperparameters (in this case the α-level for conditional independence testing), edges with high selection probability (0.3 for A549 cells and 0.4 for AT2 cells) were retained. The threshold for AT2 cells was chosen so as to approximately match the number of edges in the A549 network.

Appendix A

Appendix for Network analysis identifies chromosome intermingling regions as regulatory hotspots for transcription

Supplementary Figures

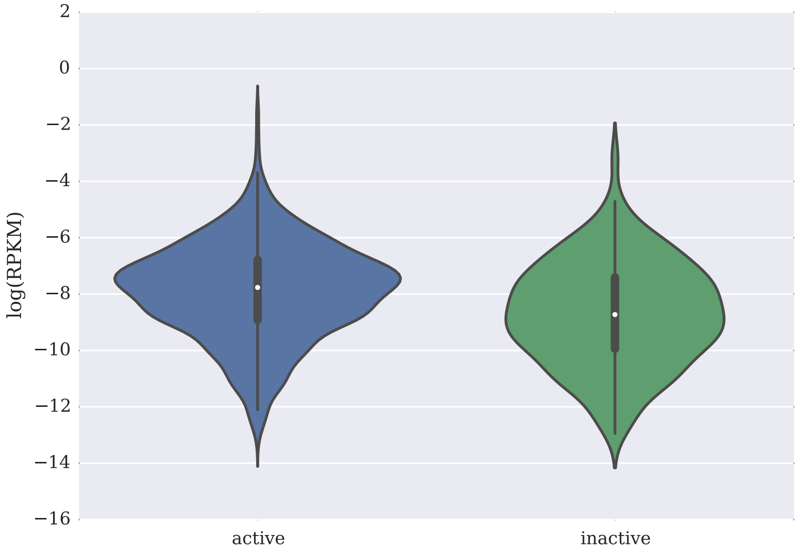


Figure A-1: Comparison of gene expression in reads per kilobase of transcript per million mapped reads (RPKM) on log scale between active and inactive clusters. Active clusters show significantly higher gene expression (p-value = 0.004 under t-test).

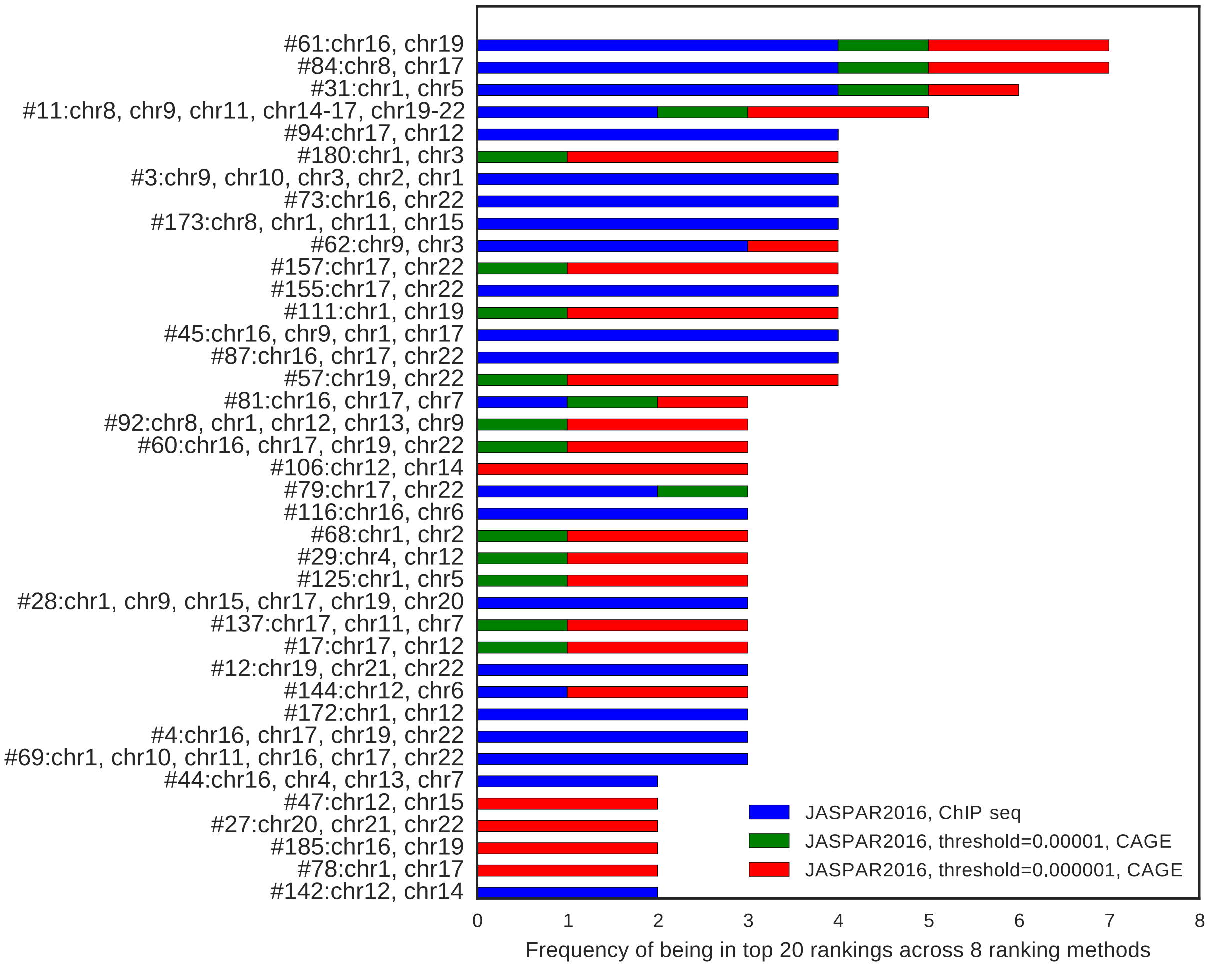


Figure A-2: Top ranked clusters that appeared in the top 20 clusters across 8 different methods of evaluation. The eight rankings are grouped by the different choices for filtering the JASPAR 2016 database: The JASPAR2016 database was filtered by ChIP-seq (blue) and the rankings were obtained i) using active clusters as background and by generating random matrices from the observed counts for the permutation test, ii) using active clusters as background and by generating random matrices based on the Dirichlet distribution for the permutation test, iii) using all intermingling regions as background and by generating random matrices from the observed counts for the permutation test, iv) using the whole genome as background and by generating random matrices from the observed counts for the permutation test. The JASPAR2016 database TFBS were obtained with a threshold of 0.00001 and filtered by CAGE (green) and the rankings were obtained v) using active clusters as background and by generating random matrices from the observed counts for the permutation test. The JASPAR2016 database TFBS were obtained with a threshold of 0.000001 and filtered by CAGE (red) and the rankings were obtained vi) using active clusters as background and by generating random matrices from the observed counts for the permutation test, vii) using active clusters as background and by generating random matrices from the Dirichlet distribution for the permutation test, and viii) using intermingling regions as background and by generating random matrices from the observed counts for the permutation test.

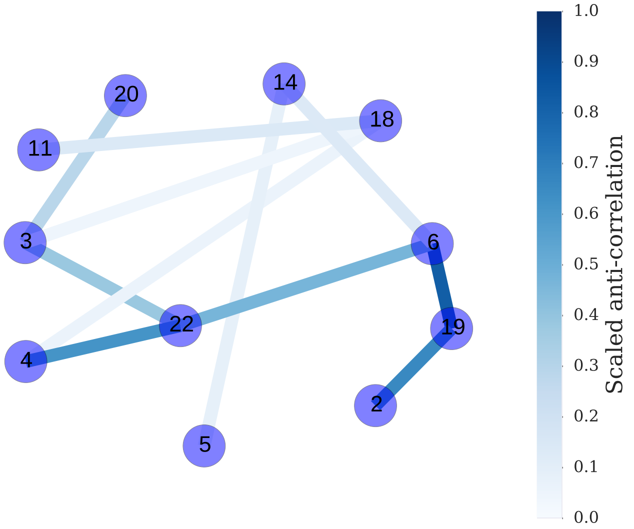


Figure A-3: Determining chromosomes that do not intermingle for predicting negative controls. Edges link nodes, i.e., pairs of chromosomes, that showed no intermingling regions in the LAS analysis. The edge weights are given by the absolute value of the anti-correlation between the genomic features of the adjacent chromosomes at a whole chromosome level.

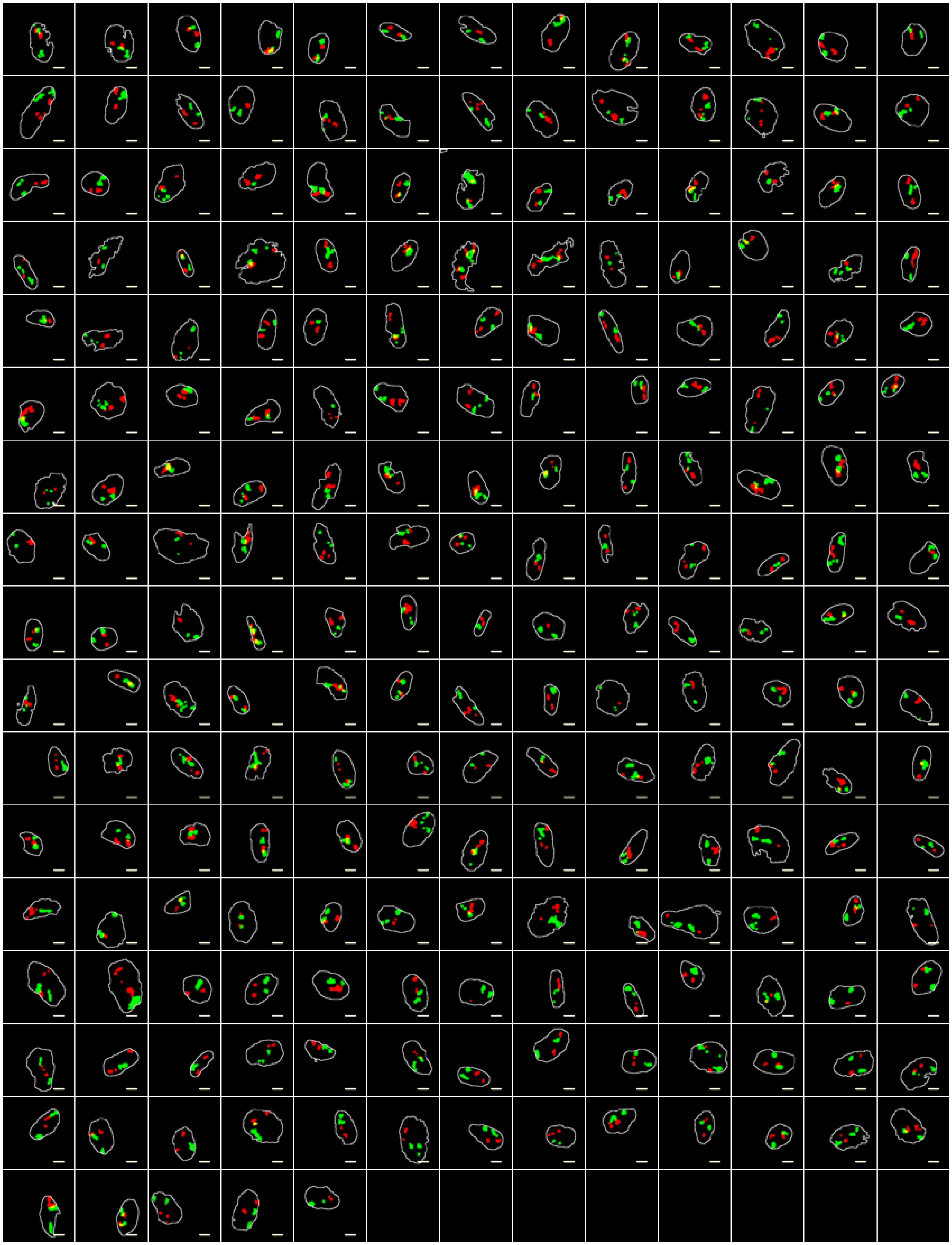


Figure A-4: Experimental validation using FISH for predicted active cluster on chromosomes 12 and 17. Segmented images of nuclei from a population of cells with nuclear boundary shown in white, chromosome 17 in red, and chromosome 12 in green. The scale bar has a length of 5.

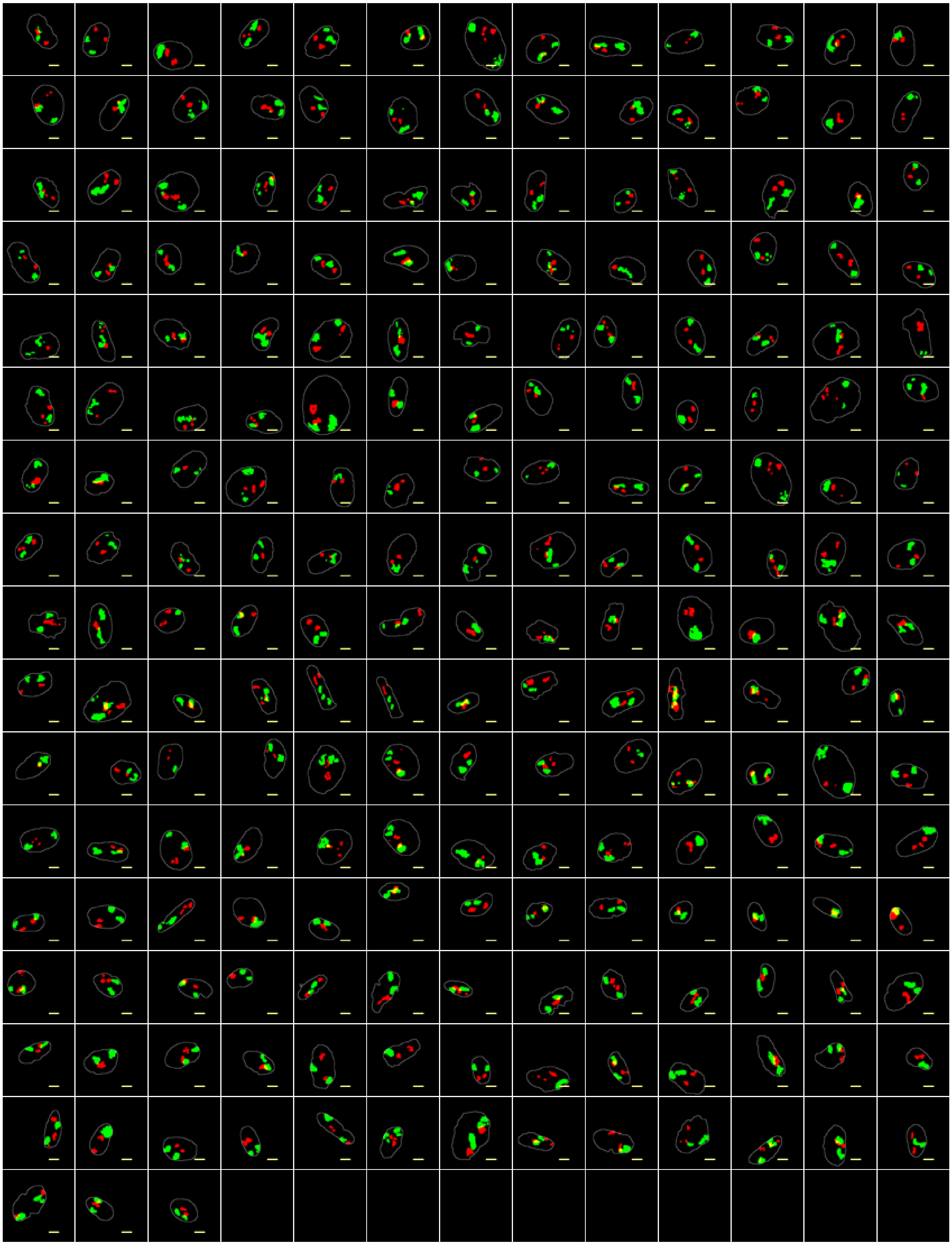


Figure A-5: Experimental validation using FISH for predicted negative control, i.e. chromosomes 3 and 20 that are predicted to not intermingle. Segmented images of nuclei from a population of cells with nuclear boundary shown in white, chromosome 20 in red, and chromosome 3 in green. The scale bar has a length of 5.

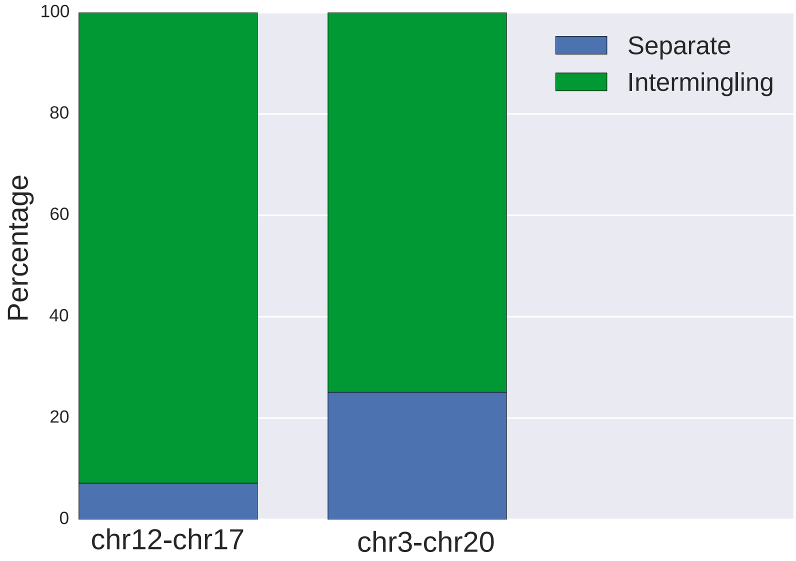


Figure A-6: Breakdown of the percentage of nuclei that intermingle (intermingling degree > 0) versus don't intermingle for the chromosome pairs 12-17 and 3-20, as found by FISH experiments.

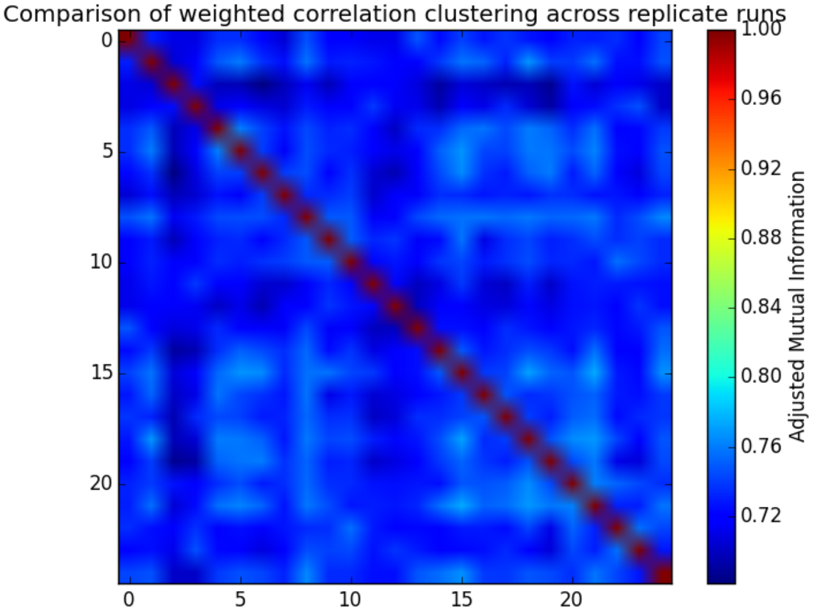


Figure A-7: Adjusted mutual information between replicate clusterings from weighted correlation clustering.

Supplementary Tables

Table A.1: Total sizes of intermingling regions

Region	Size
Intermingling domains from LAS	903.25 Mb
Intermingling regions after clustering	459.5 Mb
Intermingling regions in active clusters	179.75 Mb

Table A.2: Datasets and accessions used to obtain the genomic features. The symbol * indicates that peaks were retrieved from a previous study (Whalen et al., 2016) that had already pre-processed the data from the GEO database.

Name	Accession	Category
RNA-seq	GSE24565	active
RNAPII	GSE31477	active
H3K4me1	GSE16256	active
H3K4me2	GSE16256	active
H3K4me3	GSE16256	active
H3K36me3	GSE16256	active
H3K9ac	GSE16256	active
H3K27me3	GSE16256	repressive
H3K9me3	GSE16256	repressive
YAP1*	GSE61852	other
RFX5	GSE31477	other
RELA*	GSE43070	other
RCOR1	GSE31477	other
RBL2*	GSE19899	other
RB1*	GSE19899	other
RAD21	GSE31477	other
MXI1	GSE31477	other
MECP2*	GSE47678	other
MAZ	GSE31477	other

MAFK	GSE31477	other
LMNB1*	GSE53332	other
H4K91ac	GSE16256	other
H4K8ac	GSE16256	other
H4K5ac	GSE16256	other
H4K20me1	GSE16256	other
H3K9me1	GSE16256	other
H3K79me2	GSE16256	other
H3K79me1	GSE16256	other
H3K56ac	GSE16256	other
H3K4ac	GSE16256	other
H3K27ac	GSE16256	other
H3K23ac	GSE16256	other
H3K18ac	GSE16256	other
H3K14ac	GSE16256	other
H2BK5ac	GSE16256	other
H2BK20ac	GSE16256	other
H2BK15ac	GSE16256	other
H2BK12ac	GSE16256	other
H2BK120ac	GSE16256	other
H2A.Z	GSE16256	other
MacroH2A1.1	GSE54847	other
H2AK9ac	GSE16256	other
H2AK5ac	GSE16256	other
EP300*	GSE43070	other
DNase-seq	GSE18927	other
CTCF	GSE31477	other
CHD1	GSE31477	other
CEBPB	GSE31477	other

Table A.3: Fold enrichment of high-occupancy target (HOT) regions as compared to low-occupancy target regions (LOT) in active clusters.

Feature	Enrichment
HOT:LOT	2.94

Table A.4: Top 15 inactive clusters, ranked by enrichment for H3K9me3. Clusters are given by chromosome number and start position in kb; each region in the cluster is 250kb in length. Only clusters with less than seven 250kb regions in the cluster are included.

Cluster #	**Clusters (kb)**	**H3K9me3**	**RNAPII**
443	chr10:1500, chr8:2750	3.523	0.000
114	chr10:125000, chr1:5750, chr10:125250	3.377	0.000
204	chr8:137000, chr6:164500	3.353	0.000
213	chr17:11500, chr17:10750, chr5:6250, chr17:11000, chr17:11250	3.264	0.079
260	chr2:0, chr5:178250	3.225	0.431
370	chr22:26250, chr20:22000, chr20:21750	3.180	0.000
63	chr6:170250, chr1:13750	3.152	0.000
316	chr1:238250, chr6:162250	3.127	0.000
445	chr6:164750, chr5:166000	3.116	0.000
49	chr1:34750, chr1:34000, chr1:34250, chr8:142500, chr1:35000, chr1:34500	3.054	0.000
265	chr19:30750, chr20:19250	3.048	0.000
251	chr6:165250, chr8:138500	3.019	0.000
235	chr21:32250, chr20:15250	3.001	0.431
230	chr14:104500, chr14:104750, chr13:112000	2.994	0.000
434	chr1:239250, chr6:163000	2.992	0.000

Table A.5: Top 15 active clusters, ranked by p-value of permutation test based on TFBS (JASPAR 2016, threshold = 0.000001, CAGE). Clusters are given by chromosome number and start position in kb; each region in the cluster is 250kb in length.

Cluster #	**Clusters (kb)**	**P-value**	**RNAPII**	**H3K9me3**
137	chr7:100250, chr7:100750, chr11:65250, chr17:42750	0.0000	10.1162	0.5417
111	chr1:22250, chr19:45250	0.0000	4.3250	0.2820
61	chr16:30000, chr19:10750, chr16:29750, chr19:10500	0.0002	4.9110	0.3573
57	chr19:56000, chr19:55500, chr22:50250, chr19:55750	0.0001	8.1535	0.9214
157	chr22:50750, chr17:7250	0.0005	8.2122	0.3790
27	chr20:48750, chr20:48500, chr21:47250, chr22:30500	0.0021	13.5989	0.4189
29	chr12:53500, chr4:1000, chr12:53250	0.0024	5.0042	0.5440
144	chr12:50000, chr6:35250	0.0031	5.2105	0.3111
92	chr1:1000, chr12:123250, chr1:750, chr9:133500, chr8:145000, chr13:114750	0.0043	9.0570	0.5125
180	chr1:27750, chr3:46750	0.0068	1.0079	0.5889
84	chr17:73000, chr8:144250	0.0068	3.5725	0.5660
68	chr2:220250, chr1:16750, chr2:220000	0.0070	5.4663	0.4099
185	chr16:30750, chr19:17250, chr16:31000	0.0073	5.3819	0.3246
17	chr12:49250, chr17:38250, chr12:49500	0.0075	6.0798	0.2533
78	chr1:203250, chr17:74500	0.0103	7.9689	0.5743

Appendix B

Appendix for Multi-Domain Translation between Single-Cell Imaging and Sequencing Data using Autoencoders

Supplementary Methods

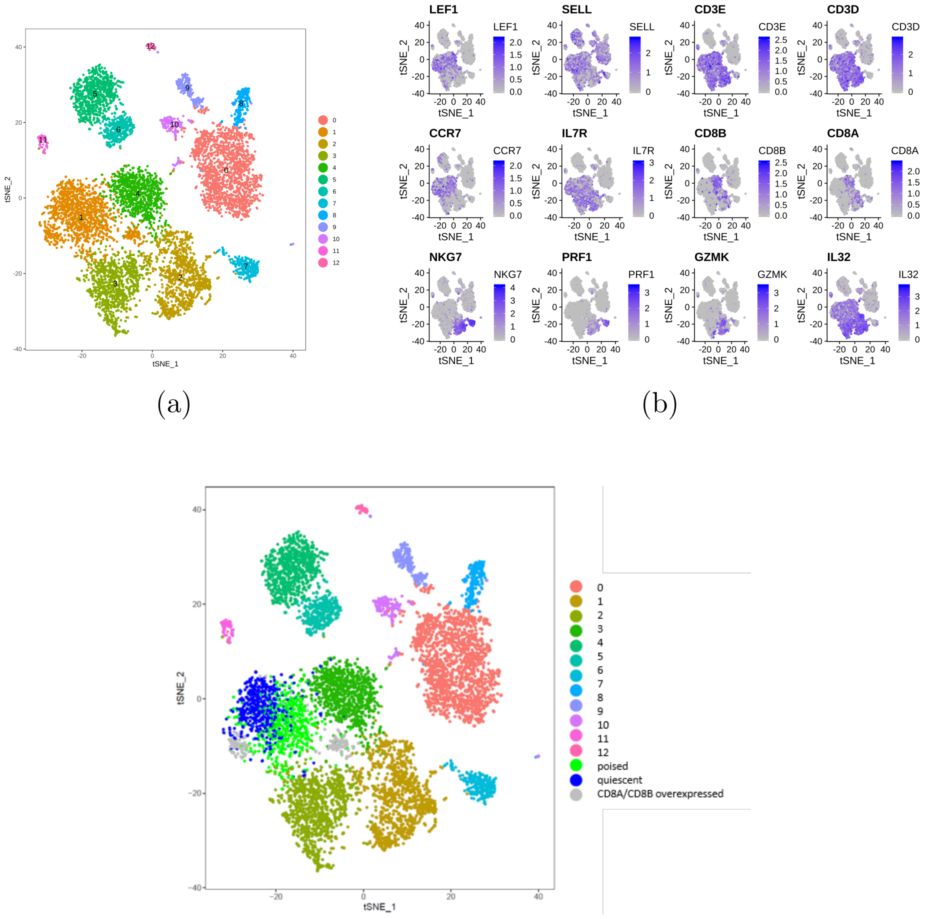
Autoencoder training on chromatin images for validation

We trained a convolutional autoencoder with the following architecture on the chromatin images: (1) We used 15 convolutional layers with 256 3×3 filters per layer followed by leaky ReLU activations throughout; (2) layers 2-6 have a stride size of 2 and layers 8-12 are followed by bilinear upsampling layers with a scale factor of 2. The bottleneck of our network thus provides a 256 dimensional representation of the images. We trained our network using the Adam optimizer (learning rate of 10^{-4}) and used a Kaiming uniform initialization for all our convolutional layers. All of the images were trimmed to remove background and resized to 32×32 images in order to remove nucleus size as a distinguishing feature. We held out 10% of the data as test data and trained until the reconstruction loss on the test data was smaller than

10^{-3}.

In order to determine whether our network was able to separate the poised and quiescent naive CD4+ T-cell clusters (as determined by the protein ratio of CORO1A to RPL10A) in an unsupervised fashion, we visualized the embedding of the images corresponding to the histogram peaks in Fig. 3-5 (namely the images with protein ratio in the range [.64, .7] and [0.93, 1]). Supplementary Fig. B-8 shows the resulting t-SNE embedding, where the color coding corresponds to the protein ratio of CORO1A to RPL10A. Interestingly, the latent embedding of the images obtained in an unsupervised fashion (with no information about the proteins) captures the protein ratio.

Supplementary Figures

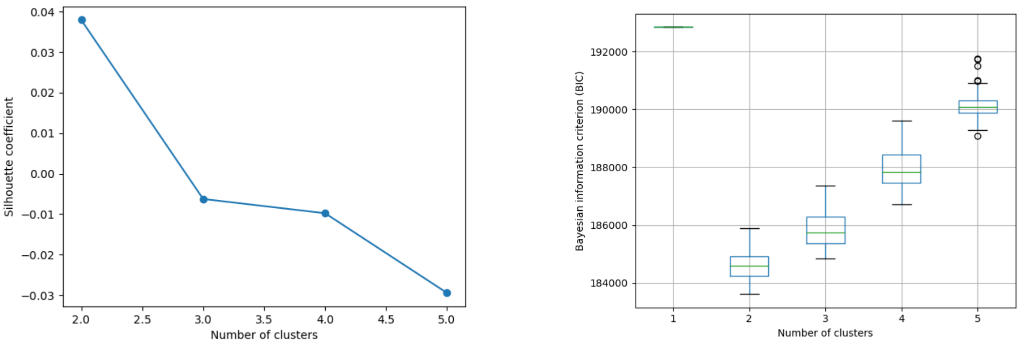


(a)

(b)

(c)

Figure B-1: Clustering of peripheral blood mononuclear cell (PBMC) data. (a) t-SNE of all cells in the PBMC data set, colored by inferred cluster label. (b) t-SNE plots of all cells in the PBMC data set, colored by expression of genes marking naive T-cell subpopulations (LEF1, SELL, CCR7), T-cells (CD3E, CD3D, IL7R, IL32), CD8 T-cells (CD8A, CD8B), natural killer, and cytotoxic T-cells (NKG7, PRF1, GZMK). (c) t-SNE plot including the clustering of the naive CD4+ T-cells (clusters denoted by different colors). Grey subpopulation differentially overexpresses CD8A and CD8B as the genes with highest average log-fold change (corrected p-value $= 1.30 \times 10^{-40}$ and 1.54×10^{-51} respectively), indicating that these cells are not naive CD4+ T-cells; thus they have been removed from further analysis.



(a) (b)

Figure B-2: Evaluating optimal number of clusters for naive CD4+ T-cell gene expression data ($n = 1166$ cells). (a) Silhouette coefficient for clusters obtained with Seurat at different resolutions (0.8, 0,9, 1.1, 1.15). (b) BIC score (averaged over 100 trials) for Gaussian mixture model with 1, 2, 3, 4 and 5 components. The boxplots illustrate the median (middle line), with box indicating the first and third quartiles and the whiskers indicating ± 1.5 × interquartile range. Outliers are plotted as separate dots.

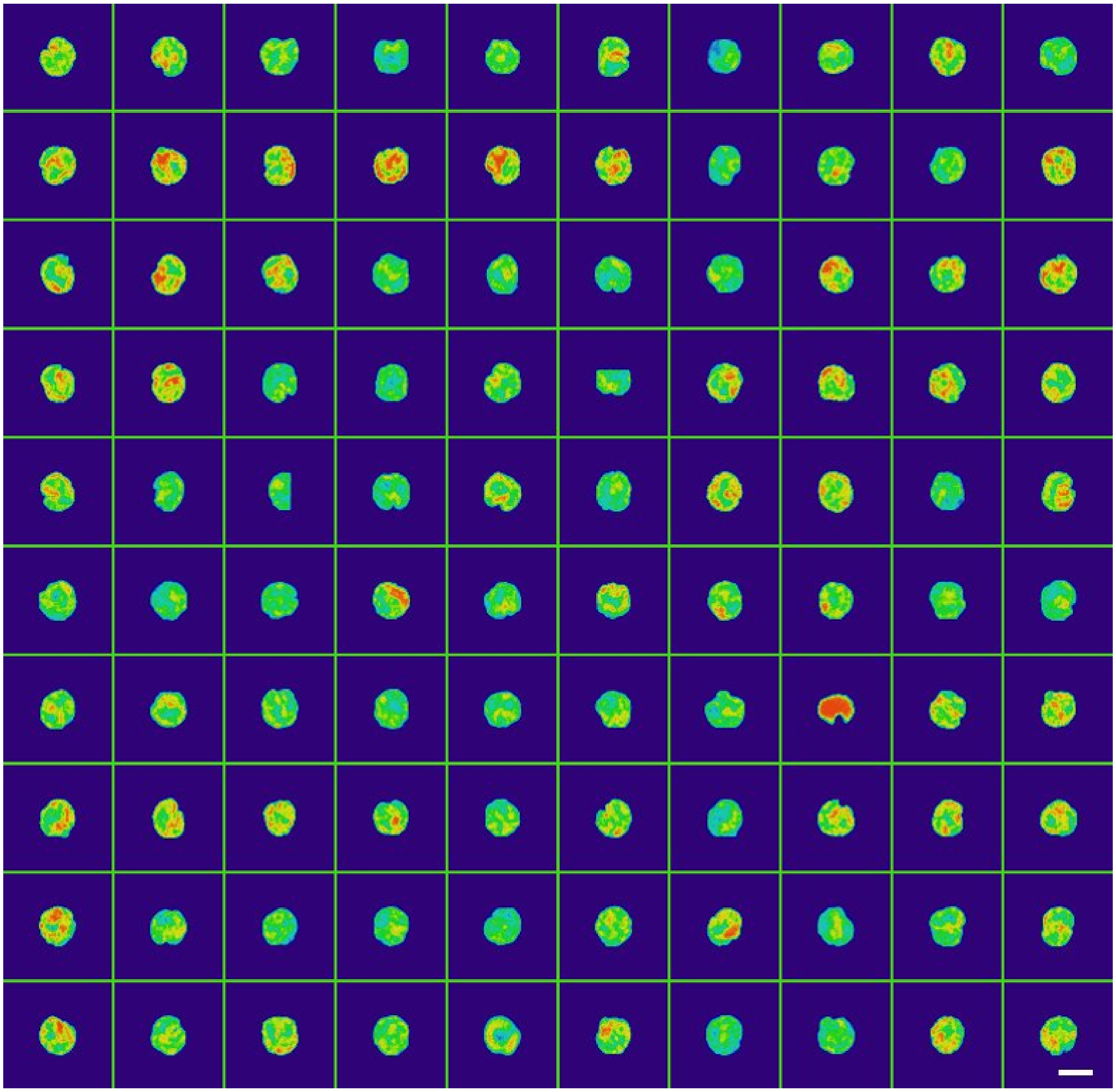
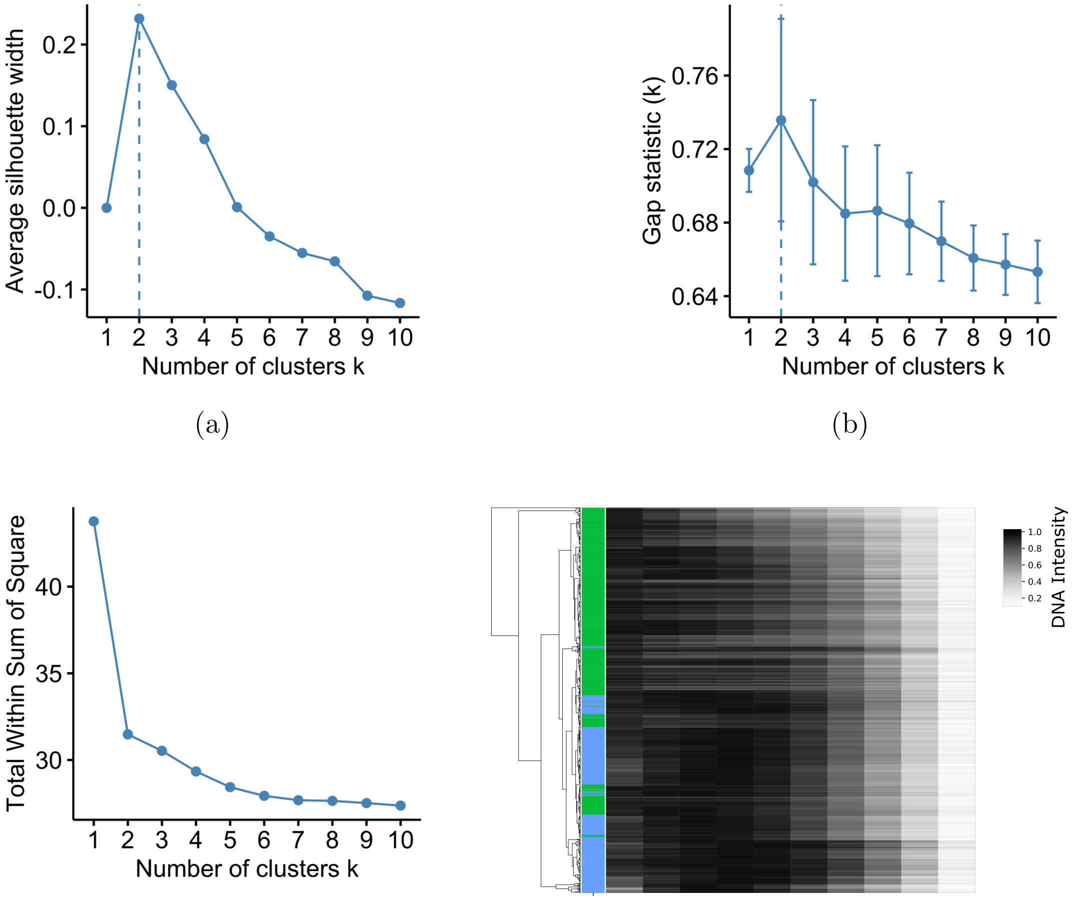


Figure B-3: Examples of naive CD4+ T-cell nuclei stained with DAPI. Scale bar is 2 microns. Images were selected randomly from 4 experiments.



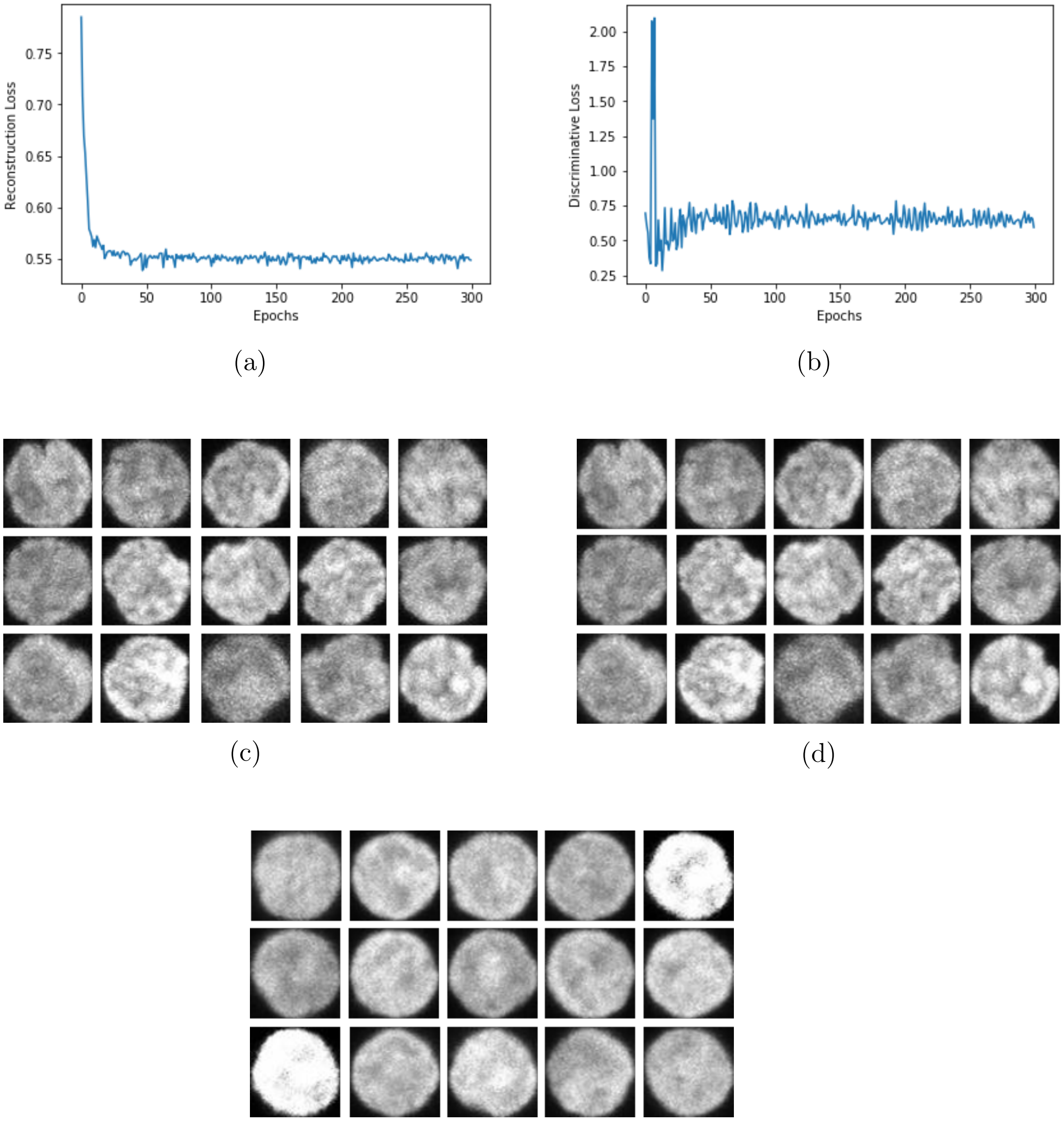
(a)

(b)

(c)

(d)

Figure B-4: Evaluating optimal number of clusters for naive CD4+ T-cell imaging data ($n = 729$ cells from two biologically independent replicates) using (a) average silhouette width, (b) gap statistic using 50 bootstrap samples, (c) total within-cluster sum of square, (d) alternative clustering using 1 - Pearson's correlation matrix with average linkage. Green and blue colors represent labels obtained based on the original clustering with 1 - Spearman's correlation and complete linkage, where green indicates the subpopulation of cells with central chromatin pattern and blue indicates the subpopulation of cells with peripheral chromatin pattern.



(a)

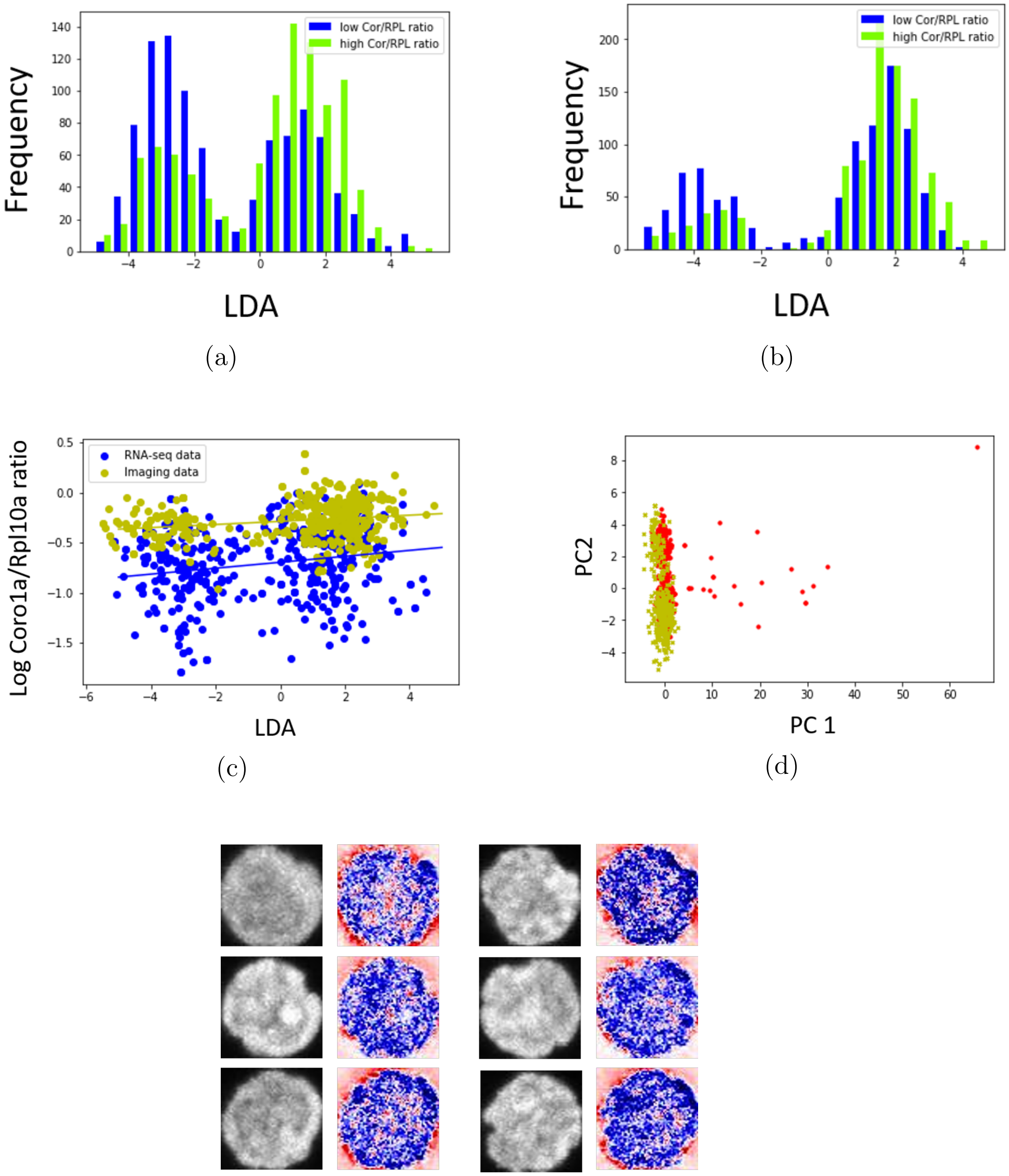
(b)

(c)

(d)

(e)

Figure B-5: Cross-modal autoencoder model trained on single-cell RNA-seq and single-cell images of DAPI-stained nuclei. (a) Reconstruction loss curve (sum of RNA-seq and image reconstruction losses). (b) Discriminative loss curve for RNA-seq and image translation model. (c) Examples of input images to the image autoencoder. (d) Reconstructed images after training the image autoencoder. (e) Generated images translated from RNA-seq to image space. Images were selected randomly.


Frequency
LDA
low Cor/RPL ratio
high Cor/RPL ratio
0
20
40
60
80
100
120
140
−4
−2
0
2
4
Frequency
LDA
low Cor/RPL ratio
high Cor/RPL ratio
0
50
100
150
200
−4
−2
0
2
4
Log Coro1a/Rpl10a ratio
LDA
RNA-seq data
Imaging data
0.5
0.0
−0.5
−1.0
−1.5
−6
−4
−2
0
2
4
PC2
PC 1
8
6
4
2
0
−2
−4
0
10
20
30
40
50
60

(a)

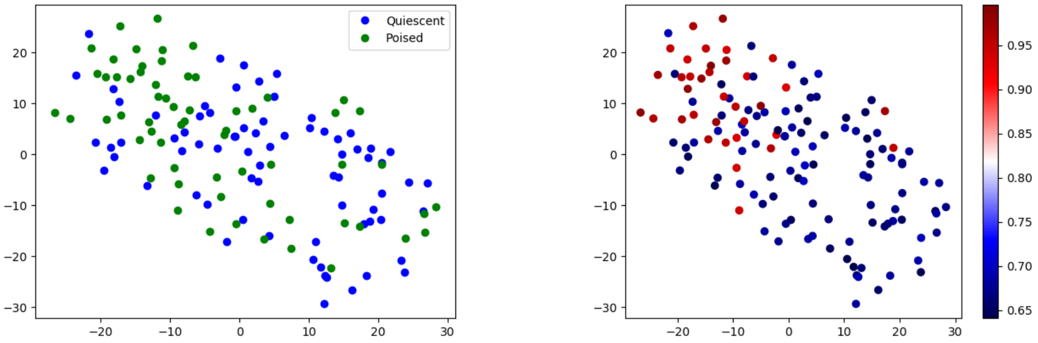
(b)

(c)

(d)

(e)

Figure B-7: Validation of inferred latent embedding. Histograms of embedded naive CD4+ T-cells from (a) RNA-seq and (b) imaging data sets, split by high (green) versus low (blue) CORO1A/RPL10A ratio. Histogram is computed along LDA axis that maximally separates two subpopulations in the latent space, showing that the axis aligns with CORO1A/RPL10A ratio. (c) Scatterplot of CORO1A/RPL10A ratio versus projection onto LDA axis. In both data sets, RNA-seq (blue) and imaging (yellow), the positive correlation between the ratio and the projection onto the LDA axis is statistically significant ($p = 8.27 \times 10^{-6}$ for RNA-seq data, $p = 2.27 \times 10^{-6}$ for imaging data, two-sided Wald test for linear fit). (d) RNA-seq (red) and imaging (yellow) data embedded in latent space, visualized using PCA. (e) Interpretation of image features along the LDA axis that maximally separates the two naive T-cell subpopulations in the latent space. Results show decreased background chromatin concentration in the nucleus.



(a) (b)

Figure B-8: t-SNE visualization of latent space learned by convolutional autoencoder on chromatin imaging data of naive CD4+ T-cells colored by (a) cluster label: quiescent (blue) and poised (green) naive CD4+ T-cells and (b) protein ratio of CORO1A to RPL10A. The autoencoder separates out the two naive CD4+ T-cell clusters by protein ratio without cluster supervision.

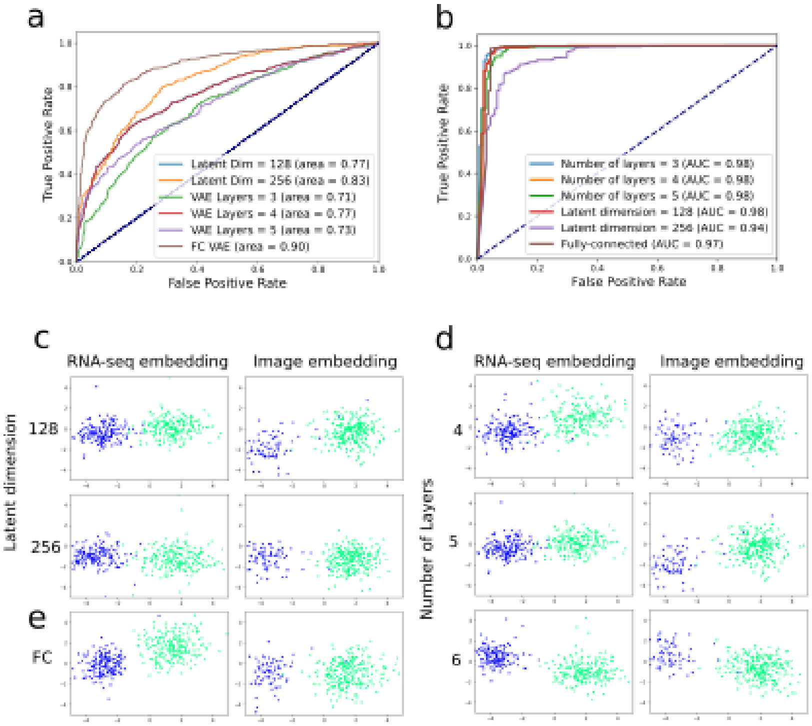


Figure B-9: Evaluation of robustness to the choice of architecture (fully-connected versus convolutional layers, number of layers, and latent space dimension) for the cross-modal autoencoder integrating RNA-seq and chromatin imaging. (a) Receiver Operating Characteristic (ROC) curve illustrating performance of classifiers trained to distinguish between peripheral and central chromatin patterns in images when evaluated on images translated from RNA-seq data. High performance of classifiers indicates that the alignment of the clusters in the latent space also holds in the original gene expression and imaging spaces and is robust to different architecture choices. The dotted dark blue line represents random guessing based on evenly-distributed classes and the remaining colors represent different model architectures. (b) ROC curves illustrating performance of classifiers trained to distinguish between quiescent and poised gene expression programs when evaluated on RNA-seq data translated from images. (c) Linear Discriminant Analysis (LDA) plots of single-cell RNA-seq (left) and imaging (right) datasets embedded in the latent space for models with different numbers of latent dimensions. The clusters with more quiescent (blue) and poised (green) gene expression programs from the RNA-seq dataset are aligned with the clusters with peripheral (blue) and central (green) chromatin patterns from the imaging dataset. (d) Same as (c), for models with different numbers of layers in the RNA-seq VAE. (e) Same as (c), for model with fully-connected image VAE. Note that the model with latent dimension of 128 is the same model as the one with 4 layers in the RNA-seq VAE.

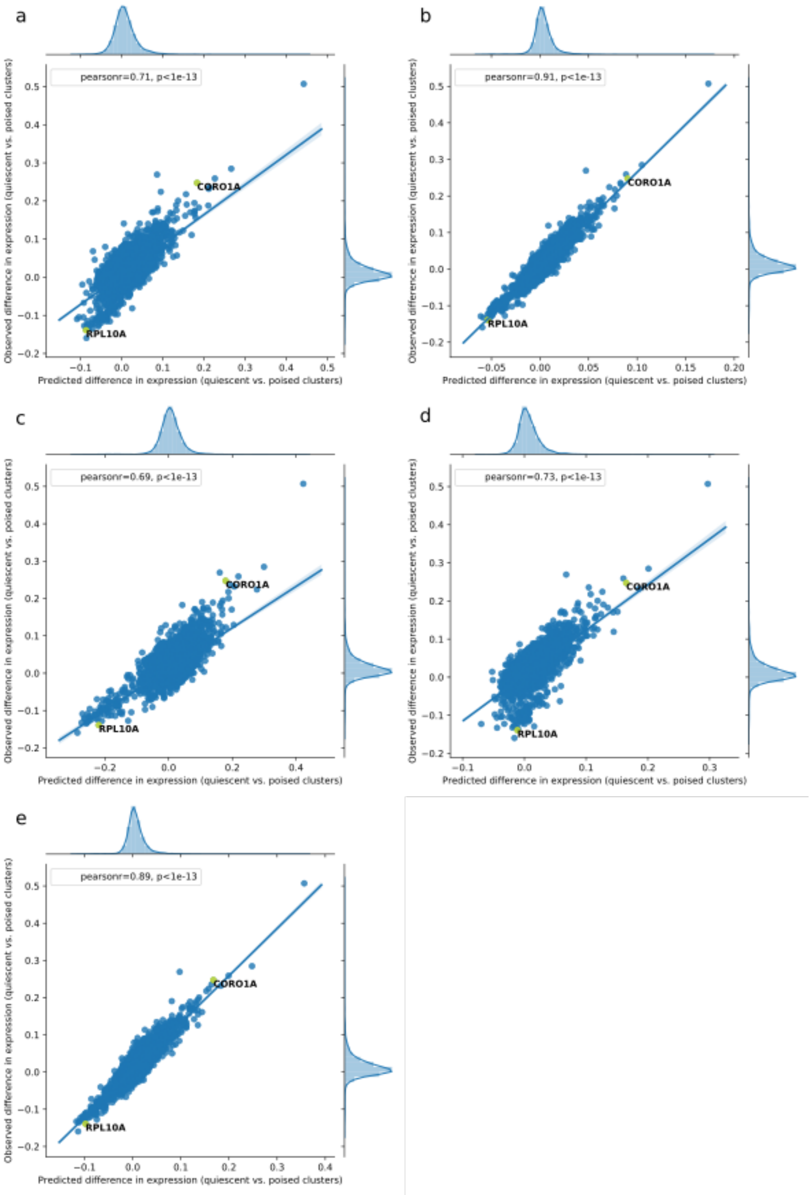


Figure B-10: Evaluation of robustness to the choice of architecture (fully-connected versus convolutional layers, number of layers, and latent space dimension) for the cross-modal autoencoder integrating RNA-seq and chromatin imaging. Differential gene expression analysis between cells with central and peripheral chromatin pattern performed on the predicted gene expression matrix translated from images using our methodology with different architecture choices. The predicted fold-change of gene expression based on images is strongly correlated with the observed fold-change of gene expression between quiescent and poised naive T-cells from the actual RNA-seq dataset. (a) Original model with 128 latent dimensions and 4 layers in the RNA-seq VAE, (b) model with 256 latent dimensions, (c) model with 3 layers in the RNA-seq VAE, (d) model with 5 layers in the RNA-seq VAE, (e) model with fully-connected image VAE instead of convolutional.

Supplementary Tables

	Input size	**Hidden layer size(s)**	**Output size**
Encoder A	815 (ATAC-seq TFs)	815, 815, 815, 100	50 (latent space)
Encoder B	2613 (RNA-seq genes)	2613, 2613, 2613, 100	50 (latent space)
Decoder A	50 (latent space)	100, 815, 815, 815	815 (ATAC-seq TFs)
Decoder B	50 (latent space)	100, 2613, 2613, 2613	2613 (RNA-seq genes)
Discriminator	50 (latent space)	50, 100	1
Classifier	50 (latent space)	N/A	3 (treatment time class probabilities)

Table B.1: Network architecture for autoencoder network trained on RNA-seq and ATAC-seq data collected from A549 cells. The discriminator, decoders, and encoders have leaky ReLU activations after each layer.

Loss description	**Type**	**Weight**
Reconstruction loss for ATAC-seq	Mean absolute error	10
Reconstruction loss for RNA-seq	Mean absolute error	10
Discriminative loss	Mean squared error	10
Shared cluster (treatment time) classification loss for ATAC-seq	Cross-entropy	10
Shared cluster (treatment time) classification loss for RNA-seq	Cross-entropy	10
Anchor/supervision loss between paired points in the latent space	Mean absolute error	0.1

Table B.2: Losses and corresponding weights for autoencoder network trained on RNA-seq and ATAC-seq data collected from A549 cells.

Cluster #	**Differentially overexpressed genes**	**Cluster annotation**
0	S100A8, S100A9, LYZ, S100A12, TYROBP, FCN1, FTL, CTSS, MNDA, CST3	
1	LDHB, CCR7, LEF1, RPL31, NOSIP, CD3E, RPS27, RPS6, SARAF, TCF7	Naive CD4+ T-cells
2	CCL5, NKG7, GZMK, GZMA, IL32, KLRB1, CST7, DUSP2, CMC1, CTSW	Cytotoxic T-cells
3	IL32, LTB, IL7R, ITGB1, KLRB1, LDHB, CD3D, CD2, AQP3, GSTK1	Activated CD4+ T-cells
4	CD8B, CD8A, JUNB, LDHB, LEF1, CCR7, NPM1, RPS6, CD7, SARAF	Naive CD8+ T-cells
5	TCL1A, CD79A, CD74, CD79B, MS4A1, HLA-DRA, HLA-DPA1, HLA-DQB1, HLA-DPB1, CD37	
6	CD79A, MS4A1, CD79B, CD74, JCHAIN, HLA-DRA, HLA-DPA1, HLA-DPB1, HLA-DQB1, BANK1	
7	GNLY, NKG7, PRF1, FGFBP2, GZMA, CTSW, KLRD1, GZMB, KLRF1, SPON2	
8	LST1, FCGR3A, AIF1, SAT1, FCER1G, COTL1, PSAP, MS4A7, FTL, IFITM3	
9	HLA-DQA1, HLA-DRB1, CST3, HLA-DPB1, HLA-DPA1, FCER1A, HLA-DRA, CD74, HLA-DQB1, LYZ	
10	PPBP, PF4, GNG11, HIST1H2AC, RGS18, TUBB1, TSC22D1, S100A9, S100A8, NRGN	
11	CD79A, CD79B, CD74, TCL1A, MS4A1, CD37, BANK1, HLA-DPA1, CD22, RALGPS2	
12	GZMB, JCHAIN, LILRA4, ITM2C, PTGDS, IRF7, IRF8, PLD4, PLAC8, CCDC50	

Table B.3: Top 10 differentially upregulated genes (average log-fold change > 0) for each cluster in PBMC data set.

	Image autoencoder	RNA-seq autoencoder
Encoder	2D Convolutional Block (1, 128, 4 x 4, 2)	Fully connected block (7633, 1024)
	2D Convolutional Block (128, 256, 4 x 4, 2)	Fully connected block (1024, 1024)
	2D Convolutional Block (256, 512, 4 x 4, 2)	Fully connected block (1024, 1024)
	2D Convolutional Block (512, 1024, 4 x 4, 2)	Fully connected block (1024, 1024)
	2D Convolutional Block (1024, 1024, 4 x 4, 2)	Fully connected block (1024, 1024)
	Fully connected (4096, 128)	Fully connected (1024, 128)
Decoder	Fully connected (128, 4096)	Fully connected (128, 1024)
	2D Transposed Convolutional Block (1024, 1024, 4 x 4, 2)	Fully connected block (1024, 1024)
	2D Transposed Convolutional Block (1024, 512, 4 x 4, 2)	Fully connected block (1024, 1024)
	2D Transposed Convolutional Block (512, 256, 4 x 4, 2)	Fully connected block (1024, 1024)
	2D Transposed Convolutional Block (256, 128, 4 x 4, 2)	Fully connected block (1024, 1024)
	2D Transposed Convolutional Block (128, 1, 4 x 4, 2)	Fully connected (1024, 7633)
	Sigmoid	

Table B.4: Network architecture for RNA-seq and image autoencoder networks. Each block consists of a batch normalization layer and ReLU nonlinearity. The discriminator has the same structure as the RNA-seq decoder with no batch normalization, 3 fully connected blocks and output dimension of 2.

Appendix C

Appendix for Identifying 3D Genome Organization in Diploid Organisms via Euclidean Distance Geometry

Proofs studying identifiability of 3D configuration from additional distance constraints between neighboring loci

For the proofs of Propositions 4.4.1 and 4.4.2 we recall from Theorem 4.3.1 that (x_i, y_i) and (x_i^*, y_i^*) are diametrically opposite points on the same sphere. Denote the i-th sphere by S_i and let it have center c_i and radius r_i. Then $\|c_i - x_i\| = r_i$ and $2c_i - x_i = y_i$. We recall Proposition 4.4.1 from Section 4.4 in Chapter 4.

Proposition 4.4.1. *For $n \geq 3$, there are unique points $x_1, \ldots, x_n, y_1, \ldots, y_n \in \mathbb{R}^2$ satisfying equations*

$$\begin{aligned} x_i + y_i = x_i^* + y_i^* \text{ and } \|x_i\|^2 + \|y_i\|^2 = \|x_i^*\|^2 + \|y_i^*\|^2 \text{ for } 1 \leq i \leq n, \\ \|x_i - x_{i+1}\| = \|x_i^* - x_{i+1}^*\| \text{ and } \|y_i - y_{i+1}\| = \|y_i^* - y_{i+1}^*\| \text{ for } 1 \leq i \leq n-1. \end{aligned} \tag{4.5}$$

Proof. We have $y_1 = 2c_1 - x_1$ and $y_2 = 2c_2 - x_2$. Plugging this into $\|y_1 - y_2\| = \|y_1^* - y_2^*\|$

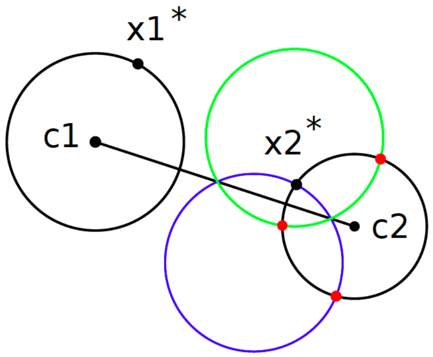


Figure C-1: Identifiability in the 2D setting with neighboring distance constraints. Two solutions for x_2 are obtained by translating the circle centered at c_1 by $x_2^* - x_1^*$ (this new circle is colored blue) and intersecting it with the circle centered at c_2. The other two solutions are obtained by reflecting the blue circle over the line through c_1 and c_2 (this new circle is colored green) and intersecting it with the circle centered at c_2. The true solution for x_2 is colored black and the three alternative solutions for x_2 are colored red.

gives

$$\begin{aligned}
\|y_1^* - y_2^*\| &= \|(2c_1 - x_1) - (2c_2 - x_2)\|^2 \\
&= \|(2c_1 - 2c_2) - (x_1 - x_2)\|^2 \\
&= \|2c_1 - 2c_2\|^2 + \|x_1 - x_2\|^2 - 2(2c_1 - 2c_2) \cdot (x_1 - x_2).
\end{aligned}$$

The quantities $\|2c_1 - 2c_2\|^2$ and $\|x_1 - x_2\|^2$ are fixed. This implies that the quantity $(2c_1 - 2c_2) \cdot (x_1 - x_2)$ is fixed. Since we know $\|x_1 - x_2\|$ and $c_1 \neq c_2$ holds by genericness, then there are two possible angles for $x_1 - x_2$ (this is where we use the 2D constraint) and thus there are two possible solutions for $x_1 - x_2$.

Because x_1, x_2 are constrained to lie on circles, the solutions for x_1 are the intersection points of the first circle and the second circle translated by $x_1 - x_2$ and the solutions for x_2 are the intersection points of the second circle and the first circle translated by $x_2 - x_1$. Hence each solution for $x_1 - x_2$ leads to at most two possible solutions for (x_1, x_2). In turn this implies there are at most four solutions for x_2.

We now investigate the four solutions. The first two solutions are obtained by translating the circle centered at c_1 by $x_2^* - x_1^*$ and intersecting it with the circle centered at c_2, see Figure C-1. One of the two solutions is x_2^*. The other two solutions are reflections of these two solutions over the line from c_1 to c_2.

Let x_1^*, x_2^*, c_1, c_2 be fixed. They determine four possible solutions for x_2. We will show that these four solutions are different from the four solutions we get from considering x_2^*, x_3^*, c_2, c_3 for generic x_3^*, c_3 (apart from x_2^*).

If either of the reflected solutions over the line from c_2 to c_3 coincides with one of the four original solutions, then we can perturb c_3 away from the line from c_2 to c_3 to change these solutions. If the solution that is the intersection point of the circle centered at c_2 and the translation by $x_2^* - x_3^*$ of the circle centered at c_3 (different from x_2^*) coincides with one of the four original solutions, then we can perturb x_3^*. This changes $x_2^* - x_3^*$ and hence the second intersection point of the circle centered at c_2 and the translation by $x_2^* - x_3^*$ of the circle centered at c_3.

A similar argument can be used to show that $x_3, \ldots, x_{n-1}$ have unique solutions. Given a unique solution for x_2, there are two solutions for x_1 if and only if x_2^* lies on the line from c_1 to c_2. This is however not a generic configuration. A similar argument applies for x_n. □

We recall Proposition 4.4.2 from Section 4.4 in Chapter 4.

Proposition 4.4.2. *For any $n \in \mathbb{N}$, there exist $x_1^*, x_2^*, \ldots, x_n^*$ and $y_1^*, y_2^*, \ldots, y_n^*$ such that there are infinitely many points $x_1, \ldots, x_n, y_1, \ldots, y_n \in \mathbb{R}^3$ satisfying equations (4.5).*

Proof. If $n = 1$, then x_1^*, y_1^* can be chosen randomly with the constraint that $x_1^* \neq y_1^*$. Then x_1 and y_1 can be any points on the sphere S_1 defined by x_1^*, y_1^*. Assume $n \geq 2$. Fix any x_1^*, y_1^* such that $x_1^* \neq y_1^*$. Choose two circles C_1 and C_1' on the sphere S_1 defined by x_1^*, y_1^* that intersect at two points one of which is x_1^*. The circle C_1 is the intersection of S_1 and another sphere T_1. Let x_2^* be the center of the sphere T_1. Let C_1'' be the circle on S_1 that consists of points antipodal to C_1'. Then C_1'' is also an intersection of S_1 and another sphere T_2''. Let y_2^* be the center of the sphere T_2''. We use the same procedure to construct x_3^* and y_3^* from x_2^* and y_2^*, x_4^* and y_4^* from x_3^* and y_3^* etc.

Now consider points x_n and y_n in an ε-neighborhood of x_n^* and y_n^*. Consider the spheres that are centered at x_n and y_n and have radii $\|x_{n-1}^* - x_n^*\|$ and $\|y_{n-1}^* - y_n^*\|$. The intersections of these spheres with S_{n-1} give circles $\tilde{C}_{n-1}$ and $\tilde{C}''_{n-1}$ that are perturbations of circles C_{n-1} and C''_{n-1}. In particular, the intersection of the circle $\tilde{C}_{n-1}$ and the circle $\tilde{C}'_{n-1}$ that consists of points antipodal to $\tilde{C}''_{n-1}$ consists of two points for ε small enough. Choosing x_{n-1} to be the intersection point corresponding to x_{n-1}^* and y_{n-1} its antipodal gives points x_{n-1}, y_{n-1} satisfying $\|x_{n-1}-x_n\| = \|x_{n-1}^* - x_n^*\|$ and $\|y_{n-1} - y_n\| = \|y_{n-1}^* - y_n^*\|$.
Assuming that ε is small enough, x_{n-1} and y_{n-1} are in small neighborhoods of x_{n-1}^* and y_{n-1}^*, and we can continue the same procedure to find x_{n-2} and y_{n-2} from x_{n-1} and y_{n-1}, x_{n-3} and y_{n-3} from x_{n-2} and y_{n-2} etc. In particular, we can find $x_1, \ldots, x_{n-1}, y_1, \ldots, y_{n-1}$ satisfying equations (4.5) for every x_n and y_n in an ε-neighborhood of x_n^* and y_n^*. □

The previous proposition suggests that there are two degrees of freedom for choosing $x_1, \ldots, x_n, y_1, \ldots, y_n$ on each homologous pair and thus that finite identifiability requires two additional algebraically independent constraints per homologous pair. Similarly this suggests that unique identifiability requires three additional algebraically independent constraints per homologous pair, where each endpoint of a chromosome needs to be included in at least one of the additional constraints.

Proofs for identifiability from higher-order contact constraints

We recall Theorem 4.5.1 from Section 4.5 in Chapter 4.

Theorem 4.5.1. *Let m be the number of chromosome pairs, let $n_1, n_2, \ldots, n_m$ be the number of domains on chromosomes $1, 2, \ldots, m$ and define $n = n_1 + n_2 + \ldots + n_m$. Let $I \subseteq [n] \times [n] \times [n]$ be such that each of $1, n_1, n_1+1, n_1+n_2, \ldots, n_1+n_2+\ldots+n_{m-1}+1, n$ (labels of domains at the beginning and at the end of each chromosome) is contained* *in at least one triple in I. Let $x_1^*, \dots, x_n^*, y_1^*, \dots, y_n^* \in \mathbb{R}^3$ be fixed such that*

$$\min_{z_i^* \in \{x_i^*, y_i^*\} \text{ for } i=k_1,k_2,k_3} \left(\sum_{j \in \{k_1,k_2,k_3\}} \|z_j^* - (z_{k_1}^* + z_{k_2}^* + z_{k_3}^*)/3\|^2 \right) = 0 \text{ for } (k_1,k_2,k_3) \in I.$$

Consider the polynomial system:

$$\begin{gathered} x_i + y_i = x_i^* + y_i^* \text{ and } \|x_i\|^2 + \|y_i\|^2 = \|x_i^*\|^2 + \|y_i^*\|^2 \text{ for } 1 \leq i \leq n, \\ \|x_i - x_{i+1}\| = \|x_i^* - x_{i+1}^*\| \text{ and } \|y_i - y_{i+1}\| = \|y_i^* - y_{i+1}^*\| \text{ for } i \in [n] \backslash \{n_1, n_1+n_2, \dots, n\}, \\ \min_{z_i \in \{x_i, y_i\} \text{ for } i=k_1,k_2,k_3} \left(\sum_{j \in \{k_1,k_2,k_3\}} \|z_j - (z_{k_1} + z_{k_2} + z_{k_3})/3\|^2 \right) = 0 \text{ for } (k_1,k_2,k_3) \in I. \end{gathered} \tag{4.6}$$

For generic $x_1^*, \dots, x_n^*, y_1^*, \dots, y_n^*$, this system has a unique solution in $(\mathbb{R}^3)^{2n}$.

Before we can prove Theorem 4.5.1, we will need two lemmas. Lemma C.0.1 states that for a fixed solution (x_1^*, y_1^*) on a sphere S_1 and given distances between solutions on S_1 and S_2, there are finitely many solutions (x_2, y_2) on the sphere S_2. Lemma C.0.2 is an extension of Lemma C.0.1. It states that if one has finitely many solutions on a sphere S_i, then given distances between neighboring beads, there are finitely many solutions on any sphere connected to S_i.

Lemma C.0.1. *Let $x_1^*, x_2^*, y_1^*, y_2^* \in \mathbb{R}^3$ be fixed. Consider the polynomial system:*

$$\begin{gathered} x_2 + y_2 = x_2^* + y_2^*, \|x_2\|^2 + \|y_2\|^2 = \|x_2^*\|^2 + \|y_2^*\|^2, \\ \|x_1^* - x_2\| = \|x_1^* - x_2^*\| \text{ and } \|y_1^* - y_2\| = \|y_1^* - y_2^*\|. \end{gathered} \tag{C.1}$$

For generic $x_1^*, x_2^*, y_1^*, y_2^*$, this system has finitely many solutions in $(\mathbb{R}^3)^{2n}$.

Proof. The first two equations of (C.1) say that x_2, y_2 and x_2^*, y_2^* are pairs of antipodal points on the same sphere. We denote this sphere by S_2. The third equation says that x_2 is the same distance from x_1^* as x_2^* is from x_1^*. Hence x_2 must lie on the circle C_{x_2} that is the intersection of S_2 and the sphere centered at x_1^* and with radius $\|x_1^* - x_2^*\|$. The last equation says that y_2 must lie on the circle C_{y_2} that is the intersection of S_2 and the sphere centered at y_1^* with radius $\|y_1^* - y_2^*\|$. We consider the circle C'_{x_2}

that consists of antipodal points to the circle C_{y_2} on the sphere S_2. The intersection of the circles C_{x_2} and C'_{x_2} gives the solutions for x_2. Unless the two circles are equal, they intersect at at most two points. Since y_2 is antipodal to x_2, then for each x_2 there is a unique y_2. The circles coincide if and only if x_1^*, y_1^* and the center of S_2 are collinear. □

Lemma C.0.2. *Let $x_1^*, \ldots, x_n^*, y_1^*, \ldots, y_n^* \in \mathbb{R}^3$ be fixed. Consider the polynomial system:*

$$x_i + y_i = x_i^* + y_i^* \text{ and } \|x_i\|^2 + \|y_i\|^2 = \|x_i^*\|^2 + \|y_i^*\|^2 \text{ for } 2 \leq i \leq n,$$
$$\|x_1^* - x_2\| = \|x_1^* - x_2^*\|, \|y_1^* - y_2\| = \|y_1^* - y_2^*\|,$$
$$\|x_i - x_{i+1}\| = \|x_i^* - x_{i+1}^*\| \text{ and } \|y_i - y_{i+1}\| = \|y_i^* - y_{i+1}^*\| \text{ for } 2 \leq i \leq n-1.$$

For generic $x_1^*, \ldots, x_n^*, y_1^*, \ldots, y_n^*$, this system has finitely many solutions in $(\mathbb{R}^3)^{2n-2}$.

Proof. By Lemma C.0.1, there are finitely many antipodal pairs $(x_2, y_2) \in \mathbb{R}^3 \times \mathbb{R}^3$ on S_2 such that $\|x_1^* - x_2\| = \|x_1^* - x_2^*\|$ and $\|y_1^* - y_2\| = \|y_1^* - y_2^*\|$. Similarly, for each of these antipodal pairs $(x_2, y_2) \in \mathbb{R}^3 \times \mathbb{R}^3$ on S_2, there are finitely many antipodal pairs $(x_3, y_3) \in \mathbb{R}^3 \times \mathbb{R}^3$ on S_3 satisfying $\|x_2 - x_3\| = \|x_2^* - x_3^*\|$ and $\|y_2 - y_3\| = \|y_2^* - y_3^*\|$ etc. □

Proof of Theorem 4.5.1. We recall that the first line of the polynomial system (4.6) gives that x_i, y_i are antipodal points on a sphere S_i. Consider a triple $(k_1, k_2, k_3) \in I$ that contains 1 and the equation on the last line of the polynomial system (4.6) corresponding to this triple. This equation gives that $z_{k_1}, z_{k_2}, z_{k_3}$, where $z_i \in \{x_i, y_i\}$, coincide. Hence $z_{k_1}, z_{k_2}, z_{k_3}$ lie on the intersection of $S_{k_1}, S_{k_2}, S_{k_3}$. Generically, if the intersection of three spheres is non-empty in $\mathbb{R}^3$, then it consists of two points P and P'. This gives four possible solutions for x_1, y_1: the points P, P' and their antipodals on S_1. By Lemma C.0.2, there are finitely many solutions for $x_2, \ldots, x_{n_1}, y_2, \ldots, y_{n_1}$ given these fixed solutions x_1, y_1 on S_1. In the next two paragraphs we will show that generically these finitely many solutions do not contain antipodal points on any of the spheres $S_2, \ldots, S_{n_1}$.

If there are two antipodal solutions on S_i, then we may assume that they come either from the same solution on S_1 or antipodal solutions on S_1, because we can perturb $S_{k_1}, S_{k_2}, S_{k_3}$ slightly to change the other pair of solutions. First we will show that generically a solution for x_i on S_i does not give a pair of antipodal solutions for x_{i+1} on S_{i+1}. If this was the case, then both the solution for x_i and its antipodal would have to lie on the plane that is perpendicular to the line through the antipodal pair of solutions for x_{i+1} on S_{i+1}. This plane contains the centers of S_i and S_{i+1}. Hence for a solution for x_i, there is only one antipodal pair on solutions on S_{i+1}. Thus for generic distance between the solutions on S_i and S_{i+1}, a solution on S_i does not give an antipodal pair of solutions on S_{n+1}.

Secondly, suppose that two different solutions on S_i give a pair of antipodal solutions on S_{i+1}. We will show that when we perturb the distance between solutions on S_i and S_{i+1}, then we do not get an antipodal pair anymore. Let x_i and x_i' be two different solutions on S_i that give solutions x_{i+1} and $2c_{i+1} - x_{i+1}$ on S_{i+1}. Hence $\|2c_{i+1} - x_{i+1} - x_i'\|^2 = \|x_{i+1} - x_i\|^2$. We want to show that generically

$$\|2c_{i+1} - (x_{i+1} + \epsilon) - x_i'\|^2 \neq \|x_{i+1} + \epsilon - x_i\|^2,$$

where $x_{i+1} + \epsilon$ is the perturbed solution. Indeed, using the identity $\|x_{i+1} - x_i\|^2 = \|2c_{i+1} - x_{i+1} - x_i'\|^2$ gives

$$\|2c_{i+1} - (x_{i+1} + \epsilon) - x_i'\|^2 - \|x_{i+1} + \epsilon - x_i\|^2 = 2\epsilon(x_i + x_i' - 2c_{i+1}).$$

This quantity is equal to zero if and only if $\epsilon = 0$ or c_{i+1} is the middle point of the line segment from x_i to x_i'. This is generically not the case.

Using a triple $(k_1', k_2', k_3') \in I$ containing n_1 and the equation for this triple, we get four possible solutions for x_{n_1}, y_{n_1}. Generically, only one of them coincides with the finitely many solutions on S_{n_1} that we get from the solutions on S_1, because perturbing the spheres slightly (with keeping the coinciding points fixed) perturbs the second intersection point of the three spheres and we know that generically the finitely many points do not contain antipodal points.

The unique solution on S_n comes from one solution on each of the spheres $S_1, \ldots, S_{n_1-1}$: If this was not the case then two different solutions on S_i give the same solution on S_{i+1}. By the proof of Proposition 4.4.1, the dot product $(c_i - c_{i+1}) \cdot (x_i - x_{i+1})$ is fixed. Hence for a fixed x_{i+1}, all possible solutions for x_i lie on a hyperplane and this hyperplane is perpendicular to $c_i - c_{i+1}$. Therefore if two solutions on S_i give the same solution on S_{i+1}, then they lie on a hyperplane perpendicular to $c_i - c_{i+1}$. By slightly perturbing the sphere S_{i+1}, this is not the case anymore, and hence generically a solution on S_{i+1} comes from a unique solution on S_i. □

Simulations

Reconstructions in the noiseless setting

Figure C-2 shows additional reconstructions of simulated data in the noiseless setting. The true structures are consistently recovered under different data generation models.

Impact of the number of tensor constraints

Figure C-3 shows the impact of the number of tensor constraints on the solution in the noisy setting. We explored the impact of the number of tensor constraints specifically when the number of chromosomes is higher (three chromosomes) since tensor constraints seem to play a more critical role in that setting, as shown in Figure 4-5. We evaluate the performance when 500, 1000 or all (4060) tensor constraints are used. Figure C-3 shows that the choice of the number of tensor constraints has little impact on the accuracy of reconstruction, so we used 1000 tensor constraints (or all possible triplets if that number was smaller) in simulations and in real data analysis.

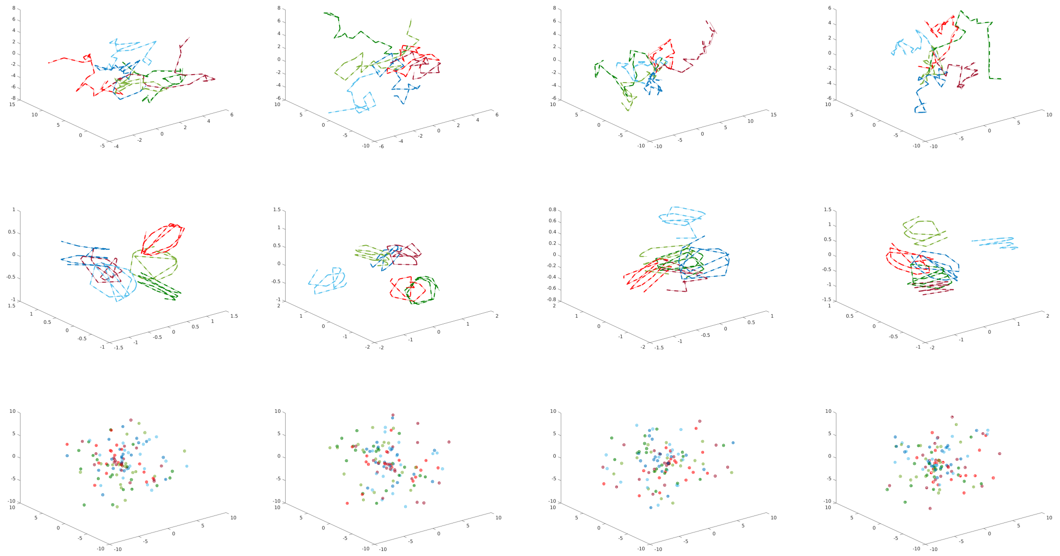
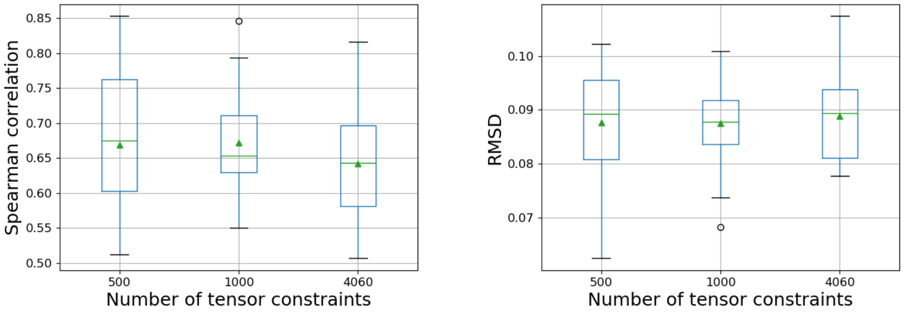


Figure C-2: Additional examples of true and reconstructed points on simulated data. True points were generated using Brownian motion model (first row), spirals (second row) and random points in a sphere (third row). We generate six chromosomes, corresponding to three homologous pairs with 20 domains per chromosome in the noiseless setting. Solid lines / points correspond to true 3D coordinates and dashed lines / unfilled points to reconstructions via our method. Each color represents a different chromosome.



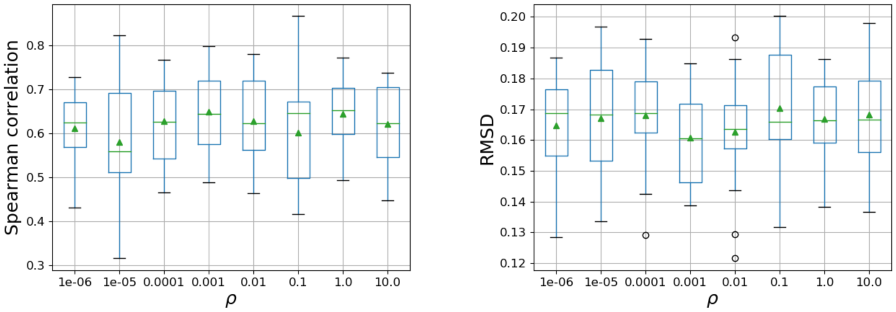
(a)

(b)

Figure C-3: The impact of the number of tensor constraints in the noisy setting. Boxplots showing (a) Spearman correlation and (b) root-mean-square deviation (RMSD) for different number of tensor constraints over 20 trials. Simulated data was generated using Brownian motion model with three chromosomes, where each chromosome had 10 domains. Noise level of 0.5 was added. We used $\rho = 0.000001$ to solve the SDP. Green triangles and lines indicate the mean and median performance respectively.

Impact of the tuning parameter ρ

Figure C-4 explores the impact of tuning parameter ρ from equation (4.8) in the noisy setting. The choice of ρ has little impact on the accuracy of reconstruction. For the simulations in the noisy setting, shown in Figure 4-6 and real data analysis in Figure 4-7, we chose $\rho = 0.000001$.



(a) (b)

Figure C-4: The impact of ρ in the noisy setting. Boxplots showing (a) Spearman correlation and (b) root-mean-square deviation (RMSD) for different values of ρ over 20 trials. Simulated data was generated using Brownian motion model with one chromosome and 10 domains per chromosome as well as noise level of 0.5. We used the maximum number of triplet tensor constraints (120) to solve the SDP. Green triangles and lines indicate the mean and median performance respectively.

Real contact frequency data

Distance between neighboring beads

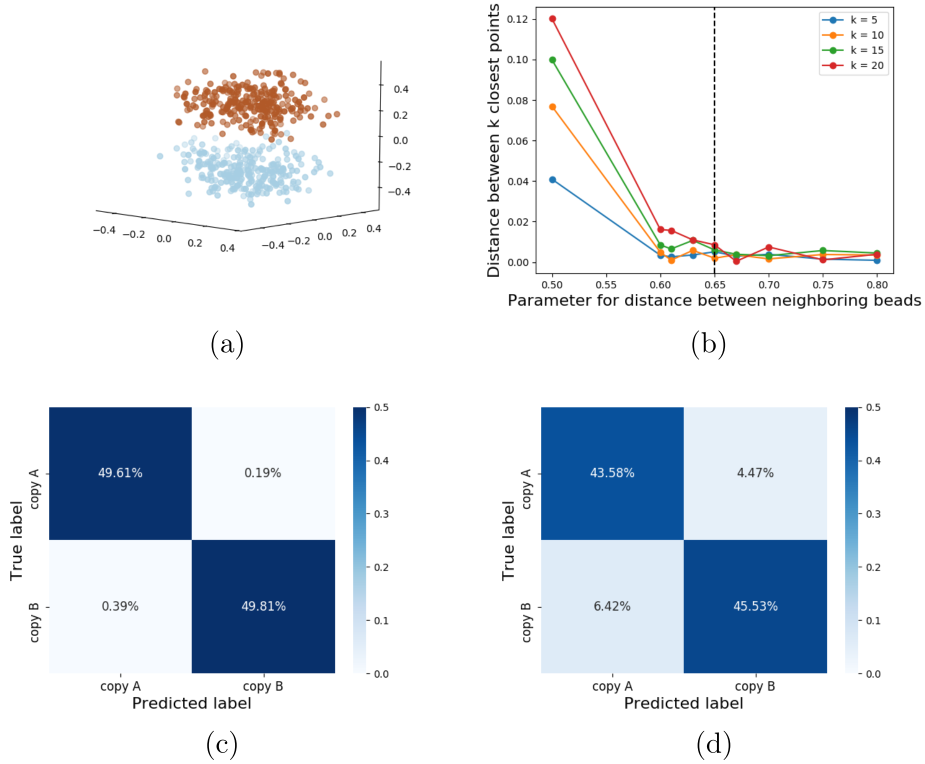
We consider different values for the distance between neighboring beads as input to our algorithm. If the distance between neighboring beads is chosen to be too small, the resulting 3D diploid reconstruction of the data may have gaps in the structure as shown in Figure C-5a, where the homologous loci $x_1, \ldots x_n$ (copy A) and $y_1, \ldots y_n$ (copy B) are completely separated. We chose increasing values for the distance between neighboring beads as input and quantified the separation between $x_1, \ldots x_n$ and $y_1, \ldots y_n$ by considering the distance between k closest points from $x_1, \ldots x_n$ and from $y_1, \ldots y_n$. We obtained the hyperplane separating $x_1, \ldots x_n$ and $y_1, \ldots y_n$ by fitting a support-vector machine (SVM) classifier. Next, we chose k points from $x_1, \ldots x_n$ and $y_1, \ldots y_n$ that are closest to the hyperplane and computed their centroids. Figure C-5b shows the sum of distances of the two centroids to the separating hyperplane, thus quantifying whether copy A points are separated from copy B. This distance should approach 0 as copy A and copy B points come closer together. Indeed, Figure C-5b shows that with parameter of 0.65, the distance between k closest points stabilizes close to 0 and thus we take 0.65 as our parameter of choice for the distance between neighboring beads.

We provide additional quantification regarding the separation of points in copy A and copy B by clustering the 3D structure using k-means into two clusters and computing a confusion matrix, where the true labels are given by copy A and copy B. If points $x_1, \ldots x_n$ and $y_1, \ldots y_n$ are completely separated, then k-means would result in near perfect accuracy of separation of all points into copy A and copy B. Figure C-5c shows that this is indeed the case when copy A and copy B are separated. For the chosen parameter of 0.65, the confusion matrix is shown in Figure C-5d, reinforcing the observation that indeed copy A and copy B are not separated by a clear gap, which is was our goal.

We note that our observations are robust to the exact choice of the distance between neighboring beads. In Figure C-6 we show the resulting 3D reconstruction as well as chromosome size and A compartment trends when parameter of 0.7 is chosen as the distance between neighboring beads.

Comparison with ChromSDE

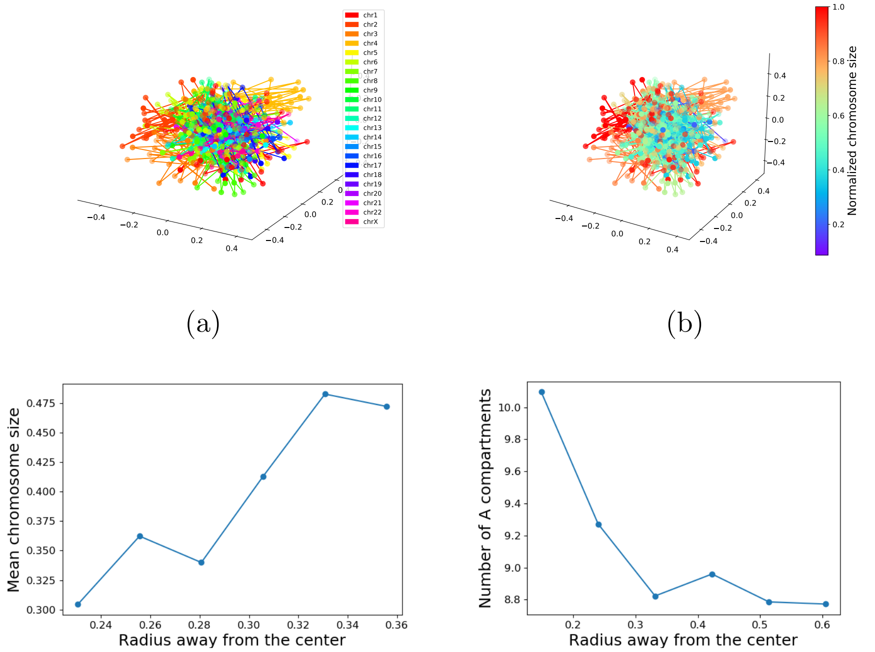
We compare our whole genome reconstruction to the reconstruction inferred by ChromSDE (Zhang et al., 2013). Since ChromSDE does not account for the fact that the measured contact frequencies and corresponding observed distances are a sum of four different distances, i.e. $\|x_i - x_j\|^2, \|x_i - y_j\|^2, \|y_i - x_j\|^2$, and $\|y_i - y_j\|^2$, we converted frequencies to distances using $D_{ij} = F_{ij}^{-1/2}$ and used $D_{ij}/4$ for each of the four distances so that the diploid configuration of the genome could be computed. We assumed that homologous loci are far apart, as has been observed in



(a) (b)

(c) (d)

Figure C-5: Empirical choice of parameter for the distance between neighboring beads. (a) The 3D genome reconstruction with parameter for the distance between neighboring beads set to 0.5. The homologous loci $x_1, \dots x_n$ (copy A) and $y_1, \dots y_n$ (copy B), colored by red and blue are completely separated. (b) The distance of centroids corresponding to k closest points to the SVM hyperplane separating copy A from copy B (red and blue points) for different parameter settings. The black dashed line corresponds to the chosen parameter of 0.65. (c) Confusion matrix quantifying how often points clustered via k-means (predicted label) were assigned their true label (copy A or copy B) when parameter of 0.5 was used. (d) Same as (c) for the chosen parameter 0.65. Higher confusion across labels indicates that points belonging to copy A and copy B are not clearly separated, as desired.



(a)

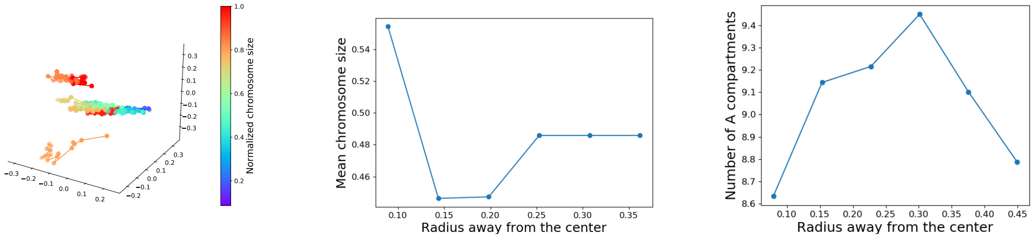
(b)

(c)

(d)

Figure C-6: 3D Diploid Genome Reconstruction with a different parameter for the distance between neighboring beads (0.7). (a) Estimated 3D positions of all chromosomes and their corresponding homologs with chromosomes colored according to chromosome number. (b) Whole diploid organization obtained via our method, colored by chromosome size. (c) Mean chromosome size as the distance from the center increases. (d) The number of A compartments as the distance from the center increases.

imaging studies (Bolzer et al., 2005; Nir et al., 2018), and thus set $\|x_i - y_i\|^2 = \infty$. Given the described distance constraints, we solved the SDP for the Gram matrix and obtained the 3D coordinates using eigenvector decomposition, similar to our method. Figure C-7 shows the corresponding solution and quantification of the mean chromosome size and number of A compartments as the radius from the center increases. The computed 3D diploid genome configuration obtained via ChromSDE does not recapitulate that chromosome size increases with distance away from the center and that the number of A compartments decreases with distance away from the center.



(a) (b) (c)

Figure C-7: 3D diploid genome reconstruction with ChromSDE. ChromSDE was run with distance matrix where distances for $\|x_i - x_j\|^2$, $\|x_i - y_j\|^2$, $\|y_i - x_j\|^2$, and $\|y_i - y_j\|^2$ were set to $D_{ij}/4$. (a) Estimated 3D positions of all chromosomes and their corresponding homologs at 10Mb resolution colored by chromosome size. (b) Mean chromosome size as the distance from the center increases. (c) The number of A compartments as the distance from the center increases.

Analysis of 3D diploid genome reconstruction

We provide further analysis of the 3D diploid genome reconstruction obtained using our algorithm from contact frequency data. Figure C-8 shows that for each chromosome, there is a strong correlation between chromosome size and distance of the chromosome away from the center.

Impact of the tuning parameter ρ

Figure C-9 explores the impact of the tuning parameter ρ from equation (4.8) on real data. We compute the RMSD between the 3D genome reconstruction com-

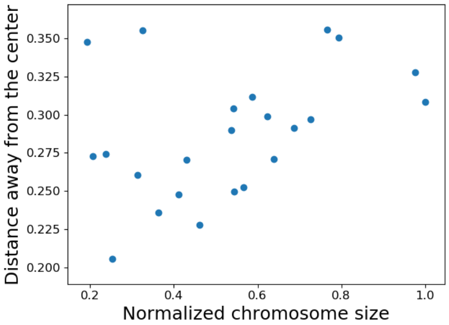


Figure C-8: Chromosome size (normalized by the size of the largest chromosome) versus the mean distance of the chromosome and its homolog away from the center.

puted with $\rho = 0.000001$ and the 3D genome reconstructions computed with other choices of $\rho \in (0.00001, 0.0001, 0.001, 0.01, 0.1, 10, 1)$ to quantify how much does the 3D structure change with increasing ρ. As shown in Figure C-9, the RMSD is low across different choices of the tuning parameter. In Figure C-10, we provide the 3D genome reconstruction and trends for mean number of A compartments and mean chromosome size versus the distance away from the center for the 3D reconstruction computed with $\rho = 10$ since the RMSD was the highest for this choice of the tuning parameter. We observe that the trends remain the same for a different choice of ρ.

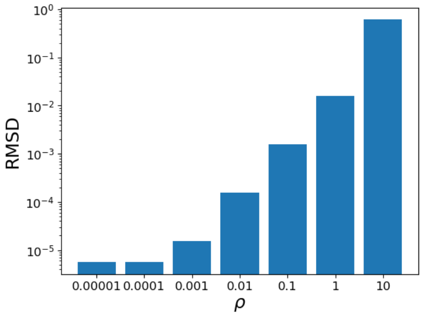
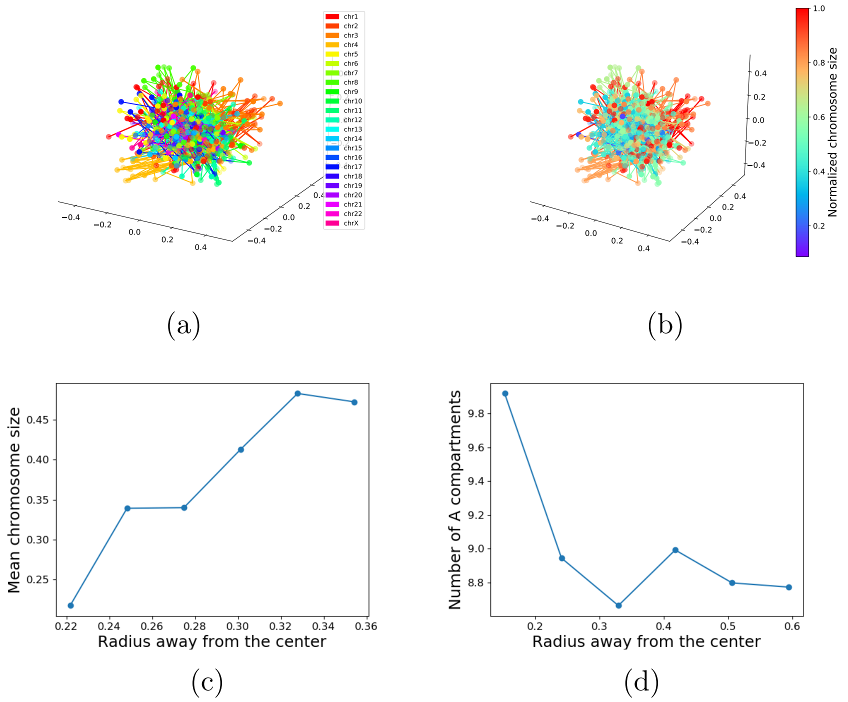


Figure C-9: The impact of ρ in real data. Root-mean-square deviation (RMSD) between the 3D genome reconstruction computed with $\rho = 0.000001$ and the 3D genome reconstructions computed with other choices of $\rho \in (0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10)$.



(a)

(b)

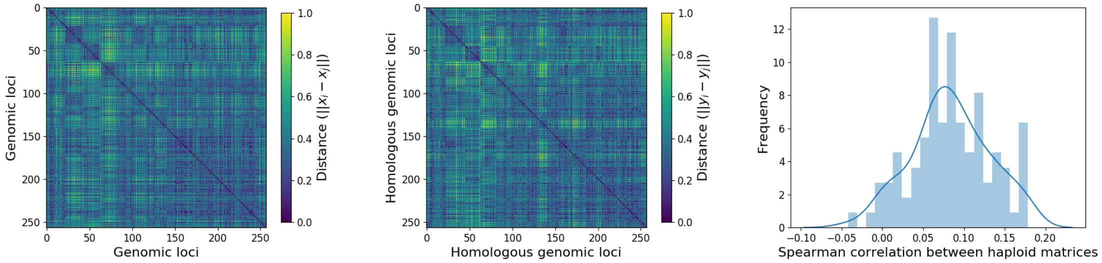
(c)

(d)

Figure C-10: 3D diploid genome reconstruction with $\rho = 10$. Estimated 3D positions of all chromosomes and their corresponding homologs at 10Mb resolution. Chromosomes are colored according to (a) chromosome number and (b) chromosome size. (c) Mean chromosome size as the distance from the center increases. (d) The number of A compartments as the distance from the center increases.

Haploid distance matrices

In order to determine whether modeling the diploid aspect of the genome provides valuable information regarding the 3D organization of the genome, we randomly labeled each homolog of a particular chromosome to correspond to either copy A or copy B of the chromosome and computed the Euclidean distances between all loci belonging to copy A, i.e. $||x_i - x_j||$ to obtain one haploid distance matrix and the Euclidean distances between all loci belonging to copy B, i.e. $||y_i - y_j||$ to obtain the second haploid distance matrix. Figure C-11a,b shows the haploid distance matrices where points $1, \ldots, n$ are assigned to copy A and points $n+1, \ldots, 2n$ are assigned to copy B. We were interested in comparing the two haploid distance matrices to see whether the two haploid matrices were the same or if modeling the diploid aspect of the genome also allowed us to learn about each homolog. Close inspection of the two matrices reveals that the distances are different in these two haploid matrices, suggesting that modeling the genome as a diploid structure gives additional information. We quantify the difference by computing the Spearman correlation between the distance matrices over 100 different samplings of assignments of chromosomes to either copy A or B. Figure C-11c shows the histogram of the calculated Spearman correlations with mean Spearman correlation of 0.08.



(a) (b) (c)

Figure C-11: Haploid distance matrices. (a) Haploid distance matrix for points belonging to copy A and (b) its corresponding homologous haploid distance matrix for points belonging to copy B. (c) Spearman correlation between haploid distance matrices over 100 different assignments of each homolog to either copy A or copy B.

Appendix D

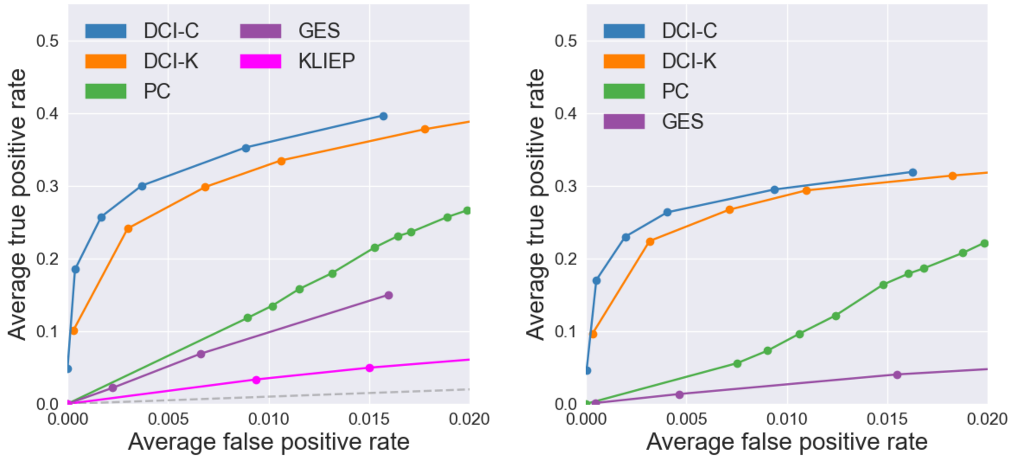
Appendix for Learning Causal Differences between Gene Regulatory Networks

Additional high-dimensional evaluation

High-dimensional setting: 10% changes. We present the results of increasing the number of changes between the two DAGs, and hence the size of S. We used the same simulation parameters as for Figure 5-5, i.e. $p = 100$ nodes, a neighbourhood size of $s = 10$, and sample size $n = 300$, except that the total number of changes was 10% of the number of edges in $B^{(1)}$, rather than 5%. As shown in Figure D-1, both initializations of the DCI algorithm still outperform separate estimation by GES and the PC algorithm. However, because the underlying DAGs have maintained constant sparsity while the difference-DAG has become more dense, the gains in performance by using the DCI algorithm have slightly diminished.

Real data analysis - ovarian cancer

We tested our method on an ovarian cancer data set (Tothill et al., 2008). This data set consists of the gene expression data of patients with ovarian cancer. The



(a)

(b)

Figure D-1: ROC curves for estimating the difference-DAG. (a) Evaluation of the estimated skeleton and (b) the estimated difference-DAG with $p = 100$ nodes, expected neighbourhood size $s = 10$, $n = 300$ samples, and 10% percent change between DAGs.

patients are divided into six subtypes (C1-C6). The C1 subtype was characterized by differential expression of genes associated with stromal and immune cell types and is associated with shorter survival rates. In this experiment, we divide the subjects into two groups, group 1 with $n_1 = 78$ subjects containing patients with C1 subtype, and group 2 with $n_2 = 113$ subjects containing patients with C2-C6 subtypes. In this work, we focused on two pathways from the KEGG database (Kanehisa et al., 2011; Ogata et al., 1999), the apoptosis pathway containing 87 genes, and the TGF-β pathway with 82 genes.

We compared our results to those obtained by the DPM method (Zhao et al., 2014), which infers the difference in the undirected setting. As input to Algorithm 2, we took $\mathcal{C}$ to be all of the nodes in the output of the DPM algorithm and took the difference undirected graph to be the fully connected graph on $\mathcal{C}$. We then learned the difference DAG using Algorithm 3. The final set of edges over different tuning parameters was chosen using stability selection as proposed in (Meinshausen and Bühlmann, 2010) and is shown in Figure 5-12. This procedure identified two hub nodes in the apoptosis pathway: BIRC3 and PRKAR2B. BIRC3 has been shown to be an inhibitor of apoptosis (Johnstone et al., 2008) and is one of the top disregulated genes in ovarian cancer (Jönsson et al., 2014). This gene has also been recovered by the DPM method as one of the hub nodes. While BIRC3 has high in-degree, hub gene PRKAR2B has high out-degree, making it a better candidate for possible interventions in ovarian cancer since knocking out a gene with high out-degree will have widespread downstream effects on the target genes. Indeed, PRKAR2B is a known important regulatory unit for cancer cell growth (Chiaradonna et al., 2008) and the RII-β protein encoded by PRKAR2B has already been studied as a therapeutic target for cancer therapy (Mikalsen et al., 2006; Cho-Chung, 1999). In addition, PRKAR2B has also been shown to play an important role in disease progression in ovarian cancer cells (Cheadle et al., 2008). Since the DPM method does not infer directionality, it is not possible to tell which of the hub genes might be a better interventional target. This is remedied by our method and its impact for identifying possible therapeutic targets in real data is showcased by finding an already known drug target for cancer.

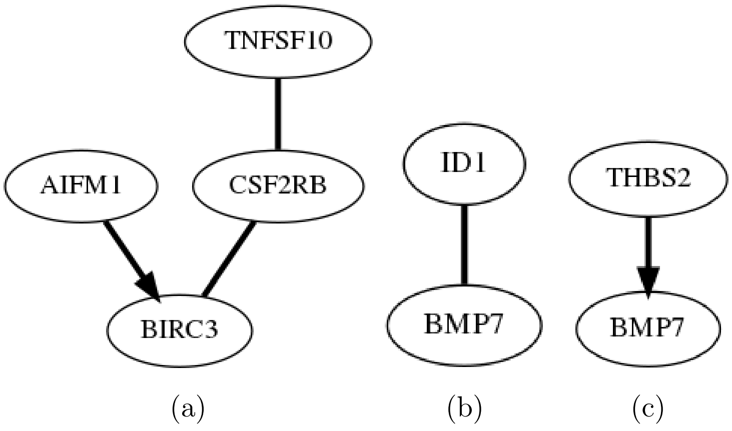


Figure D-2: Estimate of the difference-DAG between the two groups of ovarian cancer patients for (a) the apoptosis pathway estimated via PC, (b) the TGF-β pathway estimated via GES and (c) TGF-β pathway estimated via PC.

For the TGF-β pathway, our analysis identified THBS2 and COMP as hub nodes. Both of these genes have been implicated in resistance to chemotherapy in epithelial ovarian cancer (Marchini et al., 2013), confirming the importance of our findings. These nodes were also recovered by DPM.

Overall, the undirected graph discovered by DPM is similar to the DAG found by our method. The disparity in the TGF-β pathway between the difference undirected graph and the difference-DAG model can be explained by the fact that the edge between COMP$-$BMP7 in the undirected diffrence graph can be accounted for by the two edges BMP7$\rightarrow$ID1 and COMP$\rightarrow$ID1 in the difference-DAG. Though these edges might represent the true regulatory pathways, the sparsity-inducing penalty in the DPM algorithm could remove them while leaving the edge between COMP and BMP7. This disparity between the two algorithms highlights the importance of replacing correlative reasoning with causal reasoning, and accentuates the significance of our contribution.

We also applied the GES and PC algorithms on the ovarian cancer data set. We considered the set of edges that appeared in one estimated skeleton but disappeared in the other as the estimated skeleton of the difference-DAG. In determining orientations, we considered the arrows that were directed in one estimated CPDAG but disappeared in the other as the estimated set of directed arrows. Figure D-2 shows the results by applying the PC algorithm on the apoptosis and TGF-β pathway and the results by applying GES on the TGF-β pathway. Here we omitted GES results on the apoptosis pathway since GES algorithm did not discover any differences on the apoptosis pathway. Figure D-2 shows that PC and GES cannot discover any hub nodes.

Appendix E

Appendix for Causal Network Models of SARS-CoV-2 Expression and Aging to Identify Candidates for Drug Repurposing

Supplementary Note

Overview of methodology

Our drug discovery pipeline consists of three parts: mining relevant drugs, identifying the disease interactome, and investigating the drug mechanism. Fig. E-1 describes the inputs, outputs and algorithms used in each of the three parts. Briefly, the first part (mining relevant drugs) takes in normal and infected/diseased RNA-seq samples along with the public CMap database, which contains gene expression data on cell lines treated with a variety of FDA approved compounds, to train an autoencoder and subsequently construct synthetic interventions in the learned latent space. It outputs a list of drugs ranked by the correlation of each drug with the reverse disease signature. The second part of the pipeline (identifying disease interactome) also takes in the normal and infected/diseased RNA-seq samples as well as a PPI network (e.g. from the public iREF or STRING databases). It then identifies the genes that are differentially expressed in the disease and learns the disease interactome connecting these genes in the PPI network using the prize-collecting Steiner forest algorithm. In addition, the inferred ranked list of drugs output from part 1 in the pipeline is mapped to its targets using the public DrugCentral database. The drug targets are intersected with the disease interactome to further filter the list of drugs to only include those drugs that target nodes in the interactome. The third part of the pipeline (investigating drug mechanism) uses multi-sample RNA-seq data (e.g. high number of replicates or single-cell RNA-seq data) to learn the causal directions in the disease interactome using GSP, a causal structure discovery algorithm, and identifies which drugs and drug targets have the largest downstream causal effect on the disease interactome.

Comparison of SARS-CoV-2 versus IAV and RSV

In order to test how specific our findings are to SARS-CoV-2 and demonstrate the broad applicability of our pipeline, we apply our computational pipeline to two additional viral infections: respiratory syncytial virus (RSV) and influenza A virus (IAV). As for SARS-CoV-2 infection, we obtain gene expression data for these viruses from (Blanco-Melo et al., 2020). First, we perform differential expression analysis for IAV and RSV (Supplementary Fig. E-19) showing that only 3.19% and 19.6% of genes specific to SARS-CoV-2 are shared with RSV and IAV, respectively. Next, we apply our over-parameterized autoencoder and synthetic interventions framework to IAV and RSV to obtain drug lists ranked by their correlation with the reverse disease signature.

In order to quantitatively compare the drug lists obtained for RSV and IAV to the drug list for SARS-CoV-2, we measure the similarity of two rankings using curves akin to a receiver operating characteristic (ROC) curve, namely: given two rankings of n drugs, we consider the top k drugs in one of the lists and compute the number of drugs in common among these top k drugs for $k = 1, 2, \ldots n$. Thus, the x-coordinate in each plot indicates the proportion, k/n, of each drug list we consider and the y-coordinate is the size of the intersection of the two subsets normalized by k. The area under the curve (AUC) is a measure of similarity between two drug lists. When two drug lists are exactly the same, the AUC is 1 and when the two drug lists are maximally different (i.e., one drug list is the reverse of the other), the AUC is $1 - \ln(2) \approx .306$; see Supplementary Fig. E-8a. Supplementary Fig. E-20a-b show that that the drug lists for SARS-CoV-2 and RSV are significantly different and in fact very close to the lower bound, while the drug lists for SARS-CoV-2 and IAV are quite similar with an AUC of 0.843.

Finally, we perform the Steiner tree analysis based on the identified differentially expressed genes for IAV and RSV as well as the drug lists obtained by the overparameterized autoencoder. As for SARS-CoV-2, since the morbidity and fatality rate of IAV is higher in the aging population, we compute a combined IAV and aging interactome. This consists of 185 nodes and 486 edges based on 124 terminal genes. Since RSV is riskier in young children, but can also be serious for the aging population, we compute two interactomes, one without taking aging into account (234 nodes and 871 edges based on 139 terminal genes) and one combined with RSV and aging (303 nodes and 1177 edges based on 200 terminal genes) to make it more comparable to the other interactomes. To make the results comparable, since for SARS-CoV-2 we intersected the targets of the top 142 ranked drugs from the overparameterized autoencoder analysis with the interactome, we perform the analysis with the same number of drugs also for IAV and RSV. The resulting drugs and drug targets are shown in Supplementary Fig. E-21. For IAV, this results in 20 drugs, 13 of which overlap with drugs identified in the SARS-CoV-2 analysis. These drugs target 9 proteins in the interactome, 2 of which are also present in the SARS-CoV-2 interactome, namely EGFR and RIPK1. For RSV with and without aging the resulting drug lists as well as their targets have no overlap with the ones identified by SARS-CoV-2. In particular, the identified drug lists contain no tyrosine kinase inhibitors, thereby indicating the specificity of our results to SARS-CoV-2.

Supplementary Figures

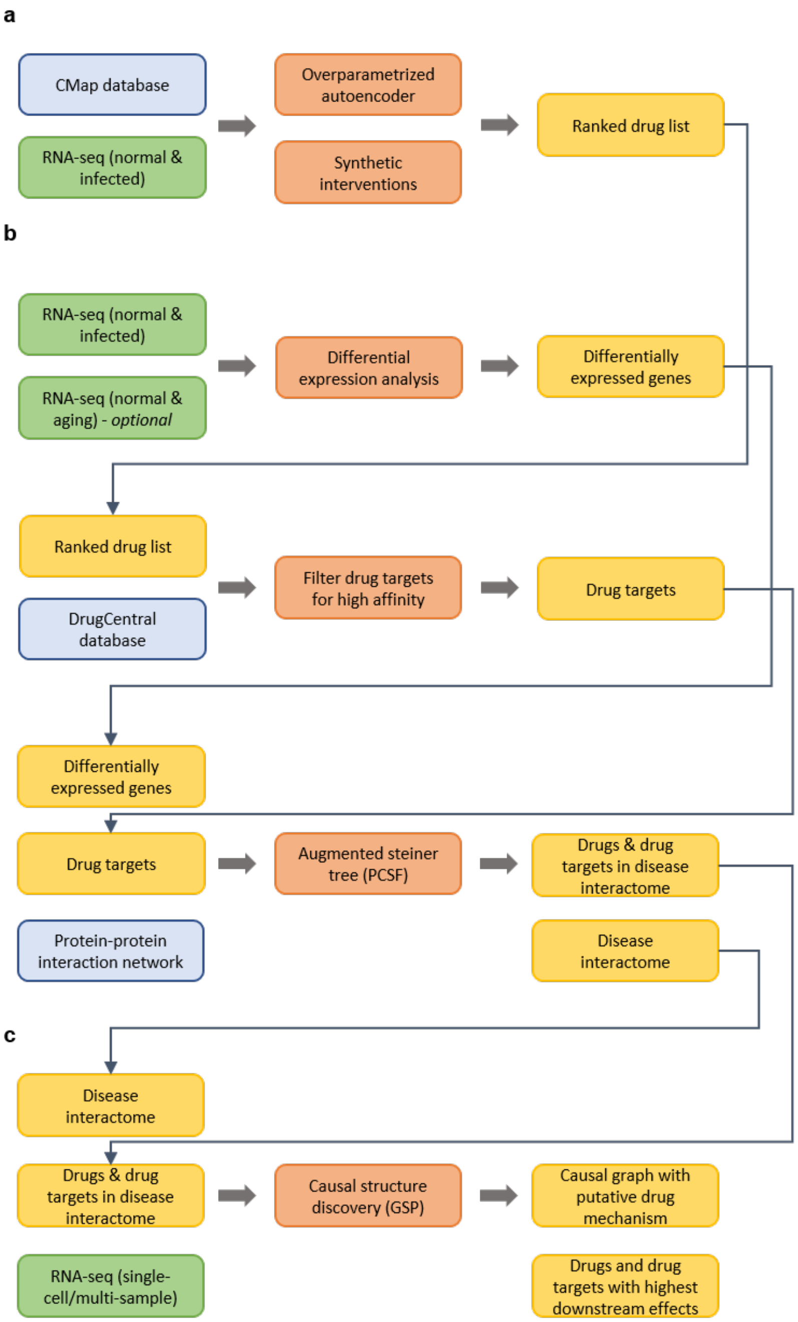

a
CMap database
RNA-seq (normal & infected)
Overparametrized autoencoder
Synthetic interventions
Ranked drug list
b
RNA-seq (normal & infected)
RNA-seq (normal & aging) - optional
Differential expression analysis
Differentially expressed genes
Ranked drug list
DrugCentral database
Filter drug targets for high affinity
Drug targets
Differentially expressed genes
Drug targets
Protein-protein interaction network
Augmented steiner tree (PCSF)
Drugs & drug targets in disease interactome
Disease interactome
c
Disease interactome
Drugs & drug targets in disease interactome
RNA-seq (single-cell/multi-sample)
Causal structure discovery (GSP)
Causal graph with putative drug mechanism
Drugs and drug targets with highest downstream effects

Figure E-1: Detailed schematic of our computational drug repurposing platform. Green boxes denote inputs that may need to be collected for the specific virus/disease and cell type of interest. Blue boxes denote inputs corresponding to databases that are publicly available. Orange boxes denote our computational methods and yellow boxes denote method outputs. Computational pipeline for (a) mining relevant drugs, (b) identifying disease interactome and (c) investigating drug mechanism.

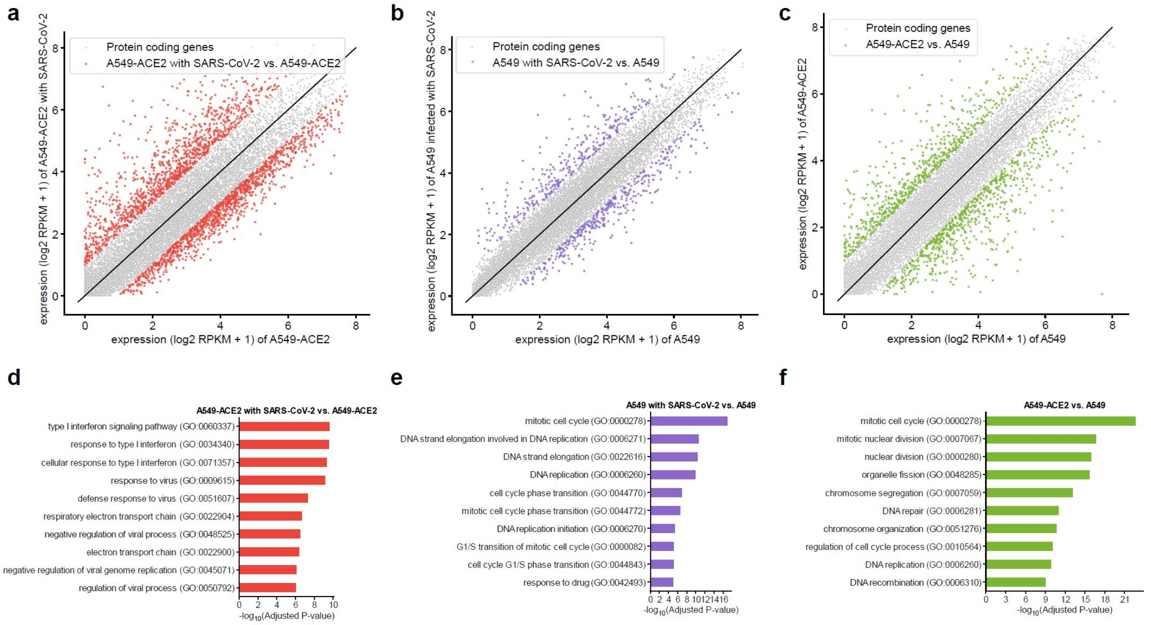


Figure E-2: (a) Gene expression of A549-ACE2 cells with and without SARS-CoV-2 infection, with differentially expressed genes in red. (b) Gene expression of A549 cells with and without SARS-CoV-2 infection, with differentially expressed genes in purple. (c) Gene expression of A549 cells with and without ACE2 receptor, with differentially expressed genes in green. (d) Top 10 gene ontology terms associated with differentially expressed genes between A549-ACE2 cells with and without SARS-CoV-2 infection . (e) Top 10 gene ontology terms associated with differentially expressed genes between A549 cells with and without SARS-CoV-2 infection. (f) Top 10 gene ontology terms associated with differentially expressed genes between A549 cells with and without ACE2 receptor. All gene ontology terms have adjusted p-value < 0.05.

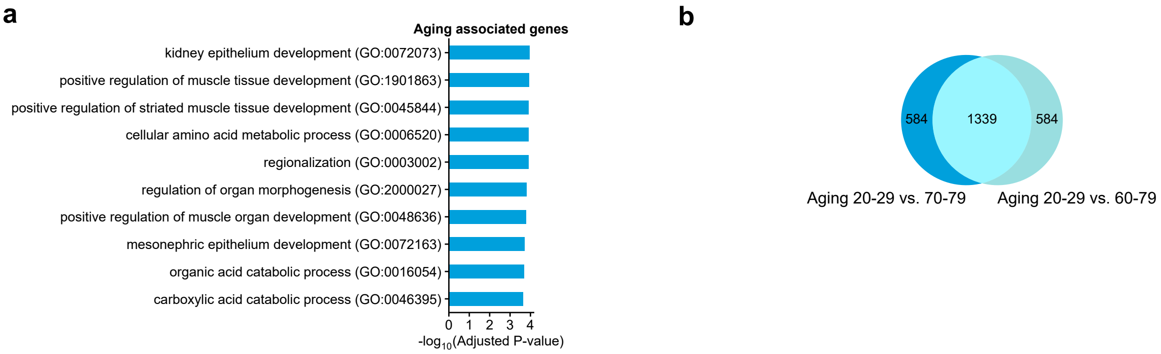


Figure E-3: (a) Top 10 gene ontology terms associated with aging. (b) Venn diagram showing significant overlap between aging associated genes considering different definitions of older, specifically just individuals in the oldest category (70-79) or individuals that are 60-79.

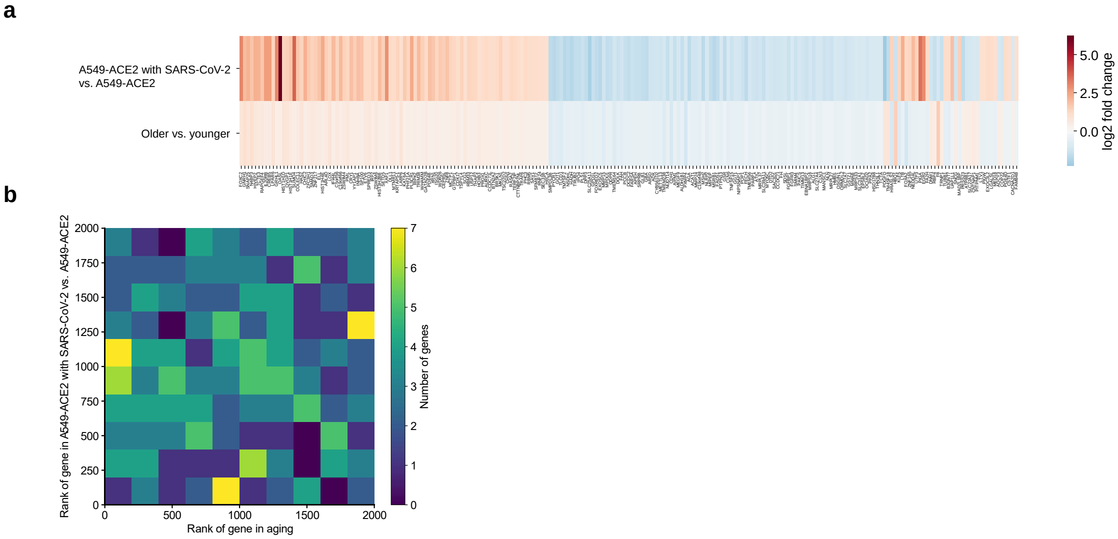


Figure E-4: (a) Heatmap of $\log_2$-fold changes of differentially expressed genes shared by SARS-CoV-2 and aging with gene names. (b) 2D histogram of the number of genes having a certain rank in aging and SARS-CoV-2 datasets.

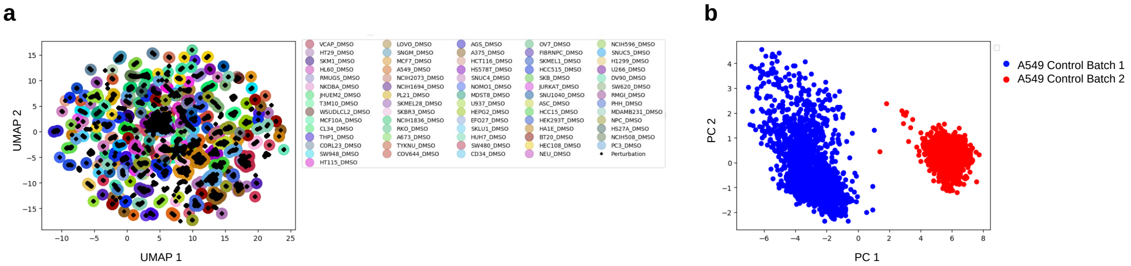
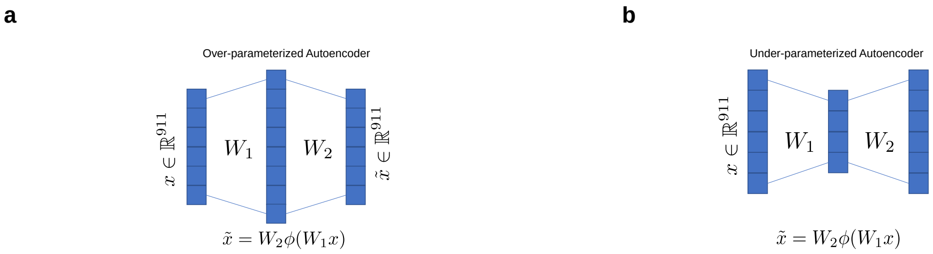


Figure E-5: (a) UMAP of control and perturbations across all cell types in CMap. The effect of a perturbation on a given cell type is small relative to the differences between cell types. (b) Principal component analysis highlighting batch effects for the control samples of the A549 cell line from CMap. K-means clustering by gene expression vector is used to identify and remove batch effects (represented as red and blue clusters).


a
Over-parameterized Autoencoder
$x \in \mathbb{R}^{911}$
W_1
W_2
$\tilde{x} \in \mathbb{R}^{911}$
$\tilde{x} = W_2\phi(W_1 x)$
b
Under-parameterized Autoencoder
$x \in \mathbb{R}^{911}$
W_1
W_2
$\tilde{x} \in \mathbb{R}^{911}$
$\tilde{x} = W_2\phi(W_1 x)$

c

Num. Hidden Units	Num. Hidden Layers	Nonlinearity	Optimizer, LR	Initialization	Seed Used	Training Loss	Test Loss
1024	1	Leaky ReLU	Adam, 1e-4	PyTorch Default	17	7.3 x 10^-7	1.1 x 10^-6
100	1	Leaky ReLU	Adam, 1e-4	PyTorch Default	17	2.8 x 10^-3	2.8 x 10^-3
1024	1	CosID	Adam, 1e-4	PyTorch Default	17	6.4 x 10^-6	6.5 x 10^-6

Figure E-6: Overview of autoencoder architectures, optimization methods and hyperparameter settings considered. (a) Diagram representing an overparameterized autoencoder. While this autoencoder is capable of learning the identity function, training leads to a solution that better aligns drug signatures across cell types in the latent space. (b) Diagram representing an underparameterized autoencoder. While this architecture is most commonly used in practice, it does not align drug signatures as well in the latent space as its overparameterized counterpart; see Supplementary Fig. 6-4. (c) Details on the width, depth, nonlinearity, optimization method, learning rate, random seed, training loss and test loss for all architectures considered in this work.

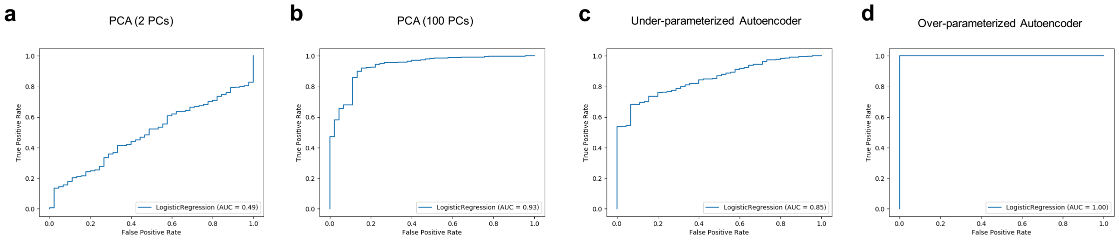


Figure E-7: Receiver operating characteristic (ROC) curves for the agreement in classification between gene expression vectors and reconstructed gene expression vectors obtained using an embedding given by the first 2 principle components in (a), the first 100 principle components in (b), an underparameterized autoencoder in (c), and an overparameterized autoencoder in (d). While a logistic regression model trained to classify between 831 A549 control samples and 32893 A549 perturbation samples shows differences in predictions on original gene expression vectors versus underparameterized autoencoder reconstructions and reconstructions from the top 2 or 100 principal component, the overparameterized embedding allows near perfect reconstruction of the original gene expression vectors with no difference in predictions between using overparameterized embeddings for gene expression vectors and original gene expression vectors.

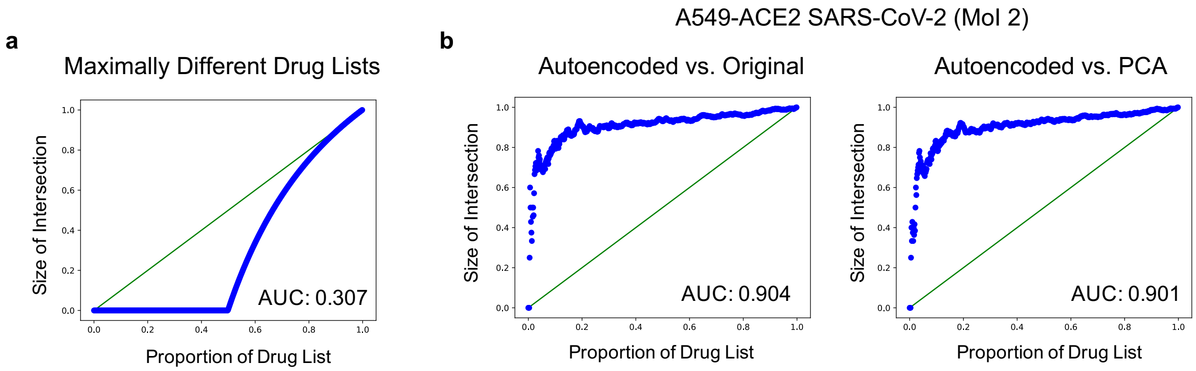


Figure E-8: Quantitative analysis of similarity between drug lists obtained using the latent space embedding as compared to the original and PCA embedding (using 2 PCs). Given two rankings of n drugs, we consider the top k drugs and plot the number of drugs in common among these top k drugs for $k = 1, 2, \ldots n$; i.e., the x-coordinate of a point indicates the proportion, k/n, of each drug list we consider and the y-coordinate is the size of the intersection of the two subsets normalized by k. AUC denotes the area under the curve; (a) shows the result when considering two maximally different drug lists, i.e., when one is the reverse of the other, resulting in an AUC of 0.307; (b) demonstrates that the drug list produced in the latent space of the over-parameterized autoencoder is similar to that produced in the original space and to that produced using 2 PCs. The advantages of using the over-parameterized autoencoder are that the resulting latent space contains enough signal to reconstruct gene expression vectors well and provides better alignment between drug signatures across cell types than in the original space.

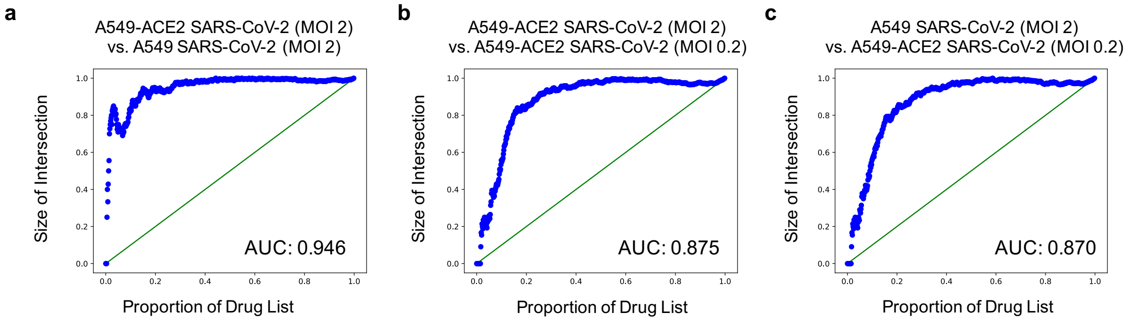
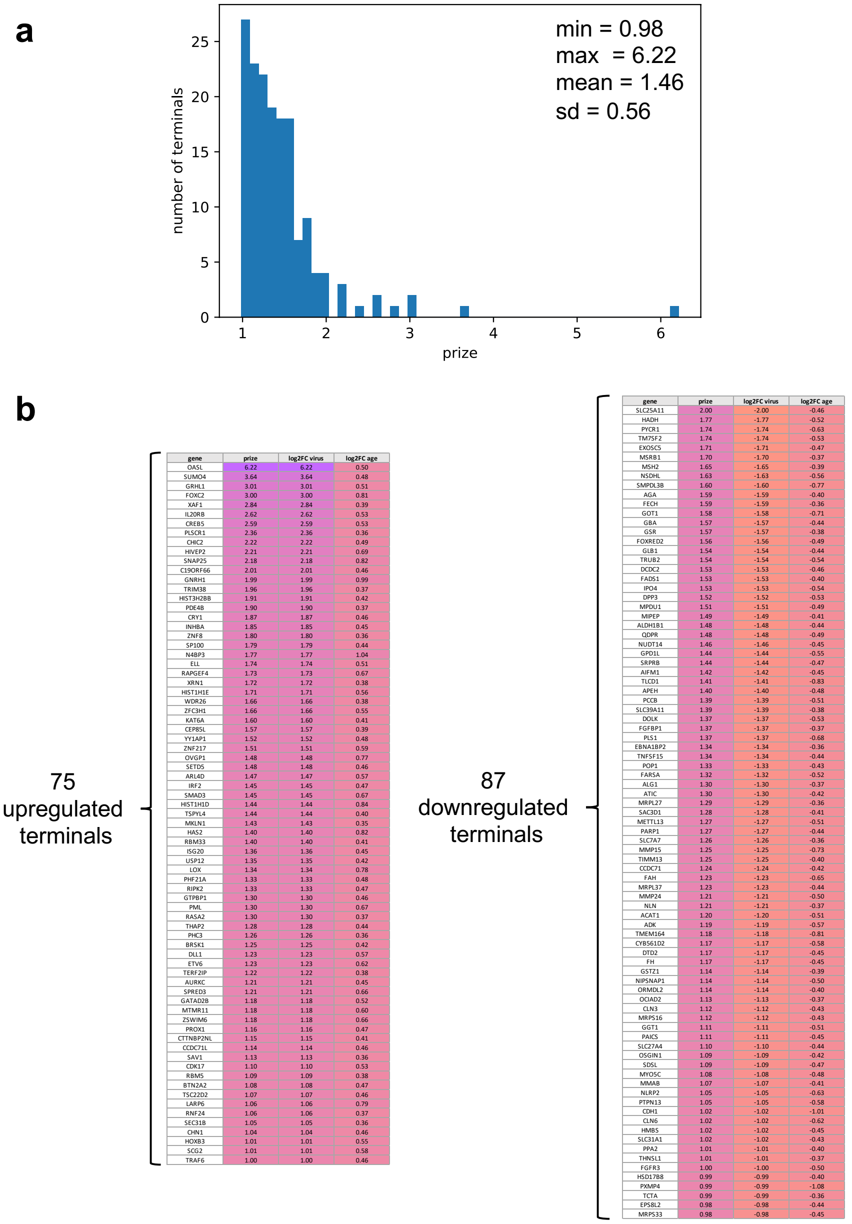


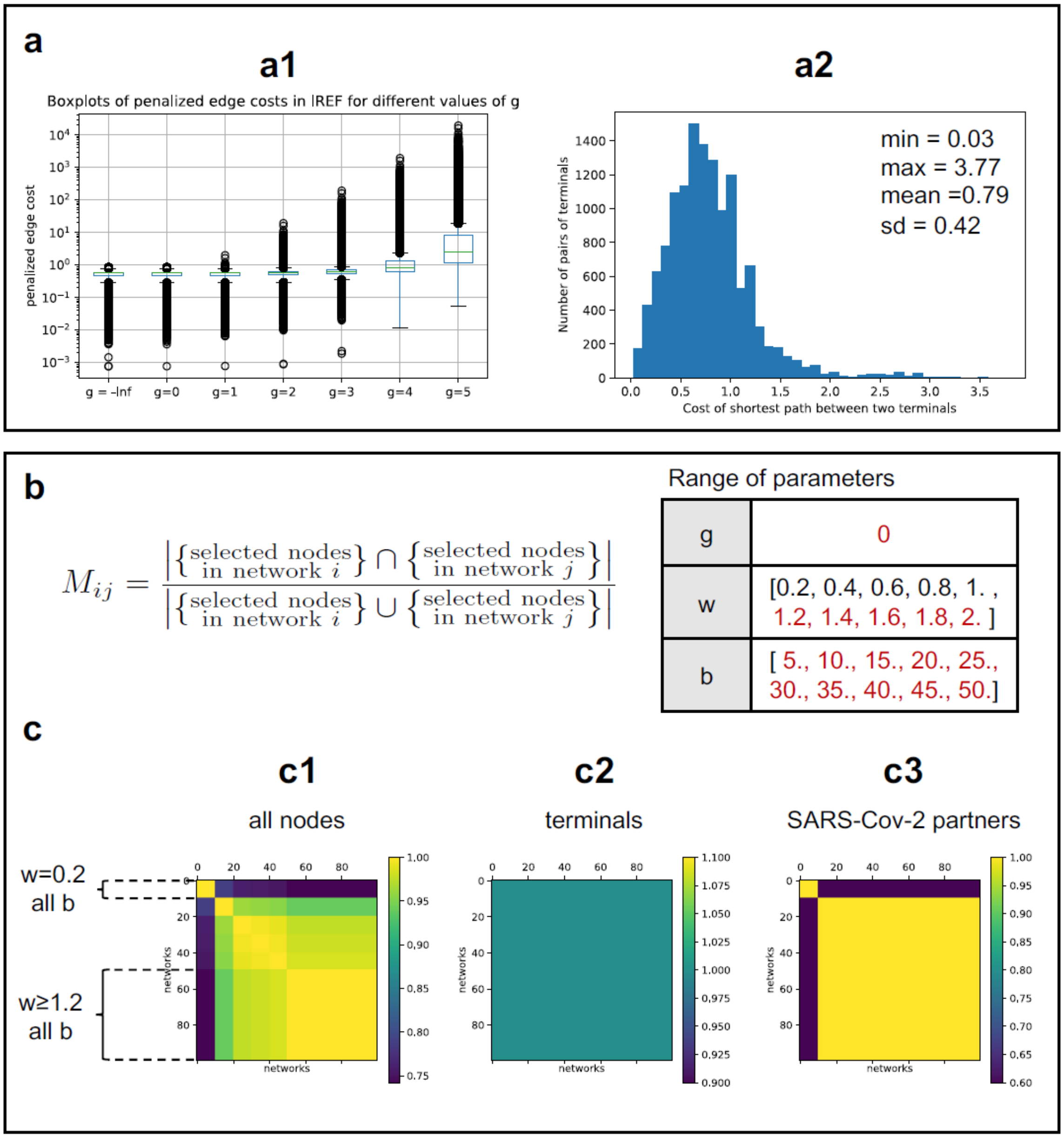
Figure E-9: Quantitative analysis of similarity between drug lists obtained using the overparameterized autoencoder on gene expression data from different MOIs for A549 cells with and without ACE2 receptor. (a) Comparison of drug lists obtained from SARS-CoV-2 infected A549-ACE2 cells with MOI 2 and A549 cells with MOI 2, (b) A549-ACE2 cells with MOI 2 and A549-ACE2 cells with MOI 0.2, and (c) A549 cells with MOI 2 and A549-ACE2 cells with MOI 0.2. The similarity between the drug lists drops when comparing an MOI of 2 to an MOI of 0.2, which is consistent with the observation by (Blanco-Melo et al., 2020) that low-MOI conditions did not stimulate an important interferon-I and -III response.



gene	prize	log2FC virus	log2FC age
OASL	6.22	6.22	0.50
SUMO4	3.64	3.64	0.48
GRHL1	3.01	3.01	0.51
FOXC2	3.00	3.00	0.81
XAF1	2.84	2.84	0.39
IL20RB	2.62	2.62	0.53
CREB5	2.59	2.59	0.53
PLSCR1	2.36	2.36	0.36
CHIC2	2.22	2.22	0.49
HIVEP2	2.21	2.21	0.69
SNAP25	2.18	2.18	0.82
C19ORF66	2.01	2.01	0.46
GNRH1	1.99	1.99	0.99
TRIM38	1.96	1.96	0.37
HIST3H2BB	1.91	1.91	0.42
PDE4B	1.90	1.90	0.37
CRY1	1.87	1.87	0.46
INHBA	1.85	1.85	0.45
ZNF8	1.80	1.80	0.36
SP100	1.79	1.79	0.44
N4BP3	1.77	1.77	1.04
ELL	1.74	1.74	0.51
RAPGEF4	1.73	1.73	0.67
XRN1	1.72	1.72	0.38
HIST1H1E	1.71	1.71	0.56
WDR26	1.66	1.66	0.38
ZFC3H1	1.66	1.66	0.55
KAT6A	1.60	1.60	0.41
CEP85L	1.57	1.57	0.39
YY1AP1	1.52	1.52	0.48
ZNF217	1.51	1.51	0.59
OVGP1	1.48	1.48	0.77
SETD5	1.48	1.48	0.46
ARL4D	1.47	1.47	0.57
IRF2	1.45	1.45	0.47
SMAD3	1.45	1.45	0.67
HIST1H1D	1.44	1.44	0.84
TSPYL4	1.44	1.44	0.40
MKLN1	1.43	1.43	0.35
HAS2	1.40	1.40	0.82
RBM33	1.40	1.40	0.41
ISG20	1.36	1.36	0.45
USP12	1.35	1.35	0.42
LOX	1.34	1.34	0.78
PHF21A	1.33	1.33	0.48
RIPK2	1.33	1.33	0.47
GTPBP1	1.30	1.30	0.46
PML	1.30	1.30	0.67
RASA2	1.30	1.30	0.37
THAP2	1.28	1.28	0.44
PHC3	1.26	1.26	0.36
BRSK1	1.25	1.25	0.42
DLL1	1.23	1.23	0.57
ETV6	1.23	1.23	0.62
TERF2IP	1.22	1.22	0.38
AURKC	1.21	1.21	0.45
SPRED3	1.21	1.21	0.66
GATAD2B	1.18	1.18	0.52
MTMR11	1.18	1.18	0.60
ZSWIM6	1.18	1.18	0.66
PROX1	1.16	1.16	0.47
CTTNBP2NL	1.15	1.15	0.41
CCDC71L	1.14	1.14	0.46
SAV1	1.13	1.13	0.36
CDK17	1.10	1.10	0.53
RBM5	1.09	1.09	0.38
BTN2A2	1.08	1.08	0.47
TSC22D2	1.07	1.07	0.46
LARP6	1.06	1.06	0.79
RNF24	1.06	1.06	0.37
SEC31B	1.05	1.05	0.36
CHN1	1.04	1.04	0.46
HOXB3	1.01	1.01	0.55
SCG2	1.01	1.01	0.58
TRAF6	1.00	1.00	0.46

gene	prize	log2FC virus	log2FC age
SLC25A11	2.00	-2.00	-0.46
HADH	1.77	-1.77	-0.52
PYCR1	1.74	-1.74	-0.63
TM7SF2	1.74	-1.74	-0.53
EXOSC5	1.71	-1.71	-0.47
MSRB1	1.70	-1.70	-0.37
MSH2	1.65	-1.65	-0.39
NSDHL	1.63	-1.63	-0.56
SMPDL3B	1.60	-1.60	-0.77
AGA	1.59	-1.59	-0.40
FECH	1.59	-1.59	-0.36
GOT1	1.58	-1.58	-0.71
GBA	1.57	-1.57	-0.44
GSR	1.57	-1.57	-0.38
FOXRED2	1.56	-1.56	-0.49
GLB1	1.54	-1.54	-0.44
TRUB2	1.54	-1.54	-0.54
DCDC2	1.53	-1.53	-0.46
FADS1	1.53	-1.53	-0.40
IPO4	1.53	-1.53	-0.54
DPP3	1.52	-1.52	-0.53
MPDU1	1.51	-1.51	-0.49
MIPEP	1.49	-1.49	-0.41
ALDH1B1	1.48	-1.48	-0.44
QDPR	1.48	-1.48	-0.49
NUDT14	1.46	-1.46	-0.45
GPD1L	1.44	-1.44	-0.55
SRPRB	1.44	-1.44	-0.47
AIFM1	1.42	-1.42	-0.45
TLCD1	1.41	-1.41	-0.83
APEH	1.40	-1.40	-0.48
PCCB	1.39	-1.39	-0.51
SLC39A11	1.39	-1.39	-0.38
DOLK	1.37	-1.37	-0.53
FGFBP1	1.37	-1.37	-0.37
PLS1	1.37	-1.37	-0.68
EBNA1BP2	1.34	-1.34	-0.36
TNFSF15	1.34	-1.34	-0.44
POP1	1.33	-1.33	-0.43
FARSA	1.32	-1.32	-0.52
ALG1	1.30	-1.30	-0.37
ATIC	1.30	-1.30	-0.42
MRPL27	1.29	-1.29	-0.36
SAC3D1	1.28	-1.28	-0.41
METTL13	1.27	-1.27	-0.51
PARP1	1.27	-1.27	-0.44
SLC7A7	1.26	-1.26	-0.36
MMP15	1.25	-1.25	-0.73
TIMM13	1.25	-1.25	-0.40
CCDC71	1.24	-1.24	-0.42
FAH	1.23	-1.23	-0.65
MRPL37	1.23	-1.23	-0.44
MMP24	1.21	-1.21	-0.50
NLN	1.21	-1.21	-0.37
ACAT1	1.20	-1.20	-0.51
ADK	1.19	-1.19	-0.57
TMEM164	1.18	-1.18	-0.81
CYB561D2	1.17	-1.17	-0.58
DTD2	1.17	-1.17	-0.45
FH	1.17	-1.17	-0.45
GSTZ1	1.14	-1.14	-0.39
NIPSNAP1	1.14	-1.14	-0.50
ORMDL2	1.14	-1.14	-0.40
OCIAD2	1.13	-1.13	-0.37
CLN3	1.12	-1.12	-0.43
MRPS16	1.12	-1.12	-0.43
GGT1	1.11	-1.11	-0.51
PAICS	1.11	-1.11	-0.45
SLC27A4	1.10	-1.10	-0.44
OSGIN1	1.09	-1.09	-0.42
SDSL	1.09	-1.09	-0.47
MYO5C	1.08	-1.08	-0.48
MMAB	1.07	-1.07	-0.41
NLRP2	1.05	-1.05	-0.63
PTPN13	1.05	-1.05	-0.58
CDH1	1.02	-1.02	-1.01
CLN6	1.02	-1.02	-0.62
HMBS	1.02	-1.02	-0.45
SLC31A1	1.02	-1.02	-0.43
PPA2	1.01	-1.01	-0.40
THNSL1	1.01	-1.01	-0.37
FGFR3	1.00	-1.00	-0.50
HSD17B8	0.99	-0.99	-0.40
PXMP4	0.99	-0.99	-1.08
TCTA	0.99	-0.99	-0.36
EPS8L2	0.98	-0.98	-0.44
MRPS33	0.98	-0.98	-0.45

Figure E-10: Terminal node selection for prize-collecting Steiner forest analysis. Terminal genes include 162 genes present in the IREF interactome that are either upregulated in both SARS-CoV-2 infection and aging or downregulated in both SARS-CoV-2 infection and aging. Each terminal gene is prized with its absolute $\log_2$-fold change between SARS-CoV-2 infected A549-ACE2 cells and normal A549-ACE2 cells. (a) Histogram of prizes for terminal genes along with descriptive statistics. (b) Table of 75 terminal genes upregulated in both SARS-CoV-2 infection and aging (left) and table of 87 terminal genes downregulated in both SARS-CoV-2 infection and aging, along with prize and $\log_2$ fold change information.



$$M_{ij} = \frac{\left| \left\{ \substack{\text{selected nodes} \\ \text{in network } i} \right\} \cap \left\{ \substack{\text{selected nodes} \\ \text{in network } j} \right\} \right|}{\left| \left\{ \substack{\text{selected nodes} \\ \text{in network } i} \right\} \cup \left\{ \substack{\text{selected nodes} \\ \text{in network } j} \right\} \right|}$$

Range of parameters

g	0
w	[0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2.]
b	[5., 10., 15., 20., 25., 30., 35., 40., 45., 50.]

Figure E-11: Parameter selection via sensitivity analysis for prize-collecting Steiner forest analysis. (a1) Boxplot of penalized edge costs in the IREF interactome for different values of g. The distribution of penalized edge costs are very similar for $g = -\infty$ and $g = 0$. For these values of g, the maximum penalized edge cost is upper bounded by 1. (a2) Histogram of shortest path cost between any two terminals in the IREF interactome for $g = 0$, along with descriptive statistics. (b) Range of parameters g, w and b used in sensitivity analysis. Red values indicate a stable range for the interactome obtained with the prize-collecting Steiner forest algorithm. We retain $g = 0$, $w = 1.4$ and $b = 40$ for our subsequent analysis. (c1-3) Heatmaps of the matrix M indexed for different types of selected nodes: all nodes (c1), terminal nodes (c2) and SARS-CoV-2 interaction partners (c3). Each row/column corresponds to a prize-collecting Steiner forest obtained from a given set of parameters $(g = 0, w, b)$. A stability region for the prize-collection Steiner forest solution appears for $g = 0$, $w \geq 1.2$ and $b \in [5, 50]$.

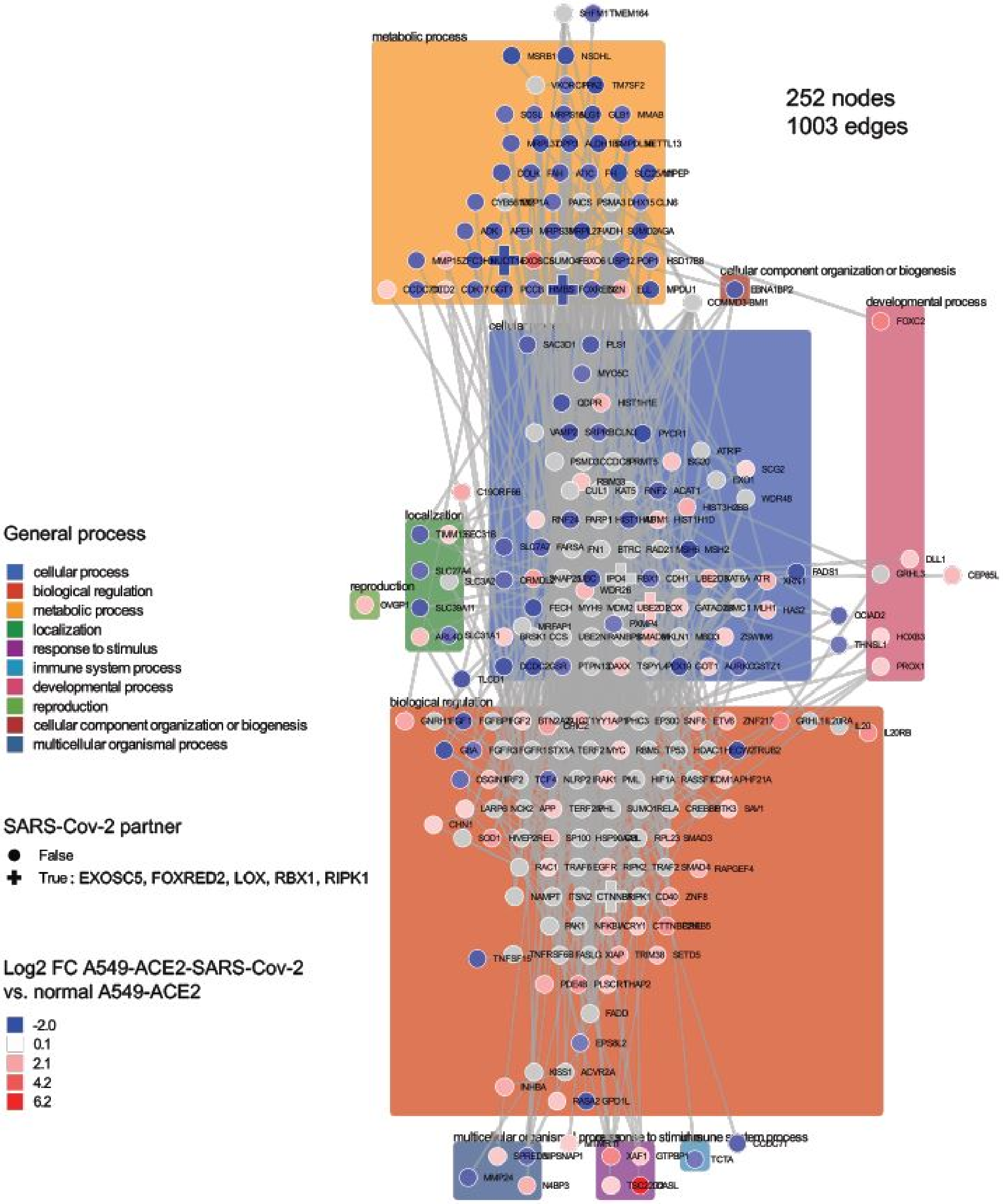
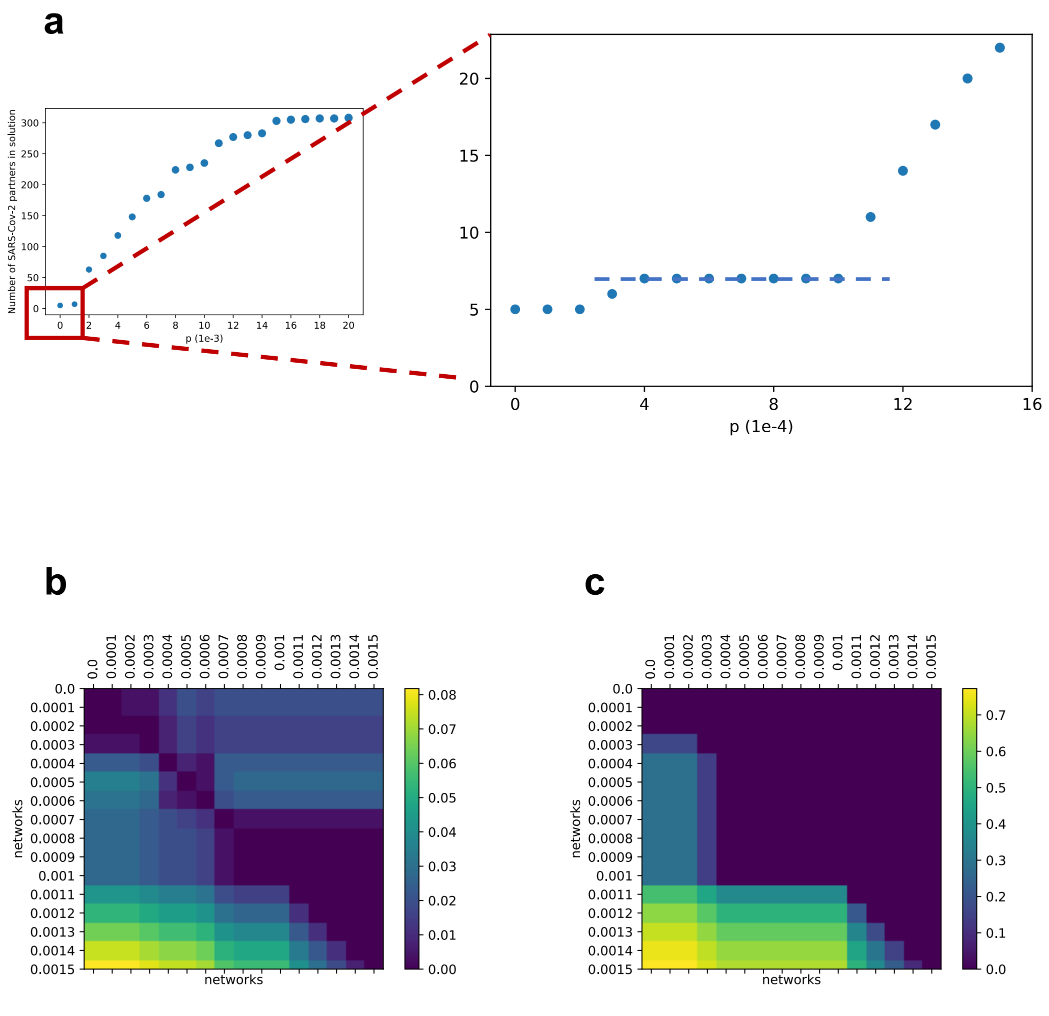

252 nodes
1003 edges
metabolic process
cellular process
localization
reproduction
developmental process
cellular component organization or biogenesis
biological regulation
multicellular organismal process
response to stimulus
immune system process
SHFM1 TMEM164
MSRB1 NSDHL
VKORC1 PRR2 TM7SF2
SOSL MRPS18 ALG1 GLB1 MMAB
MRPL37 DPP3 ALDH18 SMPDL3 METTL13
COLK FAH ATIC FH SLC25 MIPEP
CYB561 TXP1A PAICS PSMA3 DHX15 CLN6
ADK APEH MRPS30 MRPL27 HADH SUMO2 AGA
MMP15 ZFC3H1 NUDT14 EXOSC5 SUMO4 FBXO6 USP12 POP1 HSD17B8
CCDC70 TXD2 CDK17 OGT1 PCCB HMBS FOXRED2 DN DLL MPDU1
EBNA1BP2
COMMD3-BMI1
FOXC2
SAC3D1 PLS1
MYO5C
QDPR HIST1H1E
VAMP2 SRPRB CLN PYCR1
ATRIP
PSMD3 CCDC PRMT5 ISG20 SCG2
RBM33 EXO1
C19ORF66 CUL1 KAT5 RNF2 ACAT1 WDR48
HIST3H2BB
RNF24 PARP1 HIST1H4 RBM1 HIST1H1D
TIMM13 SEC31B
SLC7A7 FARSA FN1 BTRC RAD21 MSH6 MSH2
DLL1
SLC27A4 SLC3A2 ORMDL2 SNAP25 IPO4 RBX1 CDH1 UBE2D3 KAT6A ATR XRN1 FADS1 GRHL3 CEP85L
WDR26
OVGP1 SLC38A11 FECH MYH9 MDM2 UBE2D LOX GATAD2 SMC1 MLH1 HAS2 OCIAD2
MRFAP1 PXMP4
ARL4D SLC31A1 BRSK1 CCS UBE2N RANBP SMAD KLN1 MBD3 ZSWIM6 HOXB3 THNSL1
DCDC2 GSR PTPN13 DAXX TSPYL4 PEX19 GOT1 AURKC GSTZ1 PROX1
TLCD1
GNRH1 IGF1 FGFBP1 FGF2 BTN2A2 UGT1 YY1AP1 PHC3 EP300 SNF8 ETV6 ZNF217 GRHL1 IL20RA IL20 IL20RB
ERC2
GBA FGFR3 FGFR1 STX1A TERF2 MYC RBM5 TP53 HDAC1 HECW2 TRUB2
OSGIN1 IRF2 TCF4 NLRP2 IRAK1 PML HIF1A RASSF1 KDM1A PHF21A
LARP6 NCK2 APP TERF2IP PHL SUMO1 RELA CREBBP BTK3 SAV1
CHN1
SOD1 HIVEP2 REL SP100 HSP90AA1 CBL RPL23 SMAD3
RAC1 TRAF6 EGFR RIPK2 TRAF2 SMAD4 RAPGEF4
NAMPT ITSN2 CTNNB1 RIPK1 CD40 ZNF8
PAK1 NFKBIA CRY1 CTTNBP2 ERBB5
TNFSF15 TNFRSF6B FASLG XIAP TRIM38 SETD5
PDE4B PLSCR THAP2
FADD
EPS8L2
KISS1 ACVR2A
INHBA
RASA2 GPD1L
CCDC71
SPRED3 SIPSNAP1
MMP24 N4BP3
XAF1 GTPBP1 TCTA
TSC22D3 DASL
General process
🟦 cellular process
🟥 biological regulation
🟧 metabolic process
🟩 localization
🟪 response to stimulus
🟦 immune system process
🟥 developmental process
🟩 reproduction
🟫 cellular component organization or biogenesis
🟦 multicellular organismal process
SARS-Cov-2 partner
● False
✚ True : EXOSC5, FOXRED2, LOX, RBX1, RIPK1
Log2 FC A549-ACE2-SARS-Cov-2
vs. normal A549-ACE2
🟦 -2.0
□ 0.1
🟥 2.1
🟥 4.2
🟥 6.2

Figure E-12: Interactome obtained from the prize-collecting Steiner forest algorithm (with parameters $g = 0$, $w = 1.4$, $b = 40$) using the terminal gene list from Supplementary Fig. E-10. The interactome contains 1,003 edges between 252 genes, five of which are known SARS-CoV-2 interaction partners (EXOSC5, FOXRED2, LOX, RBX1, RIPK1). Genes in the interactome are grouped by general process.



$$M_{ij} = \frac{|\{\text{all nodes in network } i\} \setminus \{\text{all nodes in network } j\}|}{|\{\text{all nodes in network } i\}|}$$

$$M_{ij} = \frac{\left|\left\{\substack{\text{SARS-Cov-2 partners}\\ \text{in network } i}\right\} \setminus \left\{\substack{\text{SARS-Cov-2 partners}\\ \text{in network } j}\right\}\right|}{\left|\left\{\substack{\text{SARS-Cov-2 partners}\\ \text{in network } i}\right\}\right|}$$

Figure E-13: Selection of the prize p for non-terminal SARS-CoV-2 interaction partners (all but EXOSC5, FOXRED2 and LOX) via sensitivity analysis. (a) Number of SARS-CoV-2 interaction partners collected in the interactome obtained from the prize-collecting Steiner forest algorithm for different values of p ranging from 0 to 0.02. For $p > 0.02$, all known SARS-CoV-2 interaction partners present in the IREF network are collected in the final interactome. A stability region appears for $p \in [4 \cdot 10^{-4}, 10^{-3}]$ with 7 SARS-CoV-2 interaction partners collected. (b-c) Heatmaps of the matrix M indexed for different types of selected nodes: all nodes (b), and SARS-CoV-2 interaction partners (c). Each row/column corresponds to a prize-collecting Steiner forest obtained from a given set of parameters $(g = 0, w = 1.4, b = 40, p)$. A stability region for the prize-collection Steiner forest solution appears for $g = 0$, $w = 1.4$ and $b = 40$ and $p \in [7 \cdot 10^{-4}, 10^{-3}]$. We retain $g = 0$, $w = 1.4$, $b = 40$ and $p = 8 \cdot 10^{-4}$ for our subsequent analysis.

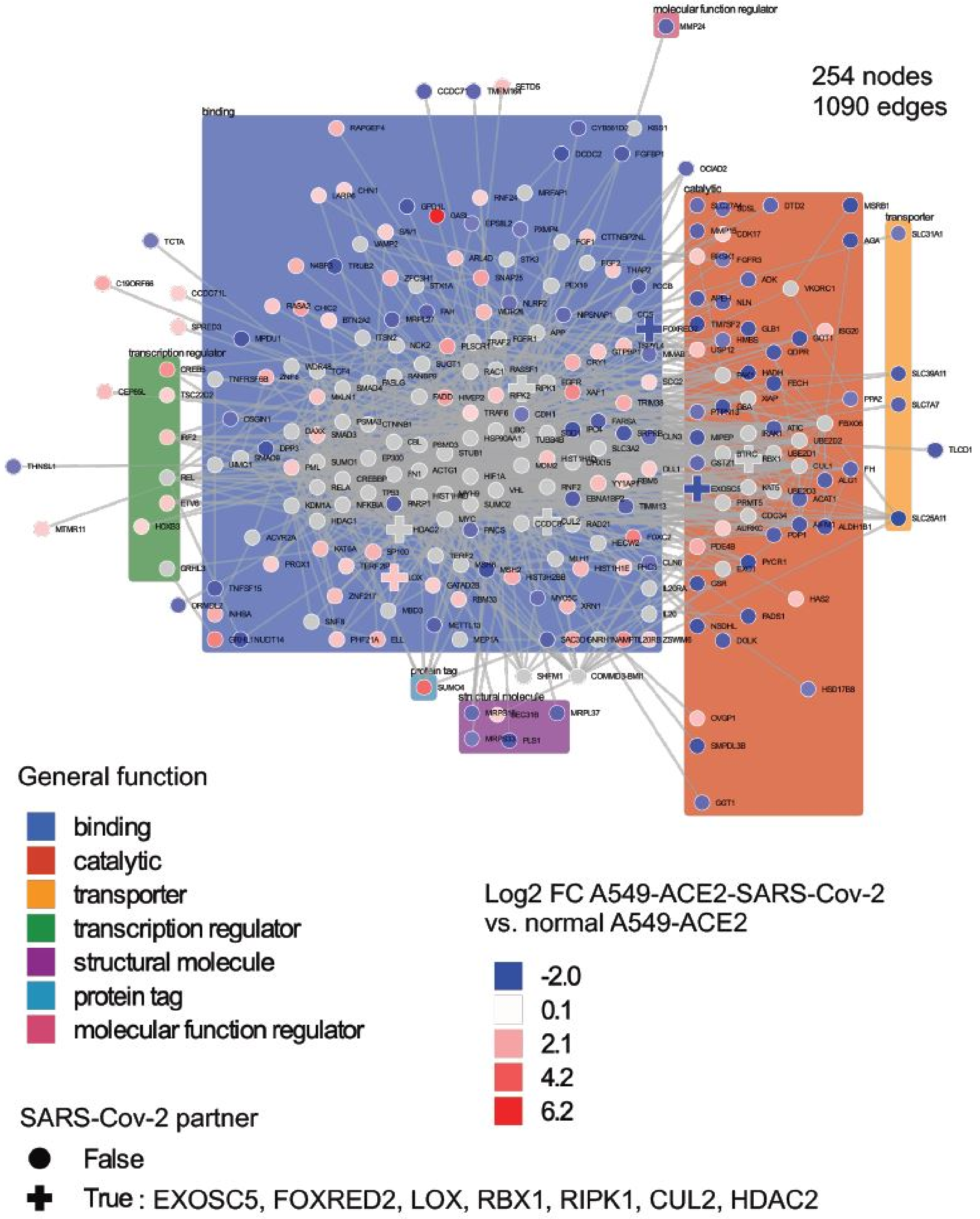


Figure E-14: Interactome obtained from the prize-collecting Steiner forest algorithm (with parameters $g = 0$, $w = 1.4$, $b = 40$) using the terminal gene list from Supplementary Fig. E-10 augmented with all other SARS-CoV-2 interaction partners prized with $p = 8 \cdot 10^{-4}$. The interactome contains 1,090 edges between 254 genes, seven of which being known SARS-CoV-2 interaction partners (EXOSC5, FOXRED2, LOX, RBX1, RIPK1, CUL2, HDAC2). Genes in the interactome are grouped by general function.

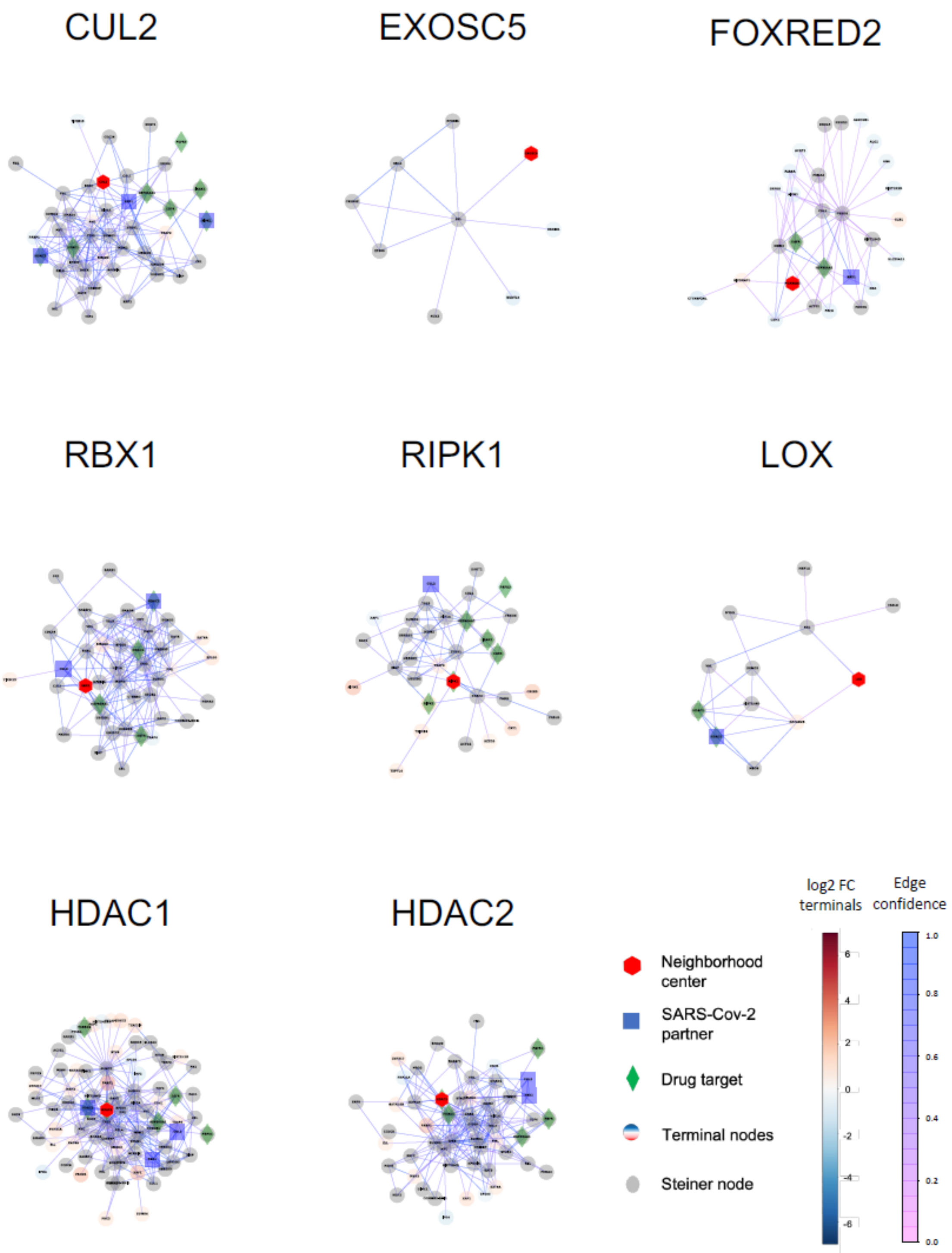
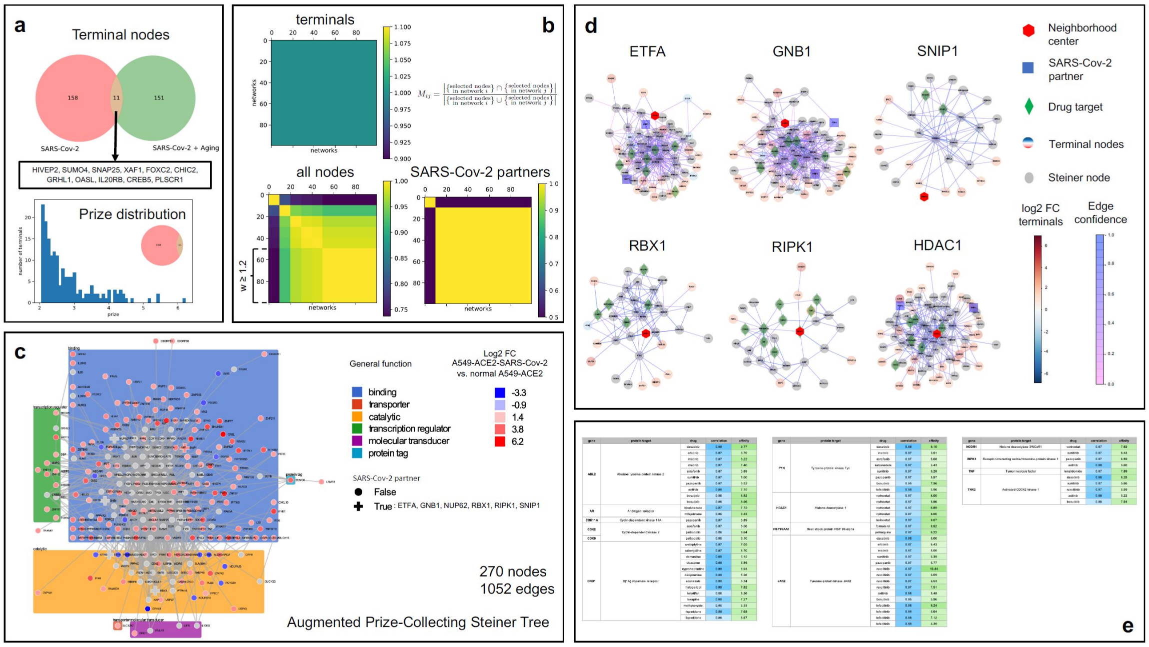


Figure E-15: 2-Nearest-Neighborhoods of nodes of interest (denoted by a red hexagon) in the interactome of Supplementary Fig. E-14 (parameters $g = 0$, $w = 1.4$, $b = 40$, $p = 8 \cdot 10^{-4}$). A threshold was applied on the edge confidence to improve legibility. Proteins known to interact with SARS-CoV-2 are denoted as blue squares, drug targets are denoted as green diamonds, terminal nodes are colored according to $\log_2$-fold change in SARS-CoV-2-infected A549-ACE2 cells versus normal A549-ACE2 cells, Steiner nodes appear in grey.



gene	protein target	drug	correlation	affinity
ABL2	Abelson tyrosine-protein kinase 2	dasatinib	0.88	9.77
		erlotinib	0.87	6.70
		imatinib	0.87	8.22
		imatinib	0.87	7.40
		sorafenib	0.87	5.89
		sunitinib	0.87	6.00
		pazopanib	0.87	5.52
		axitinib	0.88	7.15
		bosutinib	0.86	8.82
		bosutinib	0.86	8.96
AR	Androgen receptor	bicalutamide	0.87	7.72
		mifepristone	0.86	8.33
CDK11A	Cyclin-dependent kinase 11A	pazopanib	0.87	5.89
CDK2	Cyclin-dependent kinase 2	sorafenib	0.87	5.06
		palbociclib	0.86	6.64
CDK9		palbociclib	0.86	6.10
DRD1	D(1A) dopamine receptor	amitriptyline	0.87	7.05
		cabergoline	0.87	6.70
		clemastine	0.88	6.12
		clozapine	0.88	6.89
		cyproheptadine	0.88	6.93
		desipramine	0.88	5.26
		econazole	0.88	5.34
		haloperidol	0.88	7.82
		ketotifen	0.86	6.36
		loxapine	0.88	7.27
		methysergide	0.86	6.33
		risperidone	0.88	7.68
		iloperidone	0.86	6.67

gene	protein target	drug	correlation	affinity
FYN	Tyrosine-protein kinase Fyn	dasatinib	0.88	9.10
		imatinib	0.87	5.61
		sorafenib	0.87	5.08
		sulconazole	0.87	5.43
		sunitinib	0.87	6.28
		pazopanib	0.87	5.57
		bosutinib	0.86	7.96
		tofacitinib	0.88	5.96
HDAC1	Histone deacetylase 1	vorinostat	0.87	8.00
		vorinostat	0.87	6.96
		vorinostat	0.87	8.89
		vorinostat	0.87	8.00
		belinostat	0.87	9.07
HSP90AA1	Heat shock protein HSP 90-alpha	formoterol	0.87	8.52
		primaquine	0.87	8.22
JAK2	Tyrosine-protein kinase JAK2	dasatinib	0.88	6.00
		erlotinib	0.87	5.43
		imatinib	0.87	5.00
		sunitinib	0.87	6.39
		pazopanib	0.87	5.77
		ruxolitinib	0.87	10.44
		ruxolitinib	0.87	6.09
		ruxolitinib	0.87	6.53
		ruxolitinib	0.87	7.51
		axitinib	0.88	5.48
		bosutinib	0.86	5.96
		tofacitinib	0.88	9.24
		tofacitinib	0.88	6.64
		tofacitinib	0.88	7.12
		tofacitinib	0.88	6.39

gene	protein target	drug	correlation	affinity
NCOR1	Histone deacetylase 3/NCoR1	vorinostat	0.87	7.92
RIPK1	Receptor-interacting serine/threonine-protein kinase 1	sunitinib	0.87	6.43
		pazopanib	0.87	6.59
		axitinib	0.88	5.60
TNF	Tumor necrosis factor	lenalidomide	0.87	7.89
TNK2	Activated CDC42 kinase 1	dasatinib	0.88	8.25
		sunitinib	0.87	5.66
		ruxolitinib	0.87	6.59
		axitinib	0.88	5.22
		bosutinib	0.86	7.54

Figure E-16: Drug target discovery via prize-collecting Steiner forest analysis to identify putative molecular pathways linking differentially expressed genes in SARS-CoV-2 infection without taking into account age-related differential expression. (a) The general procedure to obtain the interactome is identical to the one described in Fig. 6-5a, with a different terminal gene list. (a) Terminal nodes and histogram of prize distribution. We consider 169 terminal nodes corresponding to genes differentially expressed in SARS-CoV-2 infection after removing the effect of the ACE2 receptor. Only 11 of these 169 genes belong to the terminal list used in Fig. 6-5. The prize of a terminal node equals the absolute value of its $\log_2$-fold change in SARS-CoV-2-infected A549-ACE2 cells versus normal A549-ACE2 cells based on data from (Blanco-Melo et al., 2020). (b) Sensitivity analysis to choose the parameters w and b for the prize-collecting Steiner forest algorithm. We select $g = 0$, $w = 1.4$ and $b = 40$ corresponding to a robust solution for moderate changes in the parameters. (c) Interactome obtained using the prize-collecting Steiner forest algorithm. Genes are grouped by general function and marked with a cross if known to interact with SARS-CoV-2 proteins based on data from (Gordon et al., 2020). (d) 2-Nearest-Neighborhoods of nodes of interest (denoted by a red hexagon) in the interactome. A threshold was applied on the edge confidence to improve legibility. Proteins known to interact with SARS-CoV-2 are denoted as blue squares, drug targets are denoted as green diamonds, terminal nodes are colored according to $\log_2$-fold change in SARS-CoV-2-infected A549-ACE2 cells versus normal A549-ACE2 cells, Steiner nodes appear in grey. (e) Table of drug targets in the interactome with the corresponding drugs. Selected drugs are FDA approved, high affinity (at least one of the activity constants K_i, K_d, $IC50$ or $EC50$ is below $10\mu M$), and match the SARS-CoV-2 signature well (correlation > 0.86). The affinity column displays $-\log_{10}(\text{activity})$. Protein name corresponding to each gene is included.

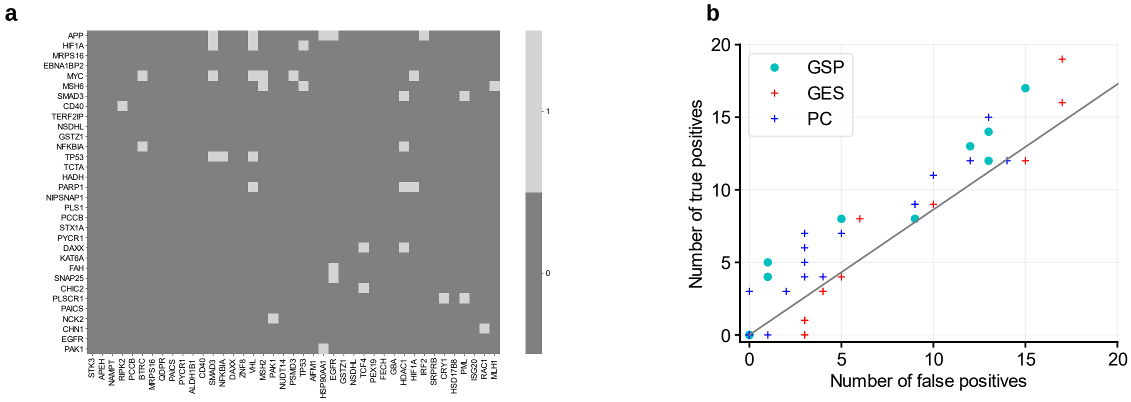


Figure E-17: (a) Matrix Q of estimated effects of interventions (columns) on measured genes (rows) in A549 cells from CMap gene knockout and overexpression data with $Q_{ij} = 1$ representing that perturbing gene j effects gene i and hence that gene i is downstream of gene j. (b) ROC curve evaluating causal structure discovery methods GSP, PC and GES for predicting the effects of interventions in A549 cells. The performance of each algorithm is measured by sampling random causal graphs and measuring number of true positives and false positives. GSP performs significantly above random guessing with p-value of 0.0177, while PC achieves p-value of 0.0694 and GES a p-value of 0.5867. The grey line represents a random guessing baseline (not used for computation of p-value) based on the number of ground truth positives and negatives, calculated from Q and scaled to extend from $(0, 0)$ to span the entirety of the plot.

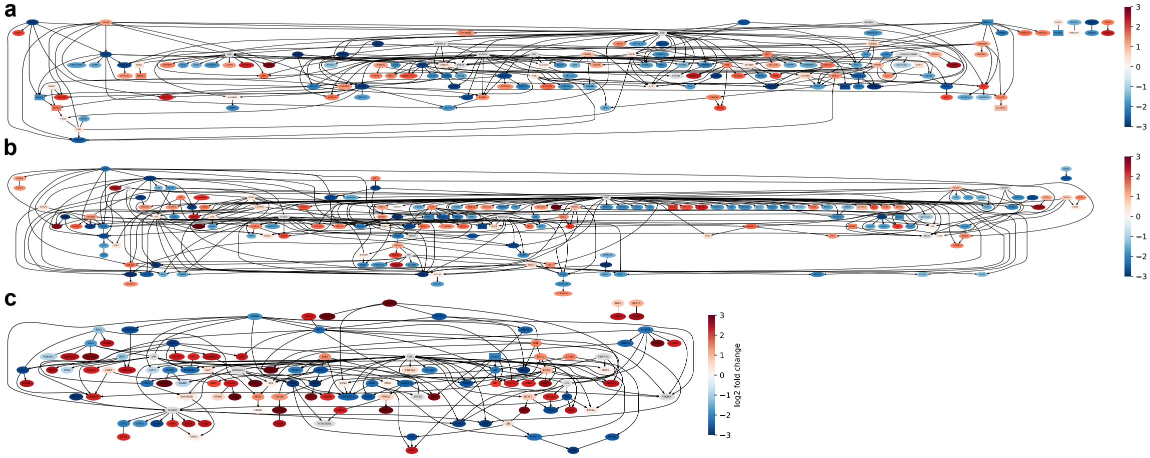


Figure E-18: (a) Causal network corresponding to A549 cells. (b) Causal network corresponding to AT2 cells. (c) Causal network corresponding to A549 cells learned using PPI interactome obtained without considering age-associated genes as a prior. All non-singleton nodes are shown, gene targets of drugs selected via our computational drug repurposing pipeline are in boxes and the node color corresponds to the $\log_2$-fold change of A549-ACE2 with versus without SARS-CoV-2.

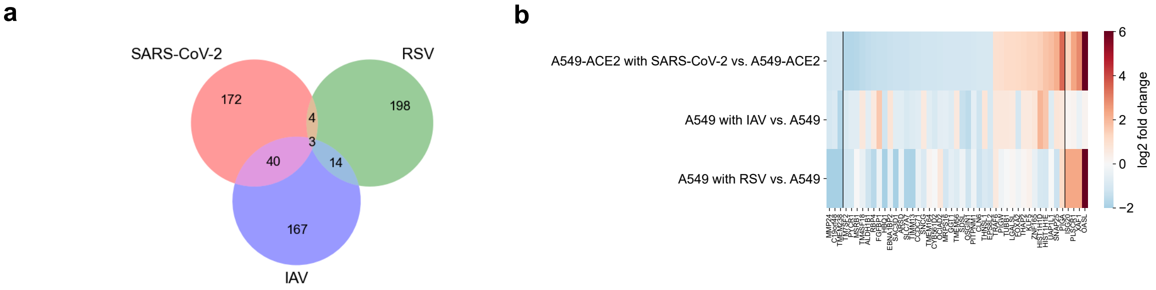


Figure E-19: (a) Venn diagram of overlap between differentially expressed genes in SARS-CoV-2, RSV and IAV infections. (b) Heatmap of $\log_2$ fold change of differentially expressed genes shared by SARS-CoV-2, IAV and RSV (first 3 genes), SARS-CoV-2 and IAV (40 genes), and SARS-CoV-2 and RSV (last 4 genes).

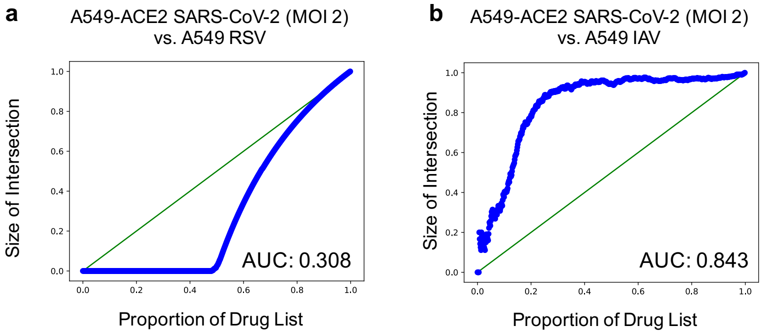


Figure E-20: Quantitative analysis of similarity between drug lists obtained using the overparameterized autoencoder on gene expression data from different virus infections. Comparison of drug lists from SARS-CoV-2 infected A549-ACE2 cells versus (a) RSV infected A549 cells, and (b) IAV infected A549 cells.

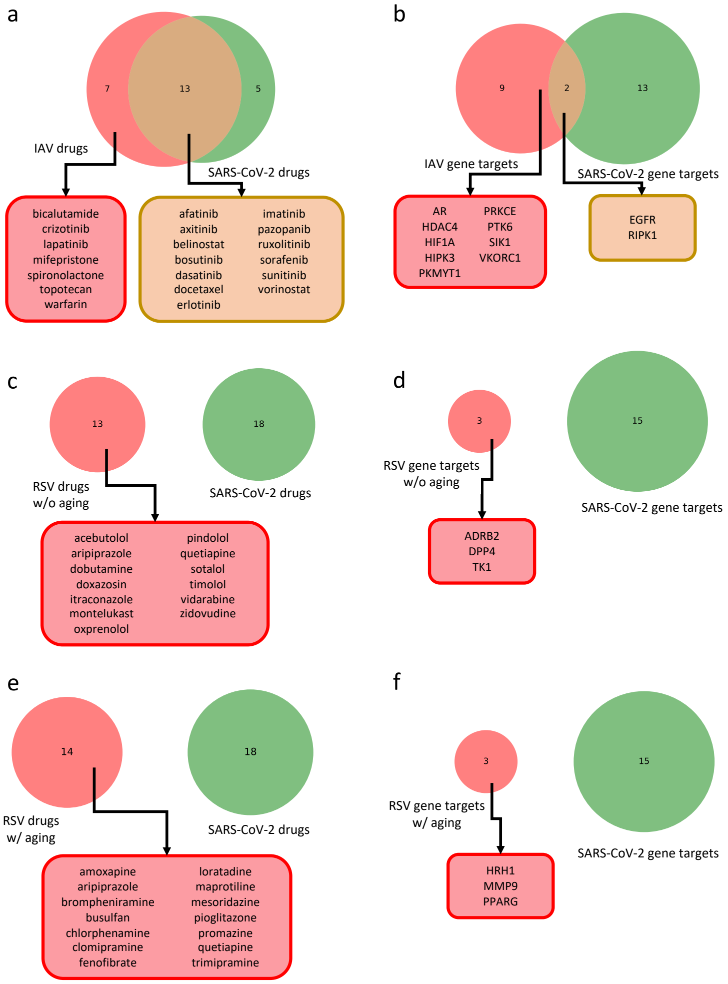


Figure E-21: Drugs and their gene targets obtained from the prize-collecting Steiner tree analysis for IAV and RSV infections in comparison to our findings for SARS-CoV-2. (a) Venn diagram between selected drugs for IAV and SARS-CoV-2 using aging as a filter in the differential gene expression analysis for both viruses, and (b) Venn diagram for the respective gene targets. (c) Venn diagram between selected drugs for RSV and SARS-CoV-2 without taking aging into account for the differential expression analysis of RSV, and (d) Venn diagram for the respective gene targets. (e) Venn diagram between selected drugs for RSV and SARS-CoV-2 using aging as a filter in the differential gene expression analysis for both viruses, and (f) Venn diagram for the respective gene targets.

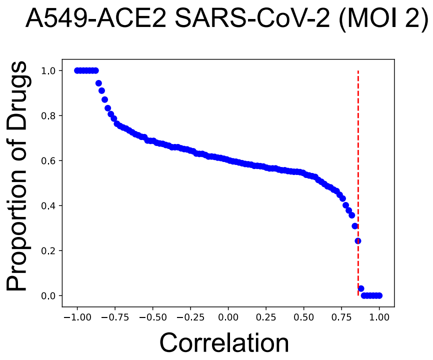


Figure E-22: Selection of correlation threshold for identifying candidate drugs. Plot showing the percentage of drugs (y-axis) with correlation higher than a given threshold (x-axis). The vertical red line indicates the x-value (0.86) for which the y-value shows the largest jump and corresponds to the threshold used for the selection of drug candidates.

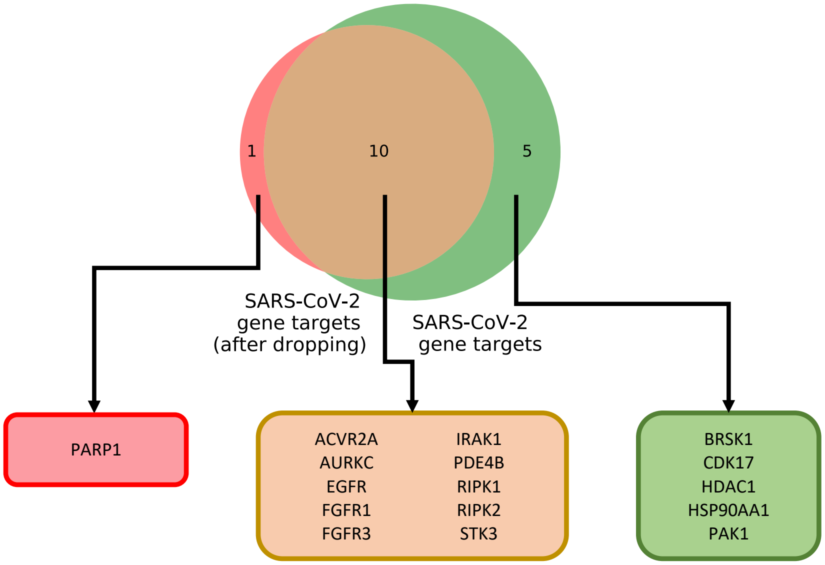


Figure E-23: Comparison of drug targets resulting from analyzing the CMap dataset with and without removing confounding 1s.

Supplementary Tables

Drug name	% differentially expressed nodes downstream (A549)	% nodes downstream (A549 no age)	% nodes downstream (AT2)
afatinib	98.51	0.00	83.93
axitinib	98.51	0.85	83.93
bosutinib	98.51	0.00	83.93
dasatinib	98.51	0.00	83.33
erlotinib	98.51	0.00	83.33
imatinib	98.51	0.00	83.93
pazopanib	98.51	0.85	83.93
ruxolitinib	98.51	0.00	83.33
sorafenib	97.01	0.00	0.60
sunitinib	98.51	0.85	83.93
tofacitinib	1.49	0.00	0.00
belinostat	98.51	94.92	83.33
vorinostat	98.51	94.92	83.33
formoterol	98.51	94.92	83.33
primaquine	98.51	94.92	83.33
vardenafil	0.00	0.00	0.00
milrinone	0.00	0.00	0.00
docetaxel	98.51	0.00	83.33

Table E.1: Percentage of nodes in the largest connected component of the corresponding causal graph that are targeted by each drug. For A549 cells, only genes that are associated with SARS-CoV-2 and aging are considered.

			Frequency of appearance in randomizations			
drug	**Selected**	**# targets in PPI**	**Gene labels**	**CMAP signatures**	**Terminal genes**	**PPI network**
sunitinib	1	260	1.0	0.25	0.997	1.0
bosutinib	1	203	0.998	0.24	0.993	1.0
axitinib	1	99	0.997	0.25	0.98	1.0
dasatinib	1	128	0.98	0.246	0.98	1.0
sorafenib	1	116	0.998	0.266	0.975	1.0
pazopanib	1	103	0.991	0.235	0.965	1.0
ruxolitinib	1	132	0.988	0.243	0.94	1.0
erlotinib	1	96	0.967	0.234	0.933	1.0
afatinib	1	38	0.94	0.226	0.863	1.0
vardenafil	1	13	0.348	0.247	0.071	1.0
milrinone	1	9	0.178	0.253	0.034	1.0
imatinib	1	69	0.947	0.238	0.921	0.971
vorinostat	1	32	0.79	0.261	0.8	0.898
belinostat	1	11	0.743	0.225	0.755	0.867
docetaxel	1	13	0.422	0.251	0.576	0.796
tofacitinib	1	43	0.481	0.243	0.58	0.709
formoterol	1	5	0.326	0.253	0.499	0.59
primaquine	1	5	0.344	0.24	0.463	0.555
palbociclib	0	13	0.924		0.741	0.863
mifepristone	0	10	0.634		0.544	0.747
vemurafenib	0	4	0.246		0.393	
danazol	0	16	0.501		0.377	
tacrolimus	0	13	0.418		0.29	
haloperidol	0	42			0.286	
bicalutamide	0	2	0.278		0.277	

clozapine	0	39			0.195	
risperidone	0	36			0.188	
sulconazole	0	25			0.186	
econazole	0	41	0.439		0.164	
amitriptyline	0	33			0.138	
clemastine	0	25			0.103	
dipyridamole	0	19	0.353		0.103	
phentolamine	0	17			0.095	
iloperidone	0	24			0.092	
methysergide	0	22			0.092	
cyproheptadine	0	29			0.09	
carteolol	0	2			0.083	
lenalidomide	0	2			0.083	
cabergoline	0	17			0.079	
loxapine	0	29			0.079	
digitoxin	0	9			0.076	
terconazole	0	17	0.198		0.069	
ketotifen	0	17			0.065	
desipramine	0	22			0.054	
rosuvastatin	0	2			0.054	
perphenazine	0	16			0.053	
naftifine	0	2			0.05	
desoximetasone	0	1			0.048	
flunisolide	0	1			0.048	
halcinonide	0	1			0.048	
irinotecan	0	7			0.048	
phenelzine	0	10			0.048	
prednisone	0	2			0.048	

buspirone	0	13			0.046	
guanfacine	0	8			0.043	
terazosin	0	7			0.039	
sertraline	0	19			0.038	
flumazenil	0	36			0.037	
daunorubicin	0	1			0.036	
bortezomib	0	15		0.241		
caffeine	0	3	0.324			
cisplatin	0	10		0.234		
clofarabine	0	2	0.216			
dobutamine	0	23		0.226		
famotidine	0	3		0.24		
gefitinib	0	72		0.232		
glimepiride	0	4	0.18			
iloprost	0	8	0.206			
lapatinib	0	13		0.256		
midodrine	0	1		0.254		
mitoxantrone	0	18		0.23		
montelukast	0	21		0.251		
nilotinib	0	70		0.292		
olaparib	0	4		0.261		
panobinostat	0	11		0.247		
sildenafil	0	20		0.233		
sitagliptin	0	2	0.183			
tamoxifen	0	51		0.278		
tolbutamide	0	2	0.18			
topotecan	0	5		0.277		
treprostinil	0	6	0.206			

warfarin	0	1		0.254		
zafirlukast	0	13		0.238		

Table E.2: Frequency of a drug's presence in the list of final drugs after performing Steiner tree analysis with randomization of gene labels, CMap signatures, terminal genes, and the PPI network (1000 randomization runs).

Bibliography

A. Abadie, A. Diamond, and J. Hainmueller. Synthetic control methods for comparative case studies: Estimating the effect of california's tobacco control program. *Journal of the American Statistical Association*, 105(490):493–505, 2010.

A. Agarwal, R. Cosson, D. Shah, and D. Shen. Synthetic interventions. In *Proceedings of CausalML NeurIPS Workshop*, 2019.

A. Y. Alfakih, A. Khandani, and H. Wolkowicz. Solving Euclidean distance matrix completion problems via semidefinite programming. *Computational Optimization and Applications*, 12(1):13–30, 1999.

M. Amodio and S. Krishnaswamy. MAGAN: Aligning biological manifolds. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 215–223, 2018.

M. Amodio and S. Krishnaswamy. TraVeLGAN: Image-to-image translation by transformation vector learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8983–8992, 2019.

M. Amodio, D. Van Dijk, K. Srinivasan, W. S. Chen, H. Mohsen, K. R. Moon, A. Campbell, Y. Zhao, X. Wang, M. Venkataswamy, et al. Exploring single-cell data with deep multitasking neural networks. *Nature Methods*, 16:1139–1145, 2019.

G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 1247–1255, 2013.

I. Angelidis, L. M. Simon, I. E. Fernandez, M. Strunz, C. H. Mayr, F. R. Greiffo, G. Tsitsiridis, M. Ansari, E. Graf, T.-M. Strom, et al. An atlas of the aging lung mapped by single cell transcriptomics and deep tissue proteomics. *Nature Communications*, 10(1):1–17, 2019.

E. Apostolou and D. Thanos. Virus Infection Induces NF-κB-Dependent Interchromosomal Associations Mediating Monoallelic IFNβ Gene Expression. *Cell*, 134(1): 85–96, 2008.

R. Argelaguet, B. Velten, D. Arnol, S. Dietrich, T. Zenz, J. C. Marioni, F. Buettner, W. Huber, and O. Stegle. Multi-omics factor analysis — a framework for unsupervised integration of multi-omics data sets. *Molecular Systems Biology*, 14(6):e8124, 2018.

P. Baldi. Autoencoders, unsupervised learning, and deep architectures. In *ICML Workshop on Unsupervised and Transfer Learning*, volume 27, pages 37–49, 2012.

F. Bantignies, V. Roure, I. Comet, B. Leblanc, B. Schuettengruber, J. Bonnet, V. Tixier, A. Mas, and G. Cavalli. Polycomb-dependent regulatory contacts between distant hox loci in drosophila. *Cell*, 144(2):214–226, 2011.

A. Barski, S. Cuddapah, K. Cui, T. Y. Roh, D. E. Schones, Z. Wang, G. Wei, I. Chepelev, and K. Zhao. High-Resolution Profiling of Histone Methylations in the Human Genome. *Cell*, 129(4):823–837, 2007.

R. A. Beagrie, A. Scialdone, M. Schueler, D. C. A. Kraemer, M. Chotalia, S. Q. Xie, M. Barbieri, I. de Santiago, L.-M. Lavitas, M. R. Branco, et al. Complex multi-enhancer contacts captured by genome architecture mapping. *Nature*, 543(7646): 519–524, 2017.

W. A. Bickmore and B. Van Steensel. Genome architecture: Domain organization of interphase chromosomes. *Cell*, 152(6):1270–1284, 2013.

D. Blanco-Melo, B. E. Nilsson-Payant, W.-C. Liu, S. Uhl, D. Hoagland, R. Møller, T. X. Jordan, K. Oishi, M. Panis, D. Sachs, et al. Imbalanced host response to SARS-CoV-2 drives development of COVID-19. *Cell*, 2020.

A. Boija, I. A. Klein, B. R. Sabari, A. Dall'Agnese, E. L. Coffey, A. V. Zamudio, C. H. Li, K. Shrinivas, J. C. Manteiga, N. M. Hannett, et al. Transcription factors activate genes through the phase-separation capacity of their activation domains. *Cell*, 175(7):1842–1855, 2018.

A. Bolzer, G. Kreth, I. Solovei, D. Koehler, K. Saracoglu, C. Fauth, S. Müller, R. Eils, C. Cremer, M. R. Speicher, and T. Cremer. Three-dimensional maps of all chromosomes in human male fibroblast nuclei and prometaphase rosettes. *PLoS Biology*, 3(5):e157, 2005.

B. Bonev and C. Giacomo. Organization and function of the 3D genome. *Nature Review Genetics*, 17:661 – 678, 2016.

B. Bonev, N. M. Cohen, Q. Szabo, L. Fritsch, G. L. Papadopoulos, Y. Lubling, X. Xu, X. Lv, J.-P. Hugnot, A. Tanay, et al. Multiscale 3D genome rewiring during mouse neural development. *Cell*, 171(3):557–572, 2017.

A. P. Boyle, S. Davis, H. P. Shulha, P. Meltzer, E. H. Margulies, Z. Weng, T. S. Furey, and G. E. Crawford. High-resolution mapping and characterization of open chromatin across the genome. *Cell*, 132(2):311–322, 2008.

M. R. Branco and A. Pombo. Intermingling of chromosome territories in interphase suggests role in translocations and transcription-dependent associations. *PLoS Biology*, 4(5):e138, 2006.

J. D. Buenrostro, P. G. Giresi, L. C. Zaba, H. Y. Chang, and W. J. Greenleaf. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, dna-binding proteins and nucleosome position. *Nature Methods*, 10(12):1213–1218, 2013.

J. D. Buenrostro, B. Wu, U. M. Litzenburger, D. Ruff, M. L. Gonzales, M. P. Snyder, H. Y. Chang, and W. J. Greenleaf. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature*, 523(7561):486–490, 2015.

A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36(5):411—-420, 2018.

J. Camps, M. R. Erdos, and T. Ried. The role of lamin B1 for the maintenance of nuclear structure and function. *Nucleus*, 6(1):8–14, 2015.

J. Cao, D. A. Cusanovich, V. Ramani, D. Aghamirzaie, H. A. Pliner, A. J. Hill, R. M. Daza, J. L. McFaline-Figueroa, J. S. Packer, L. Christiansen, et al. Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science*, 361(6409):1380–1385, 2018.

D. Capurso, H. Bengtsson, and M. R. Segal. Discovering hotspots in functional genomic data superposed on 3D chromatin configuration reconstructions. *Nucleic Acids Research*, 44(5):2028–2035, 2016.

L. J. Carithers, K. Ardlie, M. Barcus, P. A. Branton, et al. A novel approach to high-quality postmortem tissue procurement: the GTEx project. *Biopreservation and Biobanking*, 13(5):311–319, 2015.

A. G. Cauer, G. Yardimci, J.-P. Vert, N. Varoquaux, and W. S. Noble. Inferring diploid 3D chromatin structures from hi-c data. In *19th International Workshop on Algorithms in Bioinformatics (WABI 2019)*, 2019.

L. Cayton and S. Dasgupta. Robust Euclidean embedding. In *Proceedings of the 23rd International Conference on Machine Learning*.

V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

C. Cheadle, M. Nesterova, T. Watkins, K. C. Barnes, J. C. Hall, A. Rosen, K. G. Becker, and Y. S. Cho-Chung. Regulatory subunits of PKA define an axis of cellular proliferation/differentiation in ovarian cancer cells. *BMC Medical Genomics*, 1(1): 43, 2008.

H. Chen, J. Chen, L. a. Muir, S. Ronquist, W. Meixner, M. Ljungman, T. Ried, S. Smale, and I. Rajapakse. Functional organization of the human 4D Nucleome. *Proceedings of the National Academy of Sciences*, 112(26):201505822, 2015.

F. Chiaradonna, C. Balestrieri, D. Gaglio, and M. Vanoni. RAS and PKA pathways in cancer: new insight from transcriptional analysis. *Frontiers in Bioscience*, 13: 5257–5278, 2008.

D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov):507–554, 2002.

D. M. Chickering and C. Meek. Selective greedy equivalence search: Finding optimal bayesian networks using a polynomial number of score evaluations. Preprint at `https://arxiv.org/abs/1506.02113`, 2015.

Y. S. Cho-Chung. Antisense oligonucleotide inhibition of serine/threonine kinases: an innovative approach to cancer treatment. *Pharmacology & Therapeutics*, 82(2): 437–449, 1999.

S. Chong, C. Dugast-Darzacq, Z. Liu, P. Dong, G. M. Dailey, C. Cattoglio, A. Heckert, S. Banala, L. Lavis, X. Darzacq, et al. Imaging dynamic and selective low-complexity domain interactions that control gene transcription. *Science*, 361(6400), 2018.

R. D. Chow and S. Chen. The aging transcriptome and cellular landscape of the human lung in relation to SARS-CoV-2. *https://doi.org/10.1101/2020.04.07.030684*, 2020.

T. F. Cox and M. A. A. Cox. *Multidimensional scaling*. Chapman and Hall / CRC, 2000.

L. Crabbe, A. J. Cesare, J. M. Kasuboski, J. A. J. Fitzpatrick, and J. Karlseder. Human telomeres are tethered to the nuclear envelope during postmitotic nuclear assembly. *Cell Reports*, 2(6):1521–1529, 2012.

J. A. Croft, J. M. Bridger, S. Boyle, P. Perry, P. Teague, and W. A. Bickmore. Differences in the localization and morphology of chromosomes in the human nucleus. *Journal of Cell Biology*, 145(6):1119–1131, 1999.

D. A. Cusanovich, R. Daza, A. Adey, H. A. Pliner, L. Christiansen, K. L. Gunderson, F. J. Steemers, C. Trapnell, and J. Shendure. Multiplex single-cell profiling of chromatin accessibility by combinatorial cellular indexing. *Science*, 348(6237):910–914, 2015.

M. Dannappel, K. Vlantis, S. Kumari, A. Polykratis, C. Kim, L. Wachsmuth, C. Eftychi, J. Lin, T. Corona, N. Hermance, M. Zelic, P. Kirsch, M. Basic, A. Bleich, M. Kelliher, and M. Pasparakis. RIPK1 maintains epithelial homeostasis by inhibiting apoptosis and necroptosis. *Nature*, 513(7516):90–94, 2014.

C. Das, M. S. Lucia, K. C. Hansen, and J. K. Tyler. CBP / p300-mediated acetylation of histone H3 on lysine 56. *Nature*, 459(7243):113–117, 2009.

P. Datlinger et al. Pooled CRISPR screening with single-cell transcriptome readout. *Nature Methods*, 14(3):297, 2017.

J. De Las Rivas and C. Fontanillo. Protein–protein interactions essentials: Key concepts to building and analyzing interactome networks. *PLoS Computational Biology*, 6(6):e1000807, 2010.

E. De Wit, B. A. Bouwman, Y. Zhu, P. Klous, E. Splinter, M. J. Verstegen, P. H. Krijger, N. Festuccia, E. P. Nora, M. Welling, et al. The pluripotent genome in three dimensions is shaped around pluripotency factors. *Nature*, 501(7466):227–231, 2013.

E. de Wit, N. van Doremalen, D. Falzarano, and V. J. Munster. SARS and MERS: recent insights into emerging coronaviruses. *Nature Reviews Microbiology*, 14(8): 523, 2016.

A. Degterev, D. Ofengeim, and J. Yuan. Targeting RIPK1 for the treatment of human diseases. *Proceedings of the National Academy of Sciences, U.S.A*, 116(20): 9714–9722, 2019.

J. Dekker. Gene regulation in the third dimension. *Science*, 319(5871):1793–1794, 2008.

J. Dekker and L. Mirny. The 3D Genome as Moderator of Chromosomal Communication. *Cell*, 164(6):1110–1121, 2016.

J. Dekker and T. Misteli. Long-Range Chromatin Interactions. *Cold Spring Harbor perspectives in biology*, 7(10):a019356, 2015.

J. Dekker, K. Rippe, M. Dekker, and N. Kleckner. Capturing chromosome conformation. *Science*, 295(5558):1306–1311, 2002.

M. Di Pierro, B. Zhang, E. L. Aiden, P. G. Wolynes, and J. N. Onuchic. Transferable model for chromosome architecture. *Proceedings of the National Academy of Sciences*, 113(43):12168–12173, 2016.

J. Ding, X. Adiconis, S. K. Simmons, M. S. Kowalczyk, C. C. Hession, N. D. Marjanovic, T. K. Hughes, M. H. Wadsworth, T. Burks, L. T. Nguyen, et al. Systematic comparative analysis of single cell RNA-sequencing methods. Preprint at `https://doi.org/10.1101/632216`, 2019.

A. Dixit et al. Perturb-seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell*, 167(7):1853–1866, 2016.

J. R. Dixon, S. Selvaraj, F. Yue, A. Kim, Y. Li, Y. Shen, M. Hu, J. S. Liu, and B. Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 485(7398):376–380, 2012.

J. R. Dixon, I. Jung, S. Selvaraj, Y. Shen, J. E. Antosiewicz-Bourget, A. Y. Lee, Z. Ye, A. Kim, N. Rajagopal, W. Xie, et al. Chromatin architecture reorganization during stem cell differentiation. *Nature*, 518(7539):331–336, 2015.

Z. Duan, M. Andronescu, K. Schutz, S. McIlwain, Y. J. Kim, C. Lee, J. Shendure, S. Fields, C. A. Blau, and W. S. Noble. A three-dimensional model of the yeast genome. *Nature*, 465:363–367, 2010.

J. T. Dudley, T. Deshpande, and A. T. Butte. Exploiting drug–disease relationships for computational drug repositioning. *Briefings in Bioinformatics*, 12(4):303—311, 2011.

I. Dunham, E. Birney, B. R. Lajoie, A. Sanyal, X. Dong, M. Greven, X. Lin, J. Wang, T. W. Whitfield, J. Zhuang, et al. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012.

N. C. Durand, J. T. Robinson, M. S. Shamim, I. Machol, J. P. Mesirov, E. S. Lander, and E. L. Aiden. Juicebox Provides a Visualization System for Hi-C Contact Maps with Unlimited Zoom. *Cell Systems*, 3(1):99–101, 2016.

Z. Duren, X. Chen, R. Jiang, Y. Wang, and W. H. Wong. Modeling gene regulation from paired expression and chromatin accessibility data. *Proceedings of the National Academy of Sciences*, 114(25):E4914–E4923, 2017.

F. Eberhardt. *Causation and Intervention.* PhD thesis, Department of Philosophy, Carnegie Mellon University, 2007.

M. Elsner and W. Schudy. Bounding and comparing methods for correlation clustering beyond ILP. *Proceedings of the NAACL HLT Workshop on Integer Linear Programming for Natural Language Processing*, (June):19–27, 2009.

S. J. Emrich, W. B. Barbazuk, L. Li, and P. S. Schnable. Gene discovery and annotation using LCM-454 transcriptome sequencing. *Genome Research*, 17(1):69–73, 2007.

F. Erdel and K. Rippe. Formation of chromatin subcompartments by phase separation. *Biophysical Journal*, 114(10):2262–2270, 2018.

H. Fang and D. P. O'Leary. Euclidean distance matrix completion problems. *Optimization Methods and Software*, 27(4-5):695–717, 2012.

M. Fazel, H. Hindi, S. P. Boyd, et al. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, volume 6, pages 4734–4739, 2001.

M. Fazel, H. Hindi, and S. P. Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proceedings of the 2003 American Control Conference*, volume 3, pages 2156–2162, 2003.

N. Festjens, T. V. Berghe, S. Cornelis, and P. Vandenabeele. RIP1, a kinase on the crossroads of a cell's decision to live or die. *Cell Death & Differentiation*, 14: 400–410, 2007.

L. E. Finlan, D. Sproul, I. Thomson, S. Boyle, E. Kerr, P. Perry, B. Ylstra, J. R. Chubb, and W. A. Bickmore. Recruitment to the nuclear periphery can alter expression of genes in human cells. *PLoS Genetics*, 4(3):e1000039, 2008.

E. H. Finn, G. Pegoraro, H. B. Brandão, A.-L. Valton, M. E. Oomen, J. Dekker, L. Mirny, and T. Misteli. Extensive heterogeneity and intrinsic variation in spatial genome organization. *Cell*, 176(6):1502–1515, 2019.

A. R. Forrest, H. Kawaji, M. Rehli, J. K. Baillie, M. J. De Hoon, V. Haberle, T. Lassmann, I. V. Kulakovskiy, M. Lizio, M. Itoh, et al. A promoter-level mammalian expression atlas. *Nature*, 507(7493):462–470, 2014.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, 7(3-4):601–620, 2000.

A. Fukushima. DiffCorr: an R package to analyze and visualize differential correlations in biological networks. *Gene*, 518(1):209–214, 2013.

T. S. Fung and D. X. Liu. Human coronavirus: Host-pathogen interaction. *Annual Review of Microbiology*, 73:529–557, 2019.

A. Ghahramani, F. M. Watt, and N. M. Luscombe. Generative adversarial networks simulate gene expression and predict perturbations in single cells. *bioRxiv, https://doi.org/10.1101/262501*, 2018.

C. Glymour, K. Zhang, and P. Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10:524, 2019.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, volume 1. MIT Press, 2016.

D. E. Gordon, G. M. Jang, M. Bouhaddou, J. Xu, K. Obernier, K. M. White, M. J. O'Meara, V. V. Rezelj, J. Z. Guo, D. L. Swaney, et al. A SARS-CoV-2 protein interaction map reveals targets for drug repurposing. *Nature*, 583:459–468, 2020.

C. S. Greene and B. F. Voight. Pathway and network-based strategies to translate genetic discoveries into effective therapies. *Human Molecular Genetics*, 25(R2): R94–R98, 2016.

L. Guelen, L. Pagie, E. Brasset, W. Meuleman, M. B. Faza, W. Talhout, B. H. Eussen, A. de Klein, L. Wessels, W. de Laat, and B. Van Steensel. Domain organization of human chromosomes revealed by mapping nuclear lamina interactions. *Nature*, 453(7197):948–951, 2008a.

L. Guelen, L. Pagie, E. Brasset, W. Meuleman, M. B. Faza, W. Talhout, B. H. Eussen, A. de Klein, L. Wessels, W. de Laat, et al. Domain organization of human chromosomes revealed by mapping of nuclear lamina interactions. *Nature*, 453 (7197):948–951, 2008b.

G. Gundersen, B. Dumitrascu, J. T. Ash, and B. E. Engelhardt. End-to-end training of deep probabilistic cca on paired biomedical observations. In *35th Conference on Uncertainty in Artificial Intelligence*, 2019.

S. Gupta, N. Marcel, S. Talwar, M. Garg, R. Indulaxmi, L. R. Perumalsamy, A. Sarin, and G. V. Shivashankar. Developmental heterogeneity in dna packaging patterns influences t-cell activation and transmigration. *PloS One*, 7(9):e43718, 2012.

L. Haghverdi, A. T. Lun, M. D. Morgan, and J. C. Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature Biotechnology*, 36(5):421—-427, 2018.

G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

R. Hodos, P. Zhang, H.-C. Lee, Q. Duan, Z. Wang, N. R. Clark, A. Ma'ayan, F. Wang, B. Kidd, J. Hu, et al. Cell-specific prediction and application of drug-induced gene expression profiles. In *Pac Symp Biocomput*, volume 23, pages 32–43. World Scientific, 2018.

M. Hoffmann, H. Kleine-Weber, S. Schroeder, N. Krüger, T. Herrler, S. Erichsen, T. S. Schiergens, G. Herrler, N. Wu, A. Nitsche, M. A. Müller, C. Drosten, and S. Pöhlmann. SARS-CoV-2 cell entry depends on ACE2 and TMPRSS2 and is blocked by a clinically proven protease inhibitor. *Cell*, 181(2):271–280.e8, 2020.

G. Hu, K. Cui, D. Fang, S. Hirose, X. Wang, D. Wangsa, W. Jin, T. Ried, P. Liu, J. Zhu, et al. Transformation of accessible chromatin and 3D nucleome underlies lineage commitment of early t cells. *Immunity*, 48(2):227–242, 2018.

M. Hu, K. Deng, Z. Qin, J. Dixon, S. Selvaraj, J. Fang, B. Ren, and J. S. Liu. Bayesian inference of spatial organizations of chromosomes. *PLoS Computational Biology*, 9(1):e1002893, 2013.

D. W. Huang, R. a. Lempicki, and B. T. Sherman. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols*, 4(1): 44–57, 2009a.

D. W. Huang, B. T. Sherman, and R. A. Lempicki. Bioinformatics enrichment tools: Paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research*, 37(1):1–13, 2009b.

S. S. Huang and E. Fraenkel. Integrating proteomic, transcriptional, and interactome data reveals hidden components of signaling and regulatory networks. *Science Signaling*, 2(81):ra40, 2009.

J. R. Hughes, N. Roberts, S. McGowan, D. Hay, E. Giannoulatou, M. Lynch, M. De Gobbi, S. Taylor, R. Gibbons, and D. R. Higgs. Analysis of hundreds of cis-regulatory landscapes at high resolution in a single, high-throughput experiment. *Nature Genetics*, 46:205–212, 2014.

M. V. Imakaev, G. Fudenberg, and L. A. Mirny. Modeling chromosomes: Beyond pretty pictures. *FEBS Letters*, 589(20):3031–3036, 2015.

J. M. Irish, N. Kotecha, and G. P. Nolan. Mapping normal and cancer cell signalling networks: towards single-cell proteomics. *Nature Reviews Cancer*, 6(2):146—-155, 2006.

K. V. Iyer, S. Maharana, S. Gupta, A. Libchaber, T. Tlusty, and G. V. Shivashankar. Modeling and Experimental Methods to Probe the Link between Global Transcription and Spatial Organization of Chromosomes. *PLoS ONE*, 7(10):e46628, 2012.

W. Jin, Q. Tang, M. Wan, K. Cui, Y. Zhang, G. Ren, B. Ni, J. Sklar, T. M. Przytycka, R. Childs, et al. Genome-wide detection of dnase i hypersensitive sites in single cells and ffpe tissue samples. *Nature*, 528(7580):142–146, 2015.

D. S. Johnson, A. Mortazavi, R. M. Myers, and B. Wold. Genome-wide mapping of in vivo protein-DNA interactions. *Science*, 316(5830):1497–1502, 2007.

R. Johnstone, A. Frew, and M. Smyth. The TRAIL apoptotic pathway in cancer onset, progression and therapy. *Nature Reviews Cancer*, 8(10):782–798, 2008.

J. Jönsson, K. Bartuma, M. Dominguez-Valentin, K. Harbst, Z. Ketabi, S. Malander, M. Jönsson, A. Carneiro, A. Måsbäck, G. Jönsson, and M. Nilbert. Distinct gene expression profiles in ovarian cancer linked to Lynch syndrome. *Familial Cancer*, 13:537–545, 2014.

R. J. Jose and A. Manuel. COVID-19 cytokine storm: the interplay between inflammation and coagulation. *The Lancet Respiratory Medicine*, 2020.

R. Jungmann, M. S. Avendaño, J. B. Woehrstein, M. Dai, W. M. Shih, and P. Yin. Multiplexed 3D cellular super-resolution imaging with DNA-PAINT and Exchange-PAINT. *Nature Methods*, 11(3):313, 2014.

R. Kalhor, H. Tjong, N. Jayathilaka, F. Alber, and L. Chen. Genome architectures revealed by tethered chromosome conformation capture and population-based modeling. *Nature Biotechnology*, 30(1):90–98, 2012.

M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, and T. Mao. KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, 40 (D1):D109–D114, 2011.

A. J. Kedaigle. *Integrating Omics data: a new software tool and its use in implicating therapeutic targets in Huntington's disease*. PhD thesis, Massachusetts Institute of Technology, 2018.

A. M. Klein, L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D. A. Weitz, and M. W. Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, 2015.

S. L. Klemm, Z. Shipony, and W. J. Greenleaf. Chromatin accessibility and the regulatory epigenome. *Nature Reviews Genetics*, 20(4):207–220, 2019.

J. H. Korhonen, K. Palin, J. Taipale, and E. Ukkonen. Fast motif matching revisited: high-order PWMs, SNPs and indels. *Bioinformatics*, 33(4):514–521, 2016.

E. J. Kort and S. Jovinge. Streamlined analysis of lincs l1000 data with the slinky package for r. *Bioinformatics*, 35(17):3176–3177, 2019.

N. Krislock. *Semidefinite facial reduction for low-rank Euclidean distance matrix completion*. PhD thesis, University of Waterloo, 2010.

A. Kundaje, W. Meuleman, J. Ernst, M. Bilenky, A. Yen, A. Heravi-Moussavi, P. Kheradpour, Z. Zhang, J. Wang, M. J. Ziller, et al. Integrative analysis of 111 reference human epigenomes. *Nature*, 518(7539):317–330, 2015.

X. Lan, H. Witt, K. Katsumura, Z. Ye, Q. Wang, E. H. Bresnick, P. J. Farnham, and V. X. Jin. Integration of Hi-C and ChIP-seq data reveals distinct types of chromatin linkages. *Nucleic Acids Research*, 40(16):7690–7704, 2012.

E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, et al. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.

P. Langfelder and S. Horvath. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics*, 9(1):559, 2008.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

A. Lesne, J. Riposo, P. Roger, A. Cournac, and J. Mozziconacci. 3D genome reconstruction from chromosomal contacts. *Nature Methods*, 11(11):1141–1143, 2014.

H. Li, F. Liu, C. Ren, X. Bo, and W. Shu. Genome-wide identification and characterisation of HOT regions in the human genome. *BMC Genomics*, 17(1):733, 2016.

H. Li, E. T. Courtois, D. Sengupta, Y. Tan, K. H. Chen, J. J. L. Goh, S. L. Kong, C. Chua, L. K. Hon, W. S. Tan, et al. Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors. *Nature Genetics*, 49(5):708, 2017.

Y. Lichtblau et al. Comparative assessment of differential network analysis methods. *Briefings in Bioinformatics*, 18(5):837–850, 2017.

E. Lieberman-Aiden, N. L. v. Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. R. Lajoie, P. J. Sabo, M. O. Dorschner, R. Sandstrom, B. Bernstein, M. A. Bender, M. Groudine, A. Gnirke, J. Stamatoyannopoulos, L. A. Mirny, E. S. Lander, and J. Dekker. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, 326(5950): 289–293, 2009.

R. Lister, R. C. O'Malley, J. Tonti-Filippini, B. D. Gregory, C. C. Berry, A. H. Millar, and J. R. Ecker. Highly integrated single-base resolution maps of the epigenome in arabidopsis. *Cell*, 133(3):523–536, 2008.

J. Liu, Y. Huang, R. Singh, J.-P. Vert, and W. S. Noble. Jointly embedding multiple single-cell omics measurements. In *19th International Workshop on Algorithms in Bioinformatics (WABI 2019)*, pages 10:1–10:13.

R. Liu, A. Balsubramani, and J. Zou. Learning transport cost from subset correspondence. In *International Conference on Learning Representations*, 2020.

S. Liu, J. A. Quinn, M. U. Gutmann, T. Suzuki, and M. Sugiyama. Direct learning of sparse changes in Markov networks by density ratio estimation. *Neural Computation*, 26(6):1169–1197, 2014.

S. Liu, K. Fukumizu, and T. Suzuki. Learning sparse structural changes in high-dimensional Markov networks. *Behaviormetrika*, 44(1):265–286, 2017.

S. Lomvardas, G. Barnea, D. J. Pisapia, M. Mendelsohn, J. Kirkland, and R. Axel. Interchromosomal interactions and olfactory receptor choice. *Cell*, 126(2):403–413, 2006.

R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, 2018.

R. Lopez, A. Nazaret, M. Langevin, J. Samaran, J. Regier, M. I. Jordan, and N. Yosef. A joint model of unpaired data from scRNA-seq and spatial transcriptomics for imputing missing gene expression measurements. In *ICML workshop in Computational Biology*, 2019.

M. Lotfollahi, F. A. Wolf, and F. J. Theis. scGen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721, 2019.

F. Lu, S. Keleş, S. J. Wright, and G. Wahba. Framework for kernel regularization with application to protein clustering. *Proceedings of the National Academy of Sciences*, 102(35):12332–12337, 2005.

H. Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer Science & Business Media, 2005.

E. Z. Macosko, A. Basu, R. Satija, J. Nemesh, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015.

S. Maharana, K. V. Iyer, N. Jain, M. Nagarajan, Y. Wang, and G. V. Shivashankar. Chromosome intermingling-the physical basis of chromosome organization in differentiated cells. *Nucleic Acids Research*, 44(11):5148–5160, 2016.

A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. Preprint at `https://arxiv.org/abs/1511.05644`, 2015.

D. Marbach, D. Lamparter, G. Quon, M. Kellis, Z. Kutalik, and S. Bergmann. Tissue-specific regulatory circuits reveal variable modular perturbations across complex diseases. *Nature Methods*, 13(4):366–370, 2016.

S. Marchini, R. Fruscio, L. Clivio, L. Beltrame, L. Porcu, I. F. Nerini, D. Cavalieri, G. Chiorino, G. Cattoretti, C. Mangioni, et al. Resistance to platinum-based chemotherapy is associated with epithelial to mesenchymal transition in epithelial ovarian cancer. *European Journal of Cancer*, 49(2):520–530, 2013.

S. Martens, S. Hofmans, W. Declercq, K. Augustyns, and P. Vandenabeele. Inhibitors targeting RIPK1/RIPK3: Old and new drugs. *Trends in Pharmacological Sciences*, 41(3):209–224, 2020.

L. McInnes, J. Healy, and J. Melville. UMAP: uniform manifold approximation and projection for dimension reduction. Preprint at `https://arxiv.org/abs/1802.03426`, 2018.

C. Meek. *Graphical Models: Selecting Causal and Statistical Models*. PhD thesis, Carnegie Mellon University, 1997.

N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

N. Meinshausen, A. Hauser, J. M. Mooij, J. Peters, P. Versteeg, and P. Bühlmann. Methods for causal inference from gene perturbation experiments and validation. *Proceedings of the National Academy of Sciences, U.S.A.*, 113(27):7361–7368, 2016.

T. Mikalsen, N. Gerits, and U. Moens. Inhibitors of signal transduction protein kinases as targets for cancer therapy. *Biotechnology Annual Review*, 12:153–223, 2006.

L. A. Mirny. The fractal globule as a model of chromatin architecture in the cell. *Chromosome research*, 19(1):37–51, 2011.

B. Mishra, G. Meyer, and R. Sepulchre. Low-rank optimization for distance matrix completion. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 4455–4460, 2011.

A. Mitra, S. Venkatachalapathy, P. Ratna, Y. Wang, D. S. Jokhun, and G. V. Shivashankar. Cell geometry dictates TNFα-induced genome response. *Proceedings of the National Academy of Sciences, U.S.A.*, 114(20):E3882–E3891, 2017.

K. Monahan, A. Horta, and S. Lomvardas. LHX2-and LDB1-mediated trans interactions regulate olfactory receptor choice. *Nature*, 565(7740):448–453, 2019.

I. Müller, S. Boyle, R. H. Singer, W. A. Bickmore, and J. R. Chubb. Stable morphology, but dynamic internal reorganisation, of interphase human chromosomes in living cells. *PloS One*, 5(7):e11560, 2010.

T. Nagano, Y. Lubling, T. J. Stevens, S. Schoenfelder, E. Yaffe, W. Dean, E. D. Laue, A. Tanay, and P. Fraser. Single-cell hi-c reveals cell-to-cell variability in chromosome structure. *Nature*, 502(7469):59–64, 2013.

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning*, pages 689–696, 2011.

M. Niepel, M. Hafner, Q. Duan, Z. Wang, E. O. Paull, M. Chung, X. Lu, J. M. Stuart, T. R. Golub, A. Subramanian, A. Ma'ayan, and P. K. Sorger. Common and cell-type specific responses to anti-cancer drugs revealed by high throughput transcript profiling. *Nature Communications*, 8:1186, 2017.

G. Nir, I. Farabella, C. P. Estrada, C. G. Ebeling, B. J. Beliveau, H. M. Sasaki, S. H. Lee, S. C. Nguyen, R. B. McCole, S. Chattoraj, et al. Walking along chromosomes with super-resolution imaging, contact maps, and integrative modeling. *PLoS Genetics*, 14(12):e1007872, 2018.

H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 27(1):29–34, 1999.

P. Olivares-Chauvet, Z. Mukamel, A. Lifshitz, O. Schwartzman, N. O. Elkayam, Y. Lubling, G. Deikus, R. P. Sebra, and A. Tanay. Capturing pairwise and multi-way chromosomal conformations using chromosomal walks. *Nature*, 540:296–300, 2016.

G. Onder, G. Rezza, and S. Brusaferro. Case-fatality rate and characteristics of patients dying in relation to COVID-19 in Italy. *JAMA*, 323(18):1775–1776, 2020.

V. Pancaldi, E. Carrillo-de Santa-Pau, B. M. Javierre, D. Juan, P. Fraser, M. Spivakov, A. Valencia, and D. Rico. Integrating epigenomic data and 3D genomic structure with a new measure of chromatin assortativity. *Genome Biology*, 17(1): 1–19, 2016.

A. Papantonis and P. R. Cook. Transcription factories: genome organization and gene regulation. *Chemical Reviews*, 113(11):8683–8705, 2013.

A. Papantonis, T. Kohro, S. Baboo, J. D. Larkin, B. Deng, P. Short, S. Tsutsumi, S. Taylor, Y. Kanki, M. Kobayashi, G. Li, H.-M. Poh, X. Ruan, H. Aburatani, Y. Ruan, T. Kodama, Y. Wada, and P. R. Cook. TNFα signals through specialized factories where responsive coding and miRNA genes are transcribed. *The EMBO Journal*, 31(23):4404–4414, 2012.

J. Pearl. *Causality*. Cambridge University Press, Cambridge, UK, 2 edition, 2009.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2012.

A. Pombo and N. Dillon. Three-dimensional genome architecture: players and mechanisms. *Nature Reviews Molecular Cell Biology*, 16(4):245–257, 2015.

M. Poppe, S. Wittig, L. Jurida, M. Bartkuhn, J. Wilhelm, H. Müller, et al. The NF-κB-dependent and-independent transcriptome and chromatin landscapes of human coronavirus 229E-infected cells. *PLoS Pathogens*, 13(3):e1006286, 2017.

S. Pushpakom, F. Iorio, P. A. Eyers, K. J. Escott, S. Hopper, A. Wells, A. Doig, J. Guilliams, T. Latimer, C. McNamee, A. Norris, P. Sanseau, D. Cavalla, and M. Pirmohamed. Drug repurposing: progress, challenges and recommendations. *Nature Reviews Drug Discovery*, 18(1):41–58, 2019.

Y. Qi and B. Zhang. Predicting three-dimensional genome organization with chromatin states. *PLoS computational biology*, 15(6):e1007024, 2019.

S. A. Quinodoz, N. Ollikainen, B. Tabak, A. Palla, J. M. Schmidt, E. Detmar, M. M. Lai, A. A. Shishkin, P. Bhat, Y. Takei, et al. Higher-order inter-chromosomal hubs shape 3D genome organization in the nucleus. *Cell*, 174(3):744–757, 2018.

A. Radhakrishnan, M. Belkin, and C. Uhler. Overparameterized neural networks can implement associative memory. Preprint at `https://arxiv.org/abs/1909.12362`, 2019.

V. Ramani, X. Deng, R. Qiu, K. L. Gunderson, F. J. Steemers, C. M. Disteche, W. S. Noble, Z. Duan, and J. Shendure. Massively multiplex single-cell Hi-C. *Nature Methods*, 14(3):263—-266, 2017.

S. S. P. Rao, M. H. Huntley, N. C. Durand, E. K. Stamenova, I. D. Bochkov, J. T. Robinson, A. L. Sanborn, I. Machol, A. D. Omer, E. S. Lander, and E. L. Aiden. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159(7):1665–1680, 2014.

S. Razick, G. Magklaras, and I. M. Donaldson. iRefIndex: a consolidated protein interaction database with provenance. *BMC Bioinformatics*, 9(1):405, 2008.

J. Reimand, M. Kull, H. Peterson, J. Hansen, and J. Vilo. g:Profiler-a web-based toolset for functional profiling of gene lists from large-scale experiments. *Nucleic Acids Research*, 35(suppl_2):W193–W200, 2007.

J. Reimand, T. Arak, P. Adler, L. Kolberg, S. Reisberg, H. Peterson, and J. Vilo. g:Profiler-a web server for functional interpretation of gene lists (2016 update). *Nucleic Acids Research*, 44(W1):W83–W89, 2016.

D. N. Reshef et al. Detecting novel associations in large data sets. *Science*, 334(6062): 1518–1524, 2011.

P. A. Reyfman, J. M. Walter, N. Joshi, K. R. Anekalla, A. C. McQuattie-Pimentel, S. Chiu, R. Fernandez, M. Akbarpour, C.-I. Chen, Z. Ren, et al. Single-cell transcriptomic analysis of human lung provides insights into the pathobiology of pulmonary fibrosis. *American Journal of Respiratory and Critical Care Medicine*, 199 (12):1517–1536, 2019.

T. S. Richardson. Overparameterized neural networks can implement associative memory. Preprint at `https://arxiv.org/abs/1302.3599`, 2019.

L. Rieber and S. Mahony. miniMDS: 3D structural inference from high-resolution Hi-C data. *Bioinformatics*, 33(14):i261–i266, 2017.

A. Rotem, O. Ram, N. Shoresh, R. A. Sperling, A. Goren, D. A. Weitz, and B. E. Bernstein. Single-cell ChIP-seq reveals cell subpopulations defined by chromatin state. *Nature Biotechnology*, 33(11):1165–1172, 2015.

M. Rousseau, J. Fraser, M. A. Ferraiuolo, J. Dostie, and M. Blanchette. Three-dimensional modeling of chromatin structure from interaction frequency data using Markov chain Monte Carlo sampling. *BMC Bioinformatics*, 12(1):414, 2011.

A. L. Roy and R. S. Conroy. Toward mapping the human body at a cellular resolution. *Molecular Biology of the Cell*, 29(15):1779–1785, 2018.

M. Sauler, I. S. Bazan, and P. J. Lee. Cell death in the lung: the apoptosis–necroptosis axis. *Annual Review of Physiology*, 81:375–402, 2019.

A. D. Schmitt, M. Hu, and B. Ren. Genome-wide mapping and analysis of chromosome architecture. *Nature Reviews Molecular Cell Biology*, 17(12):743–755, 2016.

S. Schoenfelder, T. Sexton, L. Chakalova, N. F. Cope, A. Horton, S. Andrews, S. Kurukuti, J. a. Mitchell, D. Umlauf, D. S. Dimitrova, C. H. Eskiw, Y. Luo, C.-L. Wei, Y. Ruan, J. J. Bieker, and P. Fraser. Preferential associations between co-regulated genes reveal a transcriptional interactome in erythroid cells. *Nature Genetics*, 42 (1):53–61, 2010.

S. Schoenfelder, R. Sugar, A. Dimond, B.-M. Javierre, H. Armstrong, B. Mifsud, E. Dimitrova, L. Matheson, F. Tavares-Cadete, M. Furlan-Magaril, et al. Polycomb repressive complex PRC1 spatially constrains the mouse embryonic stem cell genome. *Nature Genetics*, 47(10):1179–1186, 2015.

M. R. Segal and H. L. Bengtsson. Reconstruction of 3D genome architecture via a two-stage algorithm. *BMC Bioinformatics*, 16(1):373, 2015.

F. Serra, M. Di Stefano, Y. G. Spill, Y. Cuartero, M. Goodstadt, D. Baù, and M. A. Marti-Renom. Restraint-based three-dimensional modeling of genomes and genomic domains. *FEBS Letters*, 589(20):2987–2995, 2015.

A. A. Shabalin, V. J. Weigman, C. M. Perou, and A. B. Nobel. Finding large average submatrices in high dimensional data. *Annals of Applied Statistics*, 3(3):985–1012, 2009.

S. Shachar, T. C. Voss, G. Pegoraro, N. Sciascia, and T. Misteli. Identification of Gene Positioning Factors Using High-Throughput Imaging Mapping. *Cell*, 162(4): 911–923, 2015. ISSN 10974172. doi: 10.1016/j.cell.2015.07.035. URL `http://dx.doi.org/10.1016/j.cell.2015.07.035`.

A. Shojaie. Differential network analysis: A statistical perspective. *Wiley Interdisciplinary Reviews: Computational Statistics*, page e1508, 04 2020. doi: 10.1002/wics.1508.

D. Sicard, A. J. Haak, K. M. Choi, A. R. Craig, L. E. Fredenburgh, and D. J. Tschumperlin. Aging and anatomical variations in lung tissue stiffness. *American Journal of Physiology-Lung Cellular and Molecular Physiology*, 314(6):L946–L955, 2018.

M. Simonis, P. Klous, E. Splinter, Y. Moshkin, R. Willemsen, E. de Wit, B. van Steensel, and W. de Laat. Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation capture–on-chip (4C). *Nature Genetics*, 38(11):1348–1354, 2006.

S. B. Smith, W. Dampier, A. Tozeren, J. R. Brown, and M. Magid-Slav. Identification of common biological pathways and drug targets across multiple respiratory viruses based on human host gene expression analysis. *PLoS One*, 7(3):e331741, 2012.

J. E. Smith-Garvin, G. A. Koretzky, and M. S. Jordan. T cell activation. *Annual Review of Immunology*, 27:591–619, 2009.

L. Solus, Y. Wang, and C. Uhler. Consistency guarantees for greedy permutation-based causal inference algorithms. *arXiv:1702.03530*, 2017.

P. Spagnolo, E. Balestro, S. Aliberti, E. Cocconcelli, D. Biondini, G. D. Casa, N. Sverzellati, and T. M. Maher. Pulmonary fibrosis secondary to COVID-19: a call to arms? *The Lancet Respiratory Medicine*, 2020.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* MIT press, 2000.

R. Stadhouders, E. Vidal, F. Serra, B. Di Stefano, F. Le Dily, J. Quilez, A. Gomez, S. Collombet, C. Berenguer, Y. Cuartero, et al. Transcription factors orchestrate dynamic interplay between genome topology and gene regulation during cell reprogramming. *Nature Genetics*, 50(2):238–249, 2018.

R. Stadhouders, G. J. Filion, and T. Graf. Transcription factors and 3D genome conformation in cell-fate decisions. *Nature*, 569(7756):345–354, 2019.

P. L. Ståhl, F. Salmén, S. Vickovic, A. Lundmark, J. F. Navarro, J. Magnusson, S. Giacomello, M. Asp, J. O. Westholm, M. Huss, et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294): 78–82, 2016.

J. S. Stanley, S. Gigante, G. Wolf, and S. Krishnaswamy. Harmonic alignment. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 316–324. SIAM, 2020.

S. Stejskal, K. Stepka, L. Tesarova, K. Stejskal, P. Simara, Z. Zdrahal, and I. Koutna. Cell cycle-dependent changes in H3K56ac in human cells. *Cell Cycle*, 14(24):3851–3863, 2015.

T. J. Stevens, D. Lando, S. Basu, L. P. Atkinson, Y. Cao, S. F. Lee, M. Leeb, K. J. Wohlfahrt, W. Boucher, A. O'Shaughnessy-Kirwan, et al. 3D structures of individual mammalian genomes studied by single-cell Hi-C. *Nature*, 544(7648):59, 2017.

A. R. Strom, A. V. Emelyanov, M. Mir, D. V. Fyodorov, X. Darzacq, and G. H. Karpen. Phase separation drives heterochromatin domain formation. *Nature*, 547 (7662):241–245, 2017.

T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. Mauck III, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902, 2019.

A. Subramanian, R. Narayan, S. M. Corsello, et al. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6):1437—1452.e1, 2017.

1000 Genomes Project Consortium et al. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65, 2012.

1000 Genomes Project Consortium et al. A global reference for human genetic variation. *Nature*, 526(7571):68–74, 2015.

H. Tjong, W. Li, R. Kalhor, C. Dai, S. Hao, K. Gong, Y. Zhou, H. Li, X. J. Zhou, M. A. Le Gros, C. A. Larabell, L. Chen, and F. Alber. Population-based 3D genome structure analysis reveals driving forces in spatial genome organization. *Proceedings of the National Academy of Sciences of the United States of America*, 113(12):E1663–E1672, 2016.

R. W. Tothill, A. V. Tinker, J. George, R. Brown, S. B. Fox, S. Lade, D. S. Johnson, M. K. Trivett, D. Etemadmoghadam, B. Locandro, et al. Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome. *Clinical Cancer Research*, 14(16):5198–5208, 2008.

T. N. Trong, R. Kramer, J. Mehtonen, G. González, V. Hautamäki, and M. Heinäniemi. Semisupervised generative autoencoder for single-cell data. *Journal of Computational Biology*, 27(8):1190–1203, 2020.

N. Tuncbag, S. McCallum, S. Huang, and E. Fraenkel. Steinernet: A web server for integrating 'omic' data to discover hidden components of response pathways. *Nucleic Acids Research*, 2012.

C. Uhler and G. Shivashankar. Mechanogenomic coupling of lung tissue stiffness, emt and coronavirus pathogenicity. *Current Opinion in Solid State and Materials Science*, 25(1):100874, 2020a.

C. Uhler and G. V. Shivashankar. Chromosome intermingling: Mechanical hotspots for genome regulation. *Trends in Cell Biology*, 27(11):810–819, 2017a.

C. Uhler and G. V. Shivashankar. Regulation of genome organization and gene expression by nuclear mechanotransduction. *Nature Reviews Molecular Cell Biology*, 18(12):717—-727, 2017b.

C. Uhler and G. V. Shivashankar. Mechano-genomic regulation of coronaviruses and its interplay with ageing. *Nature Reviews Molecular Cell Biology*, 21:247–248, 2020b.

O. Ursu, J. Holmes, J. Knockel, C. G. Bologa, J. J. Yang, S. L. Mathias, S. J. Nelson, and T. I. Oprea. DrugCentral: online drug compendium. *Nucleic Acids Research*, page gkw993, 2016.

O. Ursu, J. Holmes, C. G. Bologa, J. J. Yang, S. L. Mathias, V. Stathias, D.-T. Nguyen, S. Schürer, and T. Oprea. DrugCentral 2018: an update. *Nucleic acids research*, 47(D1):D963–D970, 2019.

B. Van Steensel and A. S. Belmont. Lamina-associated domains: links with chromosome architecture, heterochromatin, and gene repression. *Cell*, 169(5):780–791, 2017.

J. M. Vaquerizas, S. K. Kummerfeld, S. A. Teichmann, and N. M. Luscombe. A census of human transcription factors: function, expression and evolution. *Nature Reviews Genetics*, 10(4):252–263, 2009.

N. Varoquaux, F. Ay, W. S. Noble, and J.-P. Vert. A statistical approach for inferring the 3D structure of the genome. *Bioinformatics*, 30(12):i26–i33, 2014.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pages 255–270, 1990.

H. Wang, X. Xu, C. M. Nguyen, Y. Liu, Y. Gao, X. Lin, T. Daley, N. H. Kipniss, M. La Russa, and L. S. Qi. CRISPR-mediated programmable 3D genome positioning and nuclear organization. *Cell*, 175(5):1405–1417, 2018a.

S. Wang, J. Xu, and J. Zeng. Inferential modeling of 3D chromatin structure. *Nucleic Acids Research*, 43(8):e54, 2015.

Y. Wang, L. Solus, K. D. Yang, and C. Uhler. Permutation-based causal inference algorithms with interventions. In *Advances in Neural Information Processing Systems*, pages 5822–5831, 2017.

Y. Wang, C. Squires, A. Belyaeva, and C. Uhler. Direct estimation of differences in causal graphs. In *Advances in Neural Information Processing Systems*, pages 3770–3781, 2018b.

Y. R. Wang and H. Huang. Review on statistical methods for gene network reconstruction using expression data. *Journal of Theoretical Biology*, 362:53–61, 2014.

K. Q. Weinberger, F. Sha, Q. Zhu, and L. K. Saul. Graph Laplacian regularization for large-scale semidefinite programming. In *Advances in Neural Information Processing Systems*, pages 1489–1496, 2007.

S. Whalen, R. M. Truty, and K. S. Pollard. Enhancer-promoter interactions are encoded by complex genomic signatures on looping chromatin. *Nature Genetics*, 48(5):488–496, 2016.

T. Willinger, T. Freeman, M. Herbert, H. Hasegawa, A. J. McMichael, and M. F. Callan. Human naive CD8 T cells down-regulate expression of the WNT pathway transcription factors lymphoid enhancer binding factor 1 and transcription factor 7 (T cell factor-1) following antigen encounter in vitro and in vivo. *The Journal of Immunology*, 176(3):1439–1446, 2006.

J. T. Wu, K. Leung, M. Bushman, N. Kishore, R. Niehus, P. M. de Salazar, B. J. Cowling, M. Lipsitch, and G. M. Leung. Estimating clinical severity of COVID-19 from the transmission dynamics in Wuhan, China. *Nature Medicine*, 26:506–510, 2020.

C. W. Yang, Y. Z. Lee, H. Y. Hsu, C. Shih, Y. S. Chao, H. Y. Chang, and S. J. Lee. Targeting coronaviral replication and cellular JAK2 mediated dominant NF-κB activation for comprehensive and ultimate inhibition of coronaviral activity. *Scientific Reports*, 7(1):4105, 2017a.

K. D. Yang and C. Uhler. Multi-domain translation by learning uncoupled autoencoders. In *ICML workshop in Computational Biology*, 2019.

K. D. Yang, A. Katcoff, and C. Uhler. Characterizing and learning equivalence classes of causal dags under interventions. *Proceedings of Machine Learning Research*, 80: 5537–5546, 2017b.

K. D. Yang, A. Belyaeva, S. Venkatachalapathy, A. Radhakrishnan, A. Katcoff, G. V. Shivashankar, and C. Uhler. Multi-domain translation between single-cell imaging and sequencing data using autoencoders. *uhlerlab/cross-modal-autoencoders https://doi.org/10.5281/zenodo.4266733*, 2020a. URL `https://doi.org/10.5281/zenodo.4266733`.

K. D. Yang, A. Belyaeva, S. Venkatachalapathy, A. Radhakrishnan, A. Katcoff, G. V. Shivashankar, and C. Uhler. Multi-domain translation between single-cell imaging and sequencing data using autoencoders. *SaradhaVenkatachalapathy/Radial_ chromatin_ packing_ immune_ cells: Code for paper submisssion https://doi.org/10.5281/zenodo.4267003*, 2020b. URL `https://doi.org/10.5281/zenodo.4267003`.

K. D. Yang, A. Belyaeva, S. Venkatchalapathy, K. Damodaran, A. Radhakrishnan, A. Katcoff, G. V. Shivashankar, and C. Uhler. Multi-domain translation between single-cell imaging and sequencing data using autoencoders. *Nature Communications (in press) https://doi.org/10.1101/2019.12.13.875922*, 2020c.

K. D. Yang, K. Damodaran, S. Venkatchalapathy, A. C. Soylemezoglu, G. V. Shivashankar, and C. Uhler. Autoencoder and optimal transport to infer single-cell trajectories of biological processes. *PLoS Computational Biology*, 16:e1007828, 2020d.

R. Zenobi. Single-cell metabolomics: analytical and biological perspectives. *Science*, 342(6163):1243259, 2013.

L. Zhang, G. Wahba, and M. Yuan. Distance shrinkage and Euclidean embedding via regularized kernel estimation. *Journal of the Royal Statistical Society: Series B*, 78(4):849–867, 2016.

Z. Zhang, G. Li, K.-C. Toh, and W.-K. Sung. 3D Chromosome Modeling with Semi-Definite Programming and Hi-C Data. *Journal of Computational Biology*, 20(11), 2013.

S. D. Zhao, T. T. Cai, and H. Li. Direct estimation of differential networks. *Biometrika*, 101(2):253–268, 2014.

G. X. Y. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, and J. e. a. Zhu. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8:14049, 2017.

H. Zheng and W. Xie. The role of 3D genome organization in development and cell differentiation. *Nature Reviews Molecular Cell Biology*, 20:535—-550, 2019.

Y. Zhou, Y. Hou, J. Shen, Y. Huang, W. Martin, and F. Cheng. Network-based drug repurposing for novel coronavirus 2019-nCoV/SARS-CoV-2. *Cell Discovery*, 6(1): 1–18, 2020.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

Y. Zhu, Z. Chen, K. Zhang, M. Wang, D. Medovoy, J. W. Whitaker, B. Ding, N. Li, L. Zheng, and W. Wang. Constructing 3D interaction maps from 1D epigenomes. *Nature Communications*, 7:10812, 2016.