

SCALABLE NETWORK INFERENCE THROUGH NOISE
FOR GENES AND BEYOND

A dissertation

submitted by

Hao Zhu

In partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Computer Science

TUFTS UNIVERSITY

Feb 2025

ADVISOR: Prof. Donna K. Slonim

SCALABLE NETWORK INFERENCE THROUGH NOISE FOR GENES AND BEYOND

Hao Zhu

ADVISOR: Prof. Donna K. Slonim

A common theme across multiple disciplines of science is to understand the underlying dependencies between variables from observational data. Modeling such dependencies as networks helps us build a systematic understanding of the entire complex system. A particular interesting use case is the application of network inference on biological networks, especially with genes and proteins. Understanding these complex interactions under different contexts could make a huge impact on helping us understand, even at an individual level, how our body functions and how we respond to diseases. Ultimately, this knowledge could help us develop new therapies or personalize treatments for these diseases.

In this thesis, we will discuss how we can utilize noise in the inference for Gene Regulatory Networks (GRNs) and beyond. Our contributes are as follows:

- In Chapter 3, we introduce *dropout augmentation* (DA) and *DAZZLE* (Zhu and Slonim, 2023) for GRN inference on single-cell data with inflated zeros. Single-cell data often suffers from “dropout” or zero-inflation, when some transcripts’ expression values are erroneously not captured. DA regularizes model parameters by augmenting data with synthetic dropout events. It helps the models stay robust and offers a novel alternative to data imputation.
- In Chapter 4, following the noise injection and denoising idea in DA, we propose *RegDiffusion*, a new class of Denoising Diffusion Probabilistic Models

(DDPMs) for network inference (Zhu and Slonim, 2024). RegDiffusion adds Gaussian noise to the input data and then denoises the perturbed data using a neural network with a parameterized adjacency matrix. This process eliminates a major computation bottleneck in autoencoder-based SEM models, enabling very fast and accurate GRN inference at reduced complexity.

- In Chapter 5, we further propose a set of *Denoising Diffusion Causal Discovery* (DDCD) models (Zhu et al., 2024), aiming to answer the following two questions: (1) Can denoising diffusion models be applied to general causal discovery? (2) How can we make inferred edges more “causal”? In this work, we generalize and expand the denoising objective used in RegDiffusion, and we add a k-hop DAG constraint to encourage causality while allowing large feedback loops. DDCD models demonstrate great performances and runtime on both synthetic and real data, even with limited numbers of observations. However, for gene networks, our results showed that adding the acyclicity constraint, no matter if it’s full DAG or the k-hop version, would downgrade the quality of the predicted networks.

Overall, unlike traditional methods that treat noise solely as a problem to eliminate, my PhD work explores the possibility of either utilizing noise as a model regularization method or simply inferring networks using a denoising diffusion model. The improvement on the scalability and robustness of these algorithms demonstrates their potential to address the challenges of high-dimensional, noisy, and sparse data, ultimately enabling more accurate and efficient network inference across diverse applications.

To my wife Siyi Gu, who always supported me to pursue a meaningful career.

To my sons, Kaifeng and Yilong, who gave me endless joy! I wish you can always
stay as happy as you are in your future life!

To my parents, Bei and Wu, who gave me the core belief to stay positive and true.

To everyone who have helped me, rescued me, and lead me to this point of life.

Acknowledgments

I have to confess when I started my PhD in 2019, I had no idea what it meant to be a scientist, and I'm very glad I learned what it means now.

In 2018, I asked someone who I met at a conference for advice, and he told me that getting this degree would open up some new doors in my life. Sure, from my point of view at that time, it simply meant that it would gave me opportunities to work on some fancy AI models, which was very attractive.

I think my mind gradually changed after I started to work with my advisor Prof. Donna Slonim. I started to enjoy thinking (strange way to say it but, yes), and working on problems that do not have a clear solution. I started to move away from my engineering mindset and be more like a "scientist" (yet, some part of me is still a proud engineer, for the good part). At this point of my life, I can only say I'm really grateful to be able to look at the world in this way and I truly appreciate Donna for her guidance along the journey.

I also want to acknowledge Prof. Lenore Cowen, who and Donna sent out their helping hand when I needed the most. Furthermore, I would like to thank Prof. Liping Liu, who helped me setup a foundation in deep learning, math, and experiment design. I also truly appreciate Prof. Xiaozhe Hu and Prof. Kimberly Glass for being the first readers of this thesis and their valuable support. I also want to appreciate Prof. Thomas Trivison, my previous supervisor at Marcus Institute, for his kindness and his efforts on creating a supportive environment.

Finally, for my family and friends, especially my friends at the Tufts BCB group, I truly appreciate your help and support throughout the years!

HAO ZHU

TUFTS UNIVERSITY

Feb 2025

Contents

Abstract	ii
Acknowledgments	v
List of Tables	x
List of Figures	xiii
Chapter 1 Introduction	1
1.1 A Story about <i>TP53</i>	1
1.2 Gene Regulatory Networks and Where to Find Them	3
1.2.1 Characteristics of Gene Regulatory Networks	3
1.2.2 How to Identify GRNs? - Opportunities and Challenges . . .	4
1.3 Beyond Genes	6
1.3.1 Network Inference Beyond Genes	6
1.3.2 Correlation, Causality, and DAGs	6
1.3.3 Confounding factors	8
1.4 Research Questions and Scope	8
Chapter 2 Background	9
2.1 Existing Methods to Construct GRNs	9
2.1.1 Existing Regulatory Databases	9
2.2 Transcriptomics based GRN Inference	12
2.3 Review of related deep learning methods	14

2.3.1	Variational Autoencoder	14
2.3.2	Denoising Diffusion Probabilistic Models	15
2.4	Review of network inference and causal learning	18
2.4.1	Problem Statement	18
2.4.2	Structural Equation Models (SEMs)	19
2.4.3	DeepSEM	20
2.4.4	Continuous DAG constraint	21

Chapter 3 Improved GRN inference from single cell data with dropout

augmentation		22
3.1	Introduction	22
3.2	Methods	24
3.2.1	Dropout Augmentation	24
3.2.2	GRN Inference with DAZZLE	25
3.2.3	Experiment Design	28
3.3	Experiments	31
3.3.1	DAZZLE improves GRN inference on BEELINE benchmarks	31
3.3.2	Dropout Augmentation contributes to model robustness . . .	34
3.3.3	Sparsity control strategy improves DAZZLE’s stability	36
3.3.4	Case study: Using DAZZLE to predict temporal changes in GRNs in mouse microglia	37
3.4	Discussion	41

Chapter 4 Diffusion Probabilistic Model-Based GRN Inference

4.1	Introduction	43
4.2	Methods	44
4.2.1	RegDiffusion	44
4.2.2	Experiment Design	51
4.3	Experiments	53
4.3.1	RegDiffusion is Highly Accurate on BEELINE Benchmarks .	53
4.3.2	RegDiffusion Runs Faster than Competitors	54

4.3.3	An Example of Interpreting the Inferred Networks	56
4.3.4	Inferred regulatory neighborhoods are cell-type specific	58
4.3.5	Characteristics of RegDiffusion	59
4.3.6	Regulatory Networks in Microglia	63
4.4	Discussion and Conclusion	65
Chapter 5	Denoising Diffusion Causal Discovery	68
5.1	Introduction	68
5.2	Methods	70
5.2.1	Denoising Diffusion Models for Linear SEMs	70
5.2.2	Denoising Diffusion Models for Nonlinear SEMs	73
5.2.3	Scale handling with smoothed features and adjacency matrix	74
5.2.4	k-hop DAG Constraint with gradient clipping	76
5.2.5	Fixed-size Bootstrap Sampling	77
5.2.6	Optimization	78
5.2.7	Experiment Design	78
5.3	Experiments	80
5.3.1	Denoising objective leads to smoother gradients	80
5.3.2	Linear Synthetic Data	81
5.3.3	Nonlinear Synthetic Data	82
5.3.4	Real World Observational Data: Myocardial Infarction	83
5.3.5	Real World Observational Data: Gene Regulatory Networks .	85
5.4	Discussion and Conclusion	88
Chapter 6	Discussion and Future Works	90

List of Tables

3.1	DAZZLE shows improved GRN inference capacity on BEELINE benchmarks	32
3.2	Performance on additional benchmarks	33
4.1	RegDiffusion outperforms most GRN inference methods on BEELINE benchmark datasets (Metric: AUPRC Ratio; Ground truth: STRING; Number of target genes: 1000). Numbers reported are mean and std. of AUPRC Ratio compared with random guess over 10 runs. Higher ratios indicate better performance. Here, bold Red and italicized <i>orange</i> text indicate the best and <i>2nd best</i> algorithms. In this experiment, we ran each algorithm on all available data with the default settings.	53
4.2	BEELINE Benchmarking Results (STRING network , 1,000 target genes) evaluated on Early Precision Ratio (EPR). Results are from the same experiment shown in Table 1 in the main text.	53
4.3	BEELINE Benchmarking Results (Non-celltype-specific ChIP-Seq network , 1,000 target genes) evaluated on AUPRC Ratio (AUPRR). Results are from a similar experiment to that shown in Table 1 in the main text, but with a different ground truth network.	54

4.4	BEELINE Benchmarking Results (Non-celltype-specific ChIP-Seq network, 1,000 target genes) evaluated on Early Precision Ratio (EPR). Results are from a similar experiment to that shown in Table 1 in the main text, with a different ground truth network.	54
4.5	BEELINE Benchmarking Results (Celltype-specific ChIP-Seq network, 1,000 target genes) evaluated on AUPRC Ratio (AUPRR). DeepSEM*, DeepSEM, DAZZLE are all run using a different set of hyperparameters recommended by the DeepSEM paper only for use on this benchmark (L1 Loss = 1% of default; KL divergence = 1% of default). RegDiffusion also uses a different set of hyperparameters (L1 loss = 1% of default; L2 Loss = 1% of default; Adjacency matrix initialization = 10 * regulation norm). Because of the low performance ratios and unique hyperparameters here, plus a lack of understanding for why performance with the original hyperparameters is much worse, we do not recommend evaluating on this set of ground truth.	55
4.6	BBEELINE Benchmarking Results (Celltype-specific ChIP-Seq network, 1,000 target genes) evaluated on Early Precision Ratio (EPR). DeepSEM*, DeepSEM, DAZZLE are all run using a different set of hyperparameters recommended by the DeepSEM paper only for use on this benchmark (L1 Loss = 1% of default; KL divergence = 1% of default). RegDiffusion also uses a different set of hyperparameters (L1 loss = 1% of default; L2 Loss = 1% of default; Adjacency matrix initialization = 10 * regulation norm). Because of the low performance ratios and unique hyperparameters here, plus a lack of understanding for why performance with the original hyperparameters is much worse, we do not recommend evaluating on this set of ground truth.	55

4.7	Algorithm execution time (reported in seconds unless noted). Time cost on BEELINE data sets are averages over 10 runs. All code was executed on HPC computing nodes with 4 cores and 12GB memory. DeepSEM, DeepSEM*, DAZZLE, and RegDiffusion are executed with an additional NVIDIA A100 card.	56
4.8	RegDiffusion outperforms DeepSEM in two large microglia experiments on all metrics. Both of the ground truth networks (STRING and Non-cell-type-specific ChIP-Seq are the full networks from BEE-LINE.	63
5.1	Systematic Evaluation Results on the Yeast Network using similar metrics as used in Chapter 4.	87

List of Figures

1.1	The p53 regulatory pathways. Figure is adapted from Levine (2020) Figure 2 and 3.	2
2.1	Illustration of the forward and backward processes of a diffusion model for computer vision. Figure is adapted from Kreis et al. (2022). 16	
3.1	One of the major differences between DAZZLE and DeepSEM is the use of Dropout Augmentation. Dropout augmentation regularizes model training by simulating small amounts of random dropout at each training iteration such that the model is protected against the negative impact of dropout noise. Rounded boxes are trainable model parameters.	25
3.2	Quality of inferred GRN from DeepSEM-1x: AUPRC Ratio as a func- tion of the number of training iterations. Quality may quickly down- grade after the performance peak. Dashed line is the recommended stopping point from DeepSEM. Note that for the celltype-specific data sets, performance is particularly volatile.	35

3.3	Appropriate amount of augmented dropout helps maintain model robustness and may contribute to better performance. Color reflects the probability of dropout augmentation. The two thick lines represent two important conditions, 0% - no augmented dropout, and 10% - the default dropout augmentation level we recommend. Dashed lines shows the default number of training iterations used in DeepSEM and DAZZLE.	36
3.4	Comparison of 100 runs of DeepSEM-1x and DAZZLE-1x on hESC evaluated using the STRING network. In DeepSEM-1x, the prioritization of the L1 sparsity control at early stage is the main cause of unstable GRN inference performances. DAZZLE solved this issue by delaying the introduction of L1 sparse loss by 5 iterations. a. Histogram of AUPRC ratios. b. Sparse loss over time for DeepSEM-1X, colored by AUPRR at convergence. c. Sparse loss over time for DAZZLE-1x, colored by AUPRR at convergence.	38
3.5	Distribution of 100 runs of DAZZLE-1x and DeepSEM-1x on BEE-LINE evaluated using the STRING network. Results from DAZZLE-1x tend to be more stable than results from DeepSEM-1x.	38
3.6	a. Top 10 regulated genes at each life stage in mouse microglia. b. Predicted local networks around <i>Tmem119</i> and <i>Apoe</i> in mouse microglia. Here edge weights are min-max scaled at each time point. Top genes are selected according to the maximum weight at all time points.	39

4.1	The Diffusion Process is an iterative process that consists of both the forward diffusion pass and the reverse pass. The top panel (based on (Ho et al., 2020)) shows the forward diffusion process transforming all data into Gaussian noise in T steps following a diffusion schedule, while the reverse process aims to recover the original distribution from the near-Normal distribution. We are modeling the reverse process with the adjacency matrix being part of the model. The bottom panel shows four representative genes and how their distributions change over time. Each gene begins with a unique distribution over all cells, but by the final time step the distribution is nearly Normal.	45
4.2	Graphic illustration of the RegDiffusion model. Green areas represent trainable parameters within the model. RegDiffusion starts with embedding the model inputs (gene expression x_t , diffusion time step t , and cell type ct) into corresponding feature matrices (h_x^0 , h_t^0 , and h_{ct}^0). These feature vectors are then integrated through 3 layers of MLP training blocks. In each training block, the integrated gene features h_x^{l-1} are mixed with the original embeddings of the time step and the cell type feature. The final integrated feature h_x^L then multiplies with the $(\mathbf{I} - \mathbf{A})$ matrix and the outcome is used to predict the added noise z_t	47
4.3	Local regulatory neighborhood analysis shows that the inferred regulatory networks generated by RegDiffusion are biologically interpretable. All the networks visualized here are 2-hop neighborhood around target genes at indicated top-k levels.	57

4.4	Properties of RegDiffusion. a. By measuring the amount of change in the adjacency matrix, we show RegDiffusion converges after a fixed number of steps in all experiments; b. Loss curves on train/test splits (4:1) show that the model was not overfitted. Also, it's expected to see the training loss oscillating even after the model converges; c. Model performance is rather stable regardless of the hidden dimensions in the neural networks; d. Downsampling shows that RegDiffusion performs better than bulk RNAseq methods even when the sample size is limited.	60
4.5	Comparison of Inference Performance (a. AUPRC Ratio; b. Early Precision Ratio) with Limited or Abundant data. Ground truth network is STRING. Line and bands represent mean and standard deviations of 10 runs. Each run of DeepSEM ensembles 10 single runs.	61
4.6	Inferred Local Neighborhood around Hist1h1d when inferred with two different batches of 90% sampled cells. These two neighborhoods are very similar with only some small variances (84-86% overlapping nodes and 65-71% overlapping edges).	62
4.7	Inferred 2-Hop Regulatory Networks (k=20) around <i>ApoE</i> from (a. Hammond; b. Cerebellum Atlas) microglia data using DeepSEM. Overall, these two networks are similar to the ones inferred by RegDiffusion but in both cases, there are more orphan links that are probably false predictions. In addition, in the Hammond result, the MHC I group and the antiviral group are missing.	66
5.1	Model architectures of proposed models in this chapter.	72

5.2	NOTEARS-Denoising runs faster than NOTEARS-Linear because it has smoother gradient. a. Average runtime over 10 runs with various number of samples; b. The L-BFGS-B optimizer converges with fewer number of steps on NOTEARS-Denoising; c. Gradient norm during the first iteration on the Full data (n=2,000). First 5 steps are removed for visualization purpose.	81
5.3	a. Example heatmap of weight estimates on a 100-node scale free graph with different numbers of observed samples using DDCD Linear. b. Benchmark Results on 2,000 observations on linear synthetic data over 10 runs, evaluated using Structural Hamming Distance (SHD). Lower scores indicate better performance. (SF: scale-free; ER: Erdős-Rényi) c. Algorithm execution time on 2,000 observations on CPUs over 10 repeated runs. Time axis is displayed in log scale.	82
5.4	a. Example Weight Estimates on an ER graph with 100 nodes using DDCD Nonlinear. DDCD Nonlinear not only infers the causal structure but also offers a way to approximate the underlying nonlinear transformation function. (Approximation indicated by blue dots in the second column. The ground truth transformation function is indicated by the red line). b. Benchmark Results with 2,000 observations over 10 runs evaluated using SHD.	83
5.5	Inferred Causal Network around Lethal Outcome in Myocardial Infarction	84
5.6	Comparison of the inferred gene regulatory neighborhood around <i>ADH1</i> using different models and settings. The network from STRING db is considered as a noisy and non-context-specific ground truth network.	86

6.1	Screenshot of the user interface of lightGraph. In the Zoom mode, you can zoom and pan the interface. In the Select mode, you can also drag and select nodes and assign labels to them. An oval will be drawn around the cluster to improve visual clarity.	92
6.2	An example of running Cell-Cell Interaction using transposed single cell data.	94

Chapter 1

Introduction

As human beings, we are so curious to understand the causes of everything happening around and within us. What could be the cause of the temperature drop in January? Why am I feeling excited today? Sometimes, these relationships could be simple but they could also be very complex. A good example of such complex causal relationships is the regulatory relationships among genes, one of the building blocks of our life. To understand such relationships, let's use the story of *TP53* as an example.

1.1 A Story about *TP53*

If we rank the “popularity” of all human genes according to their frequency in scientific publications, *TP53* will be at the top of the list (Dolgin, 2017). Its protein, p53, was discovered in 1979 when researchers were studying the mechanism of a cancer-causing virus called the simian virus 40 (SV40) (DeLeo et al., 1979; Kress et al., 1979; Lane and Crawford, 1979). In the mid 80s, early evidences found high level expression of p53 in many cancer cells so at that time, the scientific community widely accepted that, *TP53* was one of the “oncogenes” (Eliyahu et al., 1984; Jenkins et al., 1984; Vogelstein et al., 2010), the genes that cause cancer. It was not until the late 1980s that this perception was overturned. New evidence showed that the earlier studies had been based on the mutated forms of p53. The wild type p53

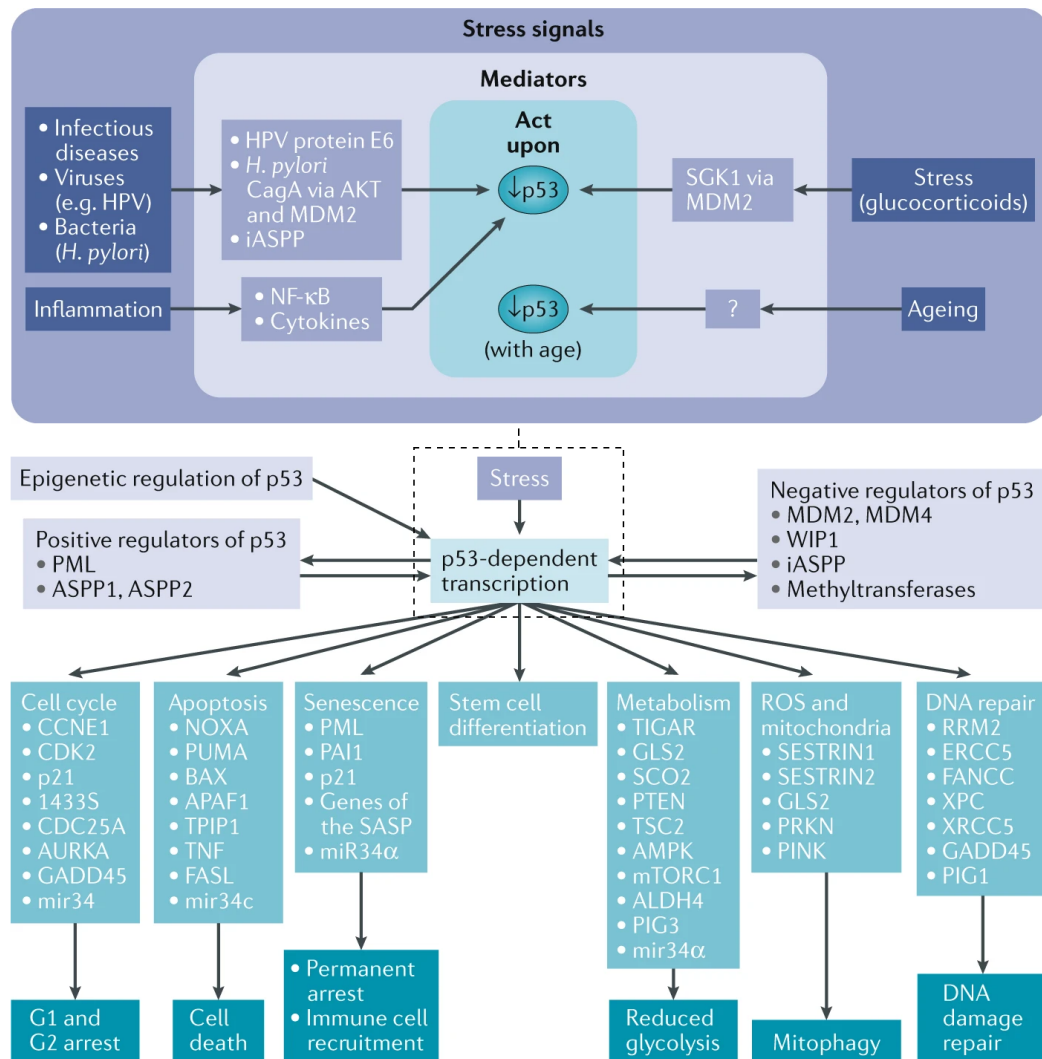


Figure 1.1: The p53 regulatory pathways. Figure is adapted from Levine (2020) Figure 2 and 3.

was found to be, in fact, a “tumor suppressor” (Finlay et al., 1989; Baker et al., 1989). When the wild type *TP53* is over expressed, it could slow down cell division by activating a wide range of molecular pathways, allowing time for DNA repair (Eliyahu et al., 1989). In the case of severe DNA damage, it could even activate another set of genes and ultimately trigger cell apoptosis (Yonish-Rouach et al., 1991, 1996), the programmed cell death that prevents the spreading of cancer.

Research in the past 40 years has presented us with a very complex and dynamic network of regulatory relationships around *TP53*, as shown in Figure 1.1 (Levine, 2020). As far as we know today, wild type *TP53* will respond to various

kinds of stress signals, including DNA damage, viral infection, inflammation, and aging. Depending on the actual context, *TP53* will act as a major transcription factor (TF), a protein that binds to specific DNA/RNA sequences to regulate the expression of target genes, and trigger different genes on several vastly different pathways, such as cell division (G1/G2) arrest, Cell death, immune responses, and DNA repair.

These regulatory relationships are not always one-way routes. Studies have shown that while *TP53* activates the transcription of MDM2, MDM2 binds with p53 and represses its transcriptional activity (Momand et al., 1992; Picksley and Lane, 1993; Manfredi, 2021). The negative feedback loop between MDM2 and p53 forms a central hub for p53 activation, playing a key role in therapeutic development related to p53. Due to the smooth surface of the p53 protein, it is challenging to target directly. The feedback loop between p53 and MDM2 makes MDM2 a popular alternative cancer treatment target today (Wang et al., 2023).

1.2 Gene Regulatory Networks and Where to Find Them

1.2.1 Characteristics of Gene Regulatory Networks

The history of our exploration of *TP53* highlights several important aspects of gene regulations.

- Gene regulatory relationships are often highly *complex*. Some “hub” genes, such as *TP53*, can interact with numerous genes that have different functions, forming complex interactions around them.
- Regulatory relationships are *directional*, meaning that transcription factors regulate their target genes. However, *feedback loops* are also common, adding further complexity to these networks.
- Regulatory relationships are *context dependent*. The interaction between a pair of genes, such as *TP53* and cell apoptosis genes, may only occur in certain cell types or under specific conditions.

- In therapeutic development, when a gene or protein is not easily targetable by drugs, neighboring genes, either upstream regulators or key players in downstream pathways, may serve as alternative therapeutic targets.

Given these characteristics, storing and modeling gene regulations as networks is a natural choice. Gene regulatory networks (GRNs) provide a systematic framework to understand how genes influence each other. Analyzing these networks helps reveal clusters of genes that are co-regulated for similar functions, identify critical hub genes or genes with higher centrality, and deepen our understanding on the underlying mechanism of various cellular developmental or diseases processes. Furthermore, similar to the MDM2-p53 pair we discussed above, they may help us identify potential points of regulation that could be amenable to therapeutic modulation (Emmert-Streib et al., 2014).

1.2.2 How to Identify GRNs? - Opportunities and Challenges

In practice, GRNs could be constructed by either querying from existing databases and scientific literatures, or performing computational network inference from experimentally obtained data. The former approach collects and curates knowledge from existing research and then builds them into databases (ENCODE Project Consortium, 2012; Zou et al., 2022; Han et al., 2018; Szklarczyk et al., 2019; Warde-Farley et al., 2010). We will have a more detailed discussions about these databases in Chapter 2. These resources are essential for developing a comprehensive understanding of GRNs, either by validating new inferences or by serving as a foundation for computational analysis. However, as of today, none of these databases could be considered as complete due to the imbalance in research interests among genes. At the same time, links collected in these databases are often not context specific, or at least to specific to the desired context. These two shortcomings raise concerns of both false negative (low coverage) and false positive (lack of specificity) errors.

These concerns can be largely addressed by the second approach, which involves computational network inference of GRNs from experimental data, such as

transcriptomic or other omics datasets. Transcriptomics, in particular, offers a snapshot of the RNA gene expression levels, providing insight into the activity of genes under the current cellular context. Even when multi-omics data is used, transcriptomics-based GRN inference often serves as a crucial starting point (Aibar et al., 2017) and its quality determines the quality of the final network. Previously, such inference was based on microarray or bulk RNA sequencing data sets (Mercatelli et al., 2020; Schaffter et al., 2011; Hecker et al., 2009). More recently, the increasing accessibility of single-cell RNA sequencing data has provided a higher-resolution view of expression in individual cells and specific cell states (Nguyen et al., 2021). In addition, Perturb-Seq is another form of transcriptomics data that is relevant to the field of GRN inference. Perturb-Seq uses CRISPR-based genetic perturbations to modulate certain genes (Dixit et al., 2016). By observing the changes, we could identify genes that are regulated by the perturbed gene. While Perturb-Seq has high potential, it is also much more expensive to carry and has various limitations. In this thesis, when we are talking about GRN inference, we are focusing on GRN inference based on regular single cell RNA-seq data.

Single cell data often is transformed into tables of genes or transcript counts, where the rows are cells and the columns are genes. Technically, there is no upper-bound on the number of cells. Today, usually you may expect the number of cells ranges from several hundreds to less than 1 million. Since the human genome has roughly 20,000 protein-coding genes, in most cases, we would expect about 20,000 or fewer columns. If we were considering the entire 20,000 genes in the regulatory network, we would need to infer the existence status of 400 million edges in a square adjacency matrix. This high dimensionality of the problem is in fact one of the major challenges in this field (Pratapa et al., 2020). Another outstanding challenge is the noisy nature of the data. Key among the recognized patterns of noise in single-cell data is “dropout” (Ghazanfar et al., 2016), where transcripts, disproportionately those with low or moderate expression levels, are lost from the expression counts. Other work has shown that even nonzero counts can be subject to technological background noise (Janssen et al., 2023).

1.3 Beyond Genes

1.3.1 Network Inference Beyond Genes

To summarize, the problem we are dealing with in the task of single-cell based GRN inference is that, given a 2-dimensional matrix, where rows are samples, columns are features, and values are counts of transcripts, infer a network that represents the (hopefully causal) dependencies among features. As you might be thinking right now, this is a common task across multiple disciplines of science. In fact, network inference can be applied to a broad range of topics beyond genetics (Pearl, 2009; Uhler et al., 2013; Spirtes and Zhang, 2016; Pingault et al., 2018; Glymour et al., 2019). For example, in economics, network analysis can explain the relationships between people's social-economic factors and their income. Such models may help create more accurate social-support policies. In epidemiology, networks could be used to describe the leading causes of a disease, which could lead to better diagnosis criteria. Similarly, in large-system robotics, network models are used to describe the interconnections within complex systems, such as sensor networks in a big factory or the coordination of multi-agent robotic systems. These examples show how network inference tools can be used beyond understanding gene relationships and make the impact more generalizable. Here we will have a short discussions of some common aspects to consider in general network inference.

1.3.2 Correlation, Causality, and DAGs

In almost every introduction to statistics class, a commonly mentioned expression is that “Correlation is not causality!!”. Technically, correlation refers to a statistical relationship between two variables, indicating that they tend to vary together but without implying a cause-and-effect relationship. Causality, on the other hand, implies that one variable directly influences the other. Typically, we may consider correlating relationships as a superset of all the causal relationships (but of course, not always). If we observe that factor A appears to be correlated with factor B, we may expect to see one of the following cases:

1. **A is causing B.**
2. **B is causing A.**
3. A is causing C, which may be directly or indirectly causing B.
4. B is causing C, which may be directly or indirectly causing A.
5. C is causing both A and B.
6. ...
7. Pure coincidence

Among these correlations, only the first two cases could be considered causal relationships.

Without perturbation, it is difficult to identify the direction of these relationships. A commonly used model in statistics and graph theory is the directed acyclic graph (DAG) (Spirtes and Glymour, 1991). By definition, DAGs are graphs without cycles so they rule out at least one from case 1 and case 2 above. They also rule out larger cycles (e.g., A is causing B, B is causing C, and C is causing A). Conceptually speaking, if we have to make a decision between case 1 and case 2, we can calculate and compare the likelihood of these two DAG structures and simply select the one that's more likely. This is in fact the main concept in the field of causal discovery (Zheng et al., 2018).

Note that while DAGs have been proven to be useful models for general causal discovery, their applications in the field of GRN inference is still an open research question. As explained above, for GRNs, our typical expectation is that they may have many feedback loops, such as the p53-MDM2 pair in the *TP53* example. Whether the DAG assumption should be applied directly here requires careful consideration in terms of model design and assumptions.

1.3.3 Confounding factors

Another important issue to consider is the existence of confounding factors. Case 5 (C is causing both A and B) is a perfect example to illustrate the point. When C is causing both A and B and we can't really observe C, we may falsely believe that there are causal dependencies between A and B. One widely known and discussed example of confounding effect is Simpson's paradox (Bickel et al., 1975).

Latent confounding effects are common problems for general causal discovery. When data is actively collected via observation (e.g. through clinical questionnaires, manually designed metrics, or planted sensors), it is very difficult to eliminate the existence of latent confounding factors. For GRN inference, it is a little less concerning because most of our data come from well-controlled experiments and all of our protein-coding genes are observable. However, a commonly used practice in single-cell analysis may increase the chance of introducing these errors into analysis. As mentioned before, the high dimensionality of our gene data is one of the major challenges in GRN inference. Without a scalable algorithm, a common practice is to only focus on the most variable genes (typically ranging from 10 to a few hundreds). Because of this step, many genes become unobservable to our model and that may lead to some downstream errors

1.4 Research Questions and Scope

In this thesis, we are studying scalable network inference methods that could be used to learn gene regulatory relationships from single-cell gene expression data. We will tackle the two major challenges in this field, namely noise and high dimensionality, by *utilizing* noise. At the end, we will extend these effort from GRN inference to the more general causal discovery setting.

Chapter 2

Background

In this chapter, we will start with a literature review of existing methods to construct GRNs. Then we will move on to a review of machine learning / deep learning methods that will be utilized in the following chapters. At the end, we will also review some concepts in the area of network inference and causal learning.

2.1 Existing Methods to Construct GRNs

This section is an extended version of section 1.2.2 and we will review what do people usually do to construct GRNs today. As mentioned previously, GRNs are usually either obtained by querying from existing regulatory databases or computationally inferred from experimental data.

2.1.1 Existing Regulatory Databases

We have made a lot of progress in terms of understanding how genes regulate each other but the findings are usually scattered in the scientific literature. To solve this problem, researchers have attempted to gather such information into databases based on various criteria. These criteria determines the “flavor” and use cases of each database. Databases such as ENCODE (ENCODE Project Consortium, 2012), ChIP-Atlas (Oki et al., 2018; Zou et al., 2022, 2024), and TRRUST (Han et al., 2015, 2018) provide experimentally validated interactions between transcrip-

tion factors and target genes, while resources like STRING (Szklarczyk et al., 2019, 2023) and GeneMANIA (Warde-Farley et al., 2010) compile predicted or experimentally supported interactions of different types from various studies. These resources are essential for developing a comprehensive understanding of GRNs, either by validating new inferences or by serving as a foundation for computational analysis.

- The ENCODE project is a very comprehensive efforts to collect functional elements in the human genome. It provides experimentally validated interactions between TFs and their target genes. This extensive collection of data serves as a critical resource for understanding gene regulation, but it focuses on mostly human data (and mouse data).
- ChIP-Atlas is a large collection of ChIP-seq and DNase-seq data from thousands of studies. Unlike ENCODE, which is highly curated, ChIP-Atlas provides an expansive collection of ChIP-seq data across various organisms and conditions. ChIP-Atlas allows for the exploration of binding sites of transcription factors across different tissues and conditions and is great for data mining purposes.
- TRRUST is another valuable resource, focusing specifically on human and mouse transcriptional regulatory networks. Unlike ENCODE and ChIP-Atlas, TRRUST compiles both experimental data and literature-derived regulatory relationships, offering a more curated and concise set of interactions. It is particularly useful for researchers interested in studying well-established regulatory pathways in humans and mice, providing a higher level of confidence in the reported interactions.
- STRING and GeneMANIA, on the other hand, take a different approach by compiling predicted or experimentally supported interactions from a variety of sources. STRING integrates data from multiple types of evidence, including co-expression, text mining, and experimental validation, to provide a comprehensive network of predicted protein-protein and protein-DNA interactions. It

is a valuable resource for hypothesis generation, especially when experimental data is lacking. However, the inclusion of predicted interactions means that users must be cautious about the reliability of individual links.

- GeneMANIA is similar to STRING. It places a greater emphasis on co-expression and functional association data, making it particularly useful for identifying genes that may work together in similar biological processes (Taye et al., 2017). GeneMANIA also provides a user-friendly interface for visualizing gene networks in cytoscape (Montejo et al., 2010), making it a great choice for data exploration.

While curated databases do offer valuable insights, they also have significant shortcomings that all users should be aware of. First of all, as of today, it's probably still not a good idea to consider these databases as complete. Scientific research is often limited by the general research interests and the knowledge we currently know. While we do have extensive knowledge of regulatory pathways related to popular genes such as *TP53*, we often have very limited knowledge of other less studied genes. Second, the interactions collected in these databases are often not context specific, or not specific to the context you need. As mentioned above, the existence of the interactions between *TP53* and cell apoptosis does not mean these pathways are always activated. At the same time, many of the links collected by these regulatory databases come from ChIP-seq, a technique that analyzes how proteins interact with DNA (Park, 2009). It is certainly true that if we could observe a binding site on the TF protein, we are more likely to observe a true regulatory relationship. However, as shown in the experiments done by Cusanovich et al. (2014), most identified binding interactions (> 85%) do not result in measurable changes in gene expression levels in target genes. The correct way of utilizing ChIP-seq to identify regulatory links requires the integration of other data sources and control of false positive rates (Laajala et al., 2009; Bailey et al., 2013) but these practices are not always in place.

2.2 Transcriptomics based GRN Inference

Despite increasing experimental data and a growing recognition of the binding sites of many regulatory proteins, GRNs are often inferred primarily from transcriptomic data (Mercatelli et al., 2020; Van Dam et al., 2018; Slonim, 2002). Such data sets offer unique functional *in vivo* snapshots of how genes are expressed in particular cells or under various conditions.

Over the past 30 years, many methods have been developed for GRN inference based on transcriptomic data. Some methods were originally developed for microarray or bulk RNA sequencing data sets but they are still being used on some single-cell datasets (usually after heavy gene filtering). In general, these methods can broadly be classified into four categories, respectively relying on mutual information, decision trees, Bayesian networks, and graph models.

- Mutual information is a measure of the statistical dependence between two random variables. In the context of GRN, it's used to quantify the amount of information that one gene provides about the expression of another gene. One common MI approach is to rank all gene pairs by the computed mutual information. Such methods, including ARACNE (Margolin et al., 2006), CLR (Faith et al., 2007), and MRNET (Meyer et al., 2007), are among the first set of methods proposed for GRN inference on bulk data. The research on applying MI methods on scRNA-seq data is active, and a good example is PIDC (Chan et al., 2017), which calculates the MI between triplets of genes in order to disentangle confounding factors. These MI-based methods are known for their simplicity and ability to capture non-linear dependencies.
- Decision trees are another major class of GRN inference tools. Decision trees classify or regress data based on a series of decision rules, and in the context of GRN, they predict the expression of a target gene based on the other genes. The predicted links are ranked based on feature importance metrics, typically evaluated using information gain. These feature importance values are then

extracted and used as weights for the predicted edges. Methods like GENIE3 (Huynh-Thu et al., 2010) and GRNBoost2 (Moerman et al., 2019) fall into this category. GENIE3, which won the DREAM4 challenge in 2010 (Greenfield et al., 2010), remains a popular choice for GRN inference, especially for single-cell data. Its strength lies in its ability to model complex relationships between genes without assuming linearity. GRNBoost2, an adaptation of GENIE3 using gradient boosting, is another effective method that offers improved speed and scalability. Researchers who developed GRNBoost2 further developed SCENIC (Aibar et al., 2017) and SCENIC+ (Bravo González-Blas et al., 2023), a popular suite of tools to prune and clean the predicted edges from GENIE3 and GRNBoost2.

- Bayesian networks and dynamic bayesian networks are another approach to GRN inference, framing the problem as a causal discovery task. BNs represent the regulatory relationships as Directed Acyclic Graphs (DAGs), where nodes represent genes and edges represent potential causal interactions. One limitation of BN approaches is the DAG constraint, which precludes regulatory loops that are common in gene networks. DBNs extend BNs by incorporating temporal dynamics, effectively modeling interactions across different time points to capture self-regulation and temporal dependencies. Methods like G1DBN (Lèbre, 2009) and ebdbNet (Rau et al., 2010) are popular in this category. While BN/DBN-based methods offer valuable insights into causality, they are computationally intensive and may struggle with scalability when applied to large datasets.
- Graph based approaches such as PANDA (Passing Attributes between Networks for Data Assimilation) (Glass et al., 2013) and tools in the networkZoo package (Ben Guebila et al., 2023) provide another powerful method for GRN inference. PANDA is a message-passing algorithm that integrates multiple types of biological data, including protein-protein interactions, gene expression, and transcription factor binding data, to iteratively refine a regulatory

network. This approach is particularly useful for integrating heterogeneous data sources to derive context-specific GRNs. `networkZoo` expands on the principles of PANDA to provide a suite of tools for network analysis, supporting inference across a variety of biological contexts and data types. These methods excel at leveraging complementary data to improve inference accuracy, but they may require significant computational resources and careful parameter tuning to achieve optimal results.

Additionally, time series data can be modeled using ordinary differential equation (ODE) frameworks, as seen in `dynGENIE3` (Huynh-Thu and Geurts, 2018), or through Granger causality approaches, like `GRNVBEM` (Sanchez-Castillo et al., 2018).

Despite all this prior work, finding high-accuracy networks from transcriptomic data alone is still not adequately solved (Pratapa et al., 2020). For single cell based methods, major of these methods scale poorly and could not finish within reasonable time, especially if we would like to consider more genes and more cells into the computation. In this thesis, one of our objectives is to develop good GRN inference methods that work well on typical single cell data without too much gene or cell filtration.

2.3 Review of related deep learning methods

In this section, we will review a few different deep learning methods that are essential for the rest of the thesis.

2.3.1 Variational Autoencoder

The method described in Chapter 3 took its inspiration from variational autoencoder (VAE) (Kingma and Welling, 2013). A VAE consists of an encoder and a decoder, which respectively compress and reconstruct the input data through a low-dimensional latent representation. While classic autoencoders are trained using reconstruction loss alone, VAEs consider the latent variable as a random variable

sampled from a parameterized distribution. As a result, VAEs are trained by optimizing the sum of reconstruction loss and the Kullback-Leibler (KL) divergence between the posterior and prior distributions, or the evidence lower bound (ELBO). This KL divergence term in the objective function encourages a more evenly distributed latent space. This evenness helps to ensure that values in the latent space remain meaningful.

VAEs have been widely applied in various fields, including computer vision and natural language processing, for tasks such as representation learning (encoder) and content generation (decoder). They offer several advantages, including the ability to process noisy data and the requirement for no additional labeled data. These advantages make VAEs well-suited for handling noisy single-cell data. For example, Wang and Gu (2018) proposed using VAEs as a dimension reduction tool and demonstrated their effectiveness in cell population clustering and visualization. Grønbech et al. (2020) introduced scVAE, which uses VAEs to estimate expected gene expression values and improve cell clustering.

2.3.2 Denoising Diffusion Probabilistic Models

Another emerging methodology that has shown potential in various domains, specifically computer vision, is that of diffusion probabilistic models (Ho et al., 2020; Sohl-Dickstein et al., 2015). By simulating the process of diffusion, or the spread of information, diffusion models are likelihood-based methods that aim to restore data from Gaussian corruption by an iterative process. In Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020), a diffusion probabilistic model, as illustrated in Figure 2.1, consists of a non-parameterized forward process, which adds small amounts of noise to the data for each of a number of steps, and a parameterized reverse process, which reconstructs a less-noisy version of the data. The reverse process is trained on predicting the added noise so that the de-noised data can be recovered by subtracting the predicted noise from the noisy data. Recent studies have suggested similarity between diffusion models and a generalized form of VAE with many latent spaces (Luo, 2022). While a classic VAE enforces normality in

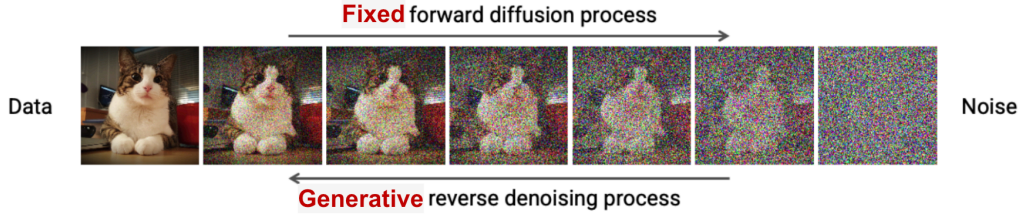


Figure 2.1: Illustration of the forward and backward processes of a diffusion model for computer vision. Figure is adapted from Kreis et al. (2022).

the center latent space with KL divergence, diffusion models enforce a trajectory to normality by the diffusion process. In addition, some studies further connect diffusion models with score matching models via annealed Langevin dynamics sampling (Song et al., 2021). This finding puts the diffusion process within the framework of stochastic differential equations (SDE) and enables continuous diffusion steps. To date, there are a few very recent preprints using diffusion models for single-cell data (Tang et al., 2023), and none of these consider modeling the regulatory relationships among genes. Here we provide a detailed description of the forward and reverse process.

2.3.2.1 Forward process

Given a collection of input data $\mathbf{x}$, we are aiming to create a generative model that can draw from the same distribution. The forward process, denoted by $q(\mathbf{x}_T|\mathbf{x}_0)$, is a non-parameterized Markov Chain that transforms an original unperturbed gene expression vector $\mathbf{x}_0$ to Gaussian noise $\mathbf{x}_T$ in T steps (eq. 2.1). This process generates a series of noisy samples $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_T$ and as it proceeds, $\mathbf{x}$ gradually loses its distinct features as illustrated in Figure 2.1.

$$q(\mathbf{x}_T|\mathbf{x}_0) = \prod_{t=0}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}) \quad (2.1)$$

In each step of the forward process, x_{t-1} was perturbed according to a constant $\beta_t \in (0, 1)$, which is part of a pre-generated noise schedule β . The value of

β_t is usually very small. Intuitively, β_t represents the amount of information loss at each step. In the original DDPM paper (Ho et al., 2020), the authors proposed using the transformation function shown below (eq. 2.2; following their notation, the first parameter is simply the perturbed input at time t , while the second and third parameters are the mean and variance of the Normal distribution at that time). This transformation uses β_t to control both the mean and variance of the sampling process. The fact that $\sqrt{1 - \beta_t}$ is always less than 1 ensures that after a number of steps, the mean of the final distribution will approach 0:

$$(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \quad (2.2)$$

In practice, however, it is not feasible to do iterative sampling during training. To solve this problem, the authors of DDPM proposed a reparameterization trick that enables the generation of $\mathbf{x}_t$ from $\mathbf{x}_0$, or $q(\mathbf{x}_t | \mathbf{x}_0)$, in one single step. We briefly explain this technique here, as it is crucial for our method.

Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=0}^t \alpha_i$. The reparameterization approach generates standard Gaussian noise $\mathbf{z} = \mathcal{N}(0, 1)$. We can rewrite the transformation with the following equations:

$$\begin{aligned} \mathbf{x}_t &= \sqrt{1 - \beta_t} \mathbf{x}_{t-1} + \sqrt{\beta_t} \mathbf{z}_{t-1} \\ &= \sqrt{\alpha_t} \mathbf{x}_{t-1} + \sqrt{1 - \alpha_t} \mathbf{z}_{t-1} \\ &= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{(1 - \alpha_t) + \alpha_t(1 - \alpha_{t-1})} \mathbf{z}_{t-2} \\ &= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}} \mathbf{z}_{t-2} \\ &= \dots \\ &= \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \mathbf{z} \end{aligned} \quad (2.3)$$

As mentioned above, the diffusion schedule β is pre-calculated. Therefore, the forward process at each iteration can be simplified as a non-parameterized func-

tion $f_{forward} : \mathbf{x}_0 \times \mathbf{z} \times t \rightarrow \mathbf{x}_t$. To be more specific, at each iteration, each cell gets a different time step t , which determines the strength of the perturbation. Then the forward process follows Equation 2.3 and generates the perturbed data $\mathbf{x}_t$ at the time step t using a randomly sampled standard Gaussian noise $\mathbf{z}$.

2.3.2.2 Reverse Process

Recall that the reverse process is a Markov Chain aiming to recover the original input $\mathbf{x}_0$ from $\mathbf{x}_T$. Since computing the exact reverse step $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is intractable, we use a parameterized model $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to approximate this process.

$$p_\theta(\mathbf{x}_0) = p(\mathbf{x}_T) \prod_{t=T}^0 p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(x_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_\theta(\mathbf{x}_t, t))$$

For simplicity, we chose to use the simplified loss function proposed in the DDPM paper:

$$L_{simple}(\theta) = \mathbb{E}_{\mathbf{x}_0, t, \mathbf{z}} [||\mathbf{z} - \mathbf{z}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\mathbf{z}, t)||^2] \quad (2.4)$$

In other words, we need to create a model $p_\theta(\mathbf{z}|\mathbf{x}_t, t)$ to estimate the standard Gaussian noise $\mathbf{z}$ used in the forward pass (Eq. 2.3). In computer vision, this challenging problem is usually resolved using image segmentation models, such as U-Net (Ronneberger et al., 2015), to ‘segment’ the noise.

2.4 Review of network inference and causal learning

2.4.1 Problem Statement

Given a dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n is the number of samples and d is the number of feature variables, the objective is to learn a meaningful dependency graph $\mathcal{G}$

represented by the weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$.

2.4.2 Structural Equation Models (SEMs)

Structural Equation Models (SEMs) (Grace, 2006; Shipley, 2016; Kline, 2023) provide a framework to model variable dependencies. For a linear SEM, we simply assume that each variable is a linear combination of its parents with some noise. In its matrix multiplication form, we have

$$\mathbf{X} = \mathbf{X}\mathbf{W} + \mathbf{E}, \quad (2.5)$$

where $\mathbf{E} \in \mathbb{R}^{n \times d}$ captures the error terms. Based on this assumption, many existing SEMs aim to estimate matrix $\mathbf{W}$ such that the reconstruction error is minimized (van de Geer and Bühlmann, 2013; Loh and Bühlmann, 2014; Zheng et al., 2018). Since the adjacency matrix is often sparse, many methods choose to add either L1 or L2 regularization on $\mathbf{W}$ to encourage sparsity (Vowels et al., 2022). In this case, we have the following training objective,

$$\min_{\mathbf{W}} \frac{1}{2n} \|\mathbf{X} - \mathbf{X}\mathbf{W}\|_F^2 + \lambda_1 \|\mathbf{W}\|_1 + \lambda_2 \|\mathbf{W}\|_2 \quad (2.6)$$

To extend the use of SEMs to real world applications, where nonlinearity is common, we rewrite Equation 2.5 in the following form, where f is a nonlinear transformation function:

$$\mathbf{X} = f(\mathbf{X}; \mathbf{W}) + \mathbf{E} \quad (2.7)$$

In practice, Equation 2.7 allows too much freedom of model formulation. Therefore, people often use the following equation to describe a nonlinear SEM, where f_1 is the nonlinear transformation function for $\mathbf{X}$ before it meets $\mathbf{W}$ and f_2 is the nonlinear transformation function for the product of graph propagation (Yu et al., 2019; Ng et al., 2019).

$$\mathbf{X} = f_2(f_1(\mathbf{X})\mathbf{W}) + \mathbf{E} \quad (2.8)$$

2.4.3 DeepSEM

DeepSEM (Shu et al., 2021) builds on top of the linear additive assumption in Equation 2.5 and the VAE model as we discussed in 2.3.1. Following a similar transformation in DAG-GNN (Yu et al., 2019), the authors of DeepSEM rearrange the terms so they can be rewritten in the following forms:

$$\mathbf{E} = \mathbf{X}(\mathbf{I} - \mathbf{W}), \quad (2.9)$$

$$\mathbf{X} = \mathbf{E}(\mathbf{I} - \mathbf{A})^{-1}, \quad (2.10)$$

If we treat $\mathbf{E}$ as the latent variable of a VAE, denoted by Z_E , Equation 2.9 and 2.10 naturally fit into the role of encoder and decoder in VAE. By treating the adjacency matrix $\mathbf{W}$ as model parameters, this model could be trained with the objective of reconstructing the input data. To control the sparsity of the learned graph, DeepSEM is applying L1 loss on the adjacency matrix. Together, the complete objective is to minimize the following function. In this function, the first term measures the reconstruction loss, the second term is the KL divergence which pulls the latent variable closer to standard Gaussian distribution, and the third term adds L1 regularization on the adjacency matrix to encourage sparsity.

$$\text{Loss} = -\mathbb{E}_{z_E \sim q_\phi(Z_E|X)}[\log p_\theta(X|Z_E)] + \beta D_{KL}(q_\phi(Z_E|X)||p_\theta(Z_E)) + \alpha \sum |W| \quad (2.11)$$

On a recent collection of benchmark datasets called BEELINE, DeepSEM reports better performances than many popular GRN inference methods, such as GENIE3 and GRNBoost2. However, its performance is not very stable on each single run of this algorithm. To overcome this problem, the authors of DeepSEM

propose to execute the algorithms 10 times and then ensemble the results.

In terms of causality, DeepSEM does not apply any constraint on the acyclicity of the learned graph. However, the L1 regularization does add some small directionality to the learned graph (Vowels et al., 2022). For a strong pair of links, you would expect to see edges going in both directions but one edge is stronger than the other. Still, overall, the learned graph is fairly symmetric.

2.4.4 Continuous DAG constraint

Traditional causal discovery approaches often rely on combinatorial optimization, which becomes computationally infeasible for large graphs. To address this issue, Zheng et al. (2018) proposed a method called NOTEARS that introduced a continuous score characterizing graph acyclicity:

$$h(\mathbf{W}) = \text{tr}(e^{\mathbf{W} \circ \mathbf{W}}) - d, \quad (2.12)$$

where $\circ$ is the Hadamard product, $e^{\mathbf{W}}$ is the matrix exponential of $\mathbf{W}$, and $\text{tr}()$ is the trace of a matrix. Essentially, matrix $\mathbf{W}$ is a DAG if and only if $h(\mathbf{W}) = 0$. Since the function $h(\mathbf{W})$ has a simple and smooth gradient function, it can be used in many gradient-based continuous optimization algorithms. In NOTEARS, the DAG $\mathbf{W}$ is learned by optimizing a modification of Equation 2.6 (with only the L1 loss) while keeping $h(\mathbf{W})$ near zero with an augmented Lagrangian method using a L-BFGS optimizer. In terms of complexity, since the score function $h(\mathbf{W})$ requires the matrix exponential, the runtime of NOTEARS is at least $O(d^3)$.

Chapter 3

Improved GRN inference from single cell data with dropout augmentation

3.1 Introduction

The application of neural networks (NNs) in the analysis of single-cell data has advanced rapidly in the last couple of years. As mentioned in Chapter 2, one of the leading NN-based GRN inference methods, DeepSEM, (Shu et al., 2021) parameterizes the adjacency matrix and uses a variational autoencoder (VAE) (Kingma and Welling, 2013) architecture optimized on reconstruction error. In fact, on the BEELINE benchmarks (Pratapa et al., 2020) where the “right” networks are (approximately) known, DeepSEM reports better performance than other methods and runs significantly faster than most.

However, as shown later in this chapter, one of the issues with DeepSEM is that as training continues, the quality of the inferred networks may degrade quickly. A possible explanation is that soon after the model converges, it may begin to over-fit the dropout noise in the data.

Single-cell data is often characterized by an excessive number of zero ex-

pression counts, referred to as “zero-inflation.” For example, in the nine data sets examined in (Ghazanfar et al., 2016), 57 to 92 percent of the observed counts are zeros. Among these zero values, “dropout” describes the situation when transcripts, often those with low or moderate expression in a cell, are not counted by the sequencing technology. Later droplet-based protocols, such as inDrops (Klein et al., 2015) and 10X Genomics Chromium (Zheng et al., 2017), helped improved detection rates. However, the “dropout” problem still persists, as even recent methods have relatively low sensitivity (Rao et al., 2021; Kang et al., 2024).

Therefore, there has been research into data imputation methods for use in single-cell analysis. Several methods have been proposed to identify and replace missing data with imputed values (Li and Li, 2018; Arisdakessian et al., 2019; Rao et al., 2021; Qi et al., 2023; Kang et al., 2024). Yet many of these methods depend on restrictive assumptions, and some require additional information, such as GRNs or bulk transcriptomic data.

In this chapter, we introduce two novel contributions to the fields of single cell analysis and GRN inference. First, we propose “dropout augmentation” (DA), a novel approach to solve the zero-inflation problem by augmenting the data with a small amount of simulated dropout noise. We found that this idea, while seemingly counter-intuitive, can effectively regularize models so that they remain robust against dropout noise.

It has long been known that by adding noise to the input data during training, we can improve the robustness, and sometimes even the performance, of many machine learning models. Bishop first pointed out that adding noise is equivalent to Tikhonov regularization (Bishop, 1995). Hinton further introduced the idea of using random “dropout” on either input or model parameters to improve training performance (Hinton et al., 2012). Thus, the theoretical foundations of DA are also solid.

Our second contribution is the DAZZLE model, or *Dropout Augmentation for Zero-inflated Learning Enhancement*. DAZZLE uses the same VAE-based GRN learning framework introduced by DeepSEM and DAG-GNN (Shu et al., 2021; Yu

et al., 2019), but it employs dropout augmentation and several other model modifications. These include a new method optimizing the adjacency matrix sparsity control strategy, a simplified model structure, and a closed-formed prior. Compared to DeepSEM, DAZZLE shows better model stability and robustness in our benchmark experiments. We further illustrate how DAZZLE’s network inference facilitates interpreting typical-sized data sets efficiently, in this case explaining microglial expression dynamics across the mouse lifespan.

3.2 Methods

3.2.1 Dropout Augmentation

Previous research suggests that zero values in single cell data include both biologically real zeros, corresponding to truly absent genes, and random dropout events. A successful single-cell model should remain robust regardless of how dropout values are distributed. This idea informs the dropout augmentation algorithm.

Let $X \in \mathbb{R}^{n \times m}$ be a gene expression matrix from a single-cell experiment, where n is the number of cells and m is the number of genes. We randomly sample a proportion (p) of data and temporarily replace these values with zeros as shown in Figure 3.1. Alternatively, the augmented data could be treated as the sum of the original expression data X and a dropout noise term E , where E is the Hadamard product of negative X and a masking term derived from Bernoulli sampling. By denoting the mask of dropout augmentation as M_{DA} and the probability of augmented dropout by p , we can write the dropout noise term E and the augmented data X' as follows:

$$E = -X \odot M_{\text{DA}}, \text{ where } M_{\text{DA}}^{ij} \sim \text{Bernoulli}(p) \quad (3.1)$$

$$X' = X + E. \quad (3.2)$$

In theory, the dropout augmentation algorithm could be applied to any it-

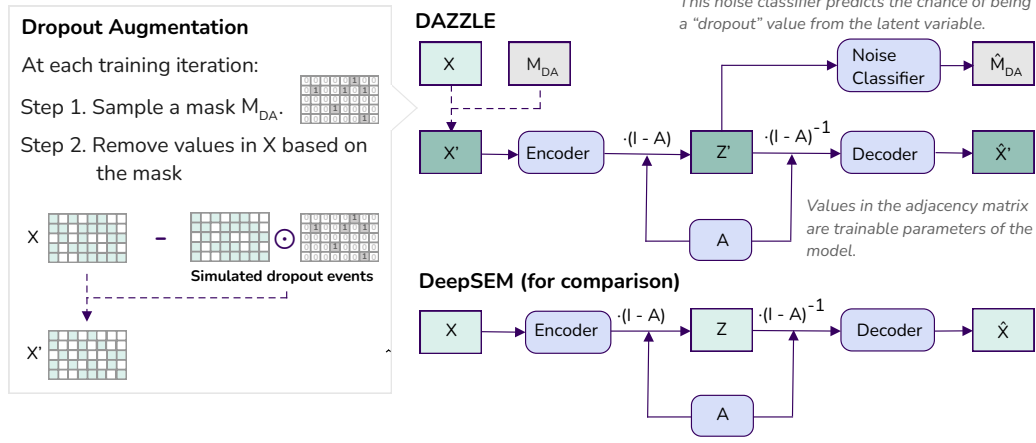


Figure 3.1: One of the major differences between DAZZLE and DeepSEM is the use of Dropout Augmentation. Dropout augmentation regularizes model training by simulating small amounts of random dropout at each training iteration such that the model is protected against the negative impact of dropout noise. Rounded boxes are trainable model parameters.

erative learning algorithm, such as expectation maximization, gradient boosting, or neural networks. As shown in Figure 3.1, M_{DA} is re-sampled at every training step, so the augmented data X' changes in every iteration. During the entire training process, the model is only exposed to the altered X' instead of X .

3.2.2 GRN Inference with DAZZLE

The task of GRN inference is to infer a weighted adjacency matrix $A \in \mathbb{R}^{m \times m}$ based on the expression data X . Previous methods DAG-GNN (Yu et al., 2019) and DeepSEM (Shu et al., 2021) rely on a linear additive assumption that can be written as

$$X = XA + Z, \quad (3.3)$$

where $Z \in \mathbb{R}^{n \times m}$ is a random variable characterizing the noise, essentially describing the gap between the overall expected counts of the genes based on their regularizers (XA) and the observed counts (X). To embrace the idea that the observed data are noisy due to dropout, we modify this assumption and rewrite equation 3.3 as:

$$X' = X'A + Z', \quad (3.4)$$

where Z' is defined for X' analogously to the definition of Z for X . Since dropout is so prevalent in single-cell data, we believe equation 3.4 describes the actual situation more accurately. Following a similar transformation to that done in DAG-GNN and DeepSEM, by rearranging the terms we can rewrite equation 3.4 in the following two forms:

$$Z' = X'(I - A), \quad (3.5)$$

$$X' = Z'(I - A)^{-1}. \quad (3.6)$$

Equation 3.5 infers Z' from X' and Equation 3.6 is a generative model that reconstructs X' based on the noise sum. These two equations naturally fit into a VAE framework with Equation 3.5 as an encoder and Equation 3.6 as a decoder. When we parameterize both the VAE model and the adjacency matrix, the encoder can be denoted by $q_\phi(Z'|X')$ and the decoder can be denoted by $p_\theta(X'|Z')$, with A being part of ϕ and θ . In this case, Z' is the latent variable.

For a VAE, the problem of finding the set of parameters θ that maximizes the log evidence $\log(P(X'))$ is intractable. Instead, it is common to maximize the evidence lower bound (ELBO), which we write as

$$\begin{aligned} ELBO = & -D_{\text{KL}}(q_\phi(Z'|X')||p_\theta(Z')) \\ & + \mathbb{E}_{z' \sim q_\phi(Z'|X')}[\log p_\theta(X'|Z')], \end{aligned} \quad (3.7)$$

where the first term is the KL divergence and the second term can be thought of as the reconstruction loss.

The random variable Z' describes the deviation of the observed value X' from the expected value $X'A$. In a particular cell, if the expression value of a particular gene happens to be observed as 0 due to dropout events, we will more likely see a

larger deviation Z' . In other words, Z' contains information that can be used to infer whether a value comes from a dropout event. Following this rationale, we can add a classifier C_{DA} based on the specified dropout augmentation masked M_{DA} , as shown in Equation 3.8 below. As a naïve approach, here we choose a simple 3-layer multi-layer perceptron (MLP) to do the classification:

$$\hat{M}_{\text{DA}} = \text{sigmoid}(C_{\text{DA}}(Z')). \quad (3.8)$$

The loss function of this classifier is simply a binary cross entropy function.

$$L_{\text{BCE}} = -\mathbb{E} \left[M_{\text{DA}} \log \hat{M}_{\text{DA}} + (I - M_{\text{DA}}) \log (I - \hat{M}_{\text{DA}}) \right] \quad (3.9)$$

The dropout augmentation classifier can be trained either separately or together with the main model using the same optimizer. In our experiments, we add the classification loss to the ELBO function scaled by a hyper parameter χ . Additionally, following the design of DeepSEM, we include an L1 sparse loss term that regulates the sparsity of the learned adjacency matrix. The final form of the objective function is to minimize the following loss function:

$$\begin{aligned} \text{Loss} = & -\mathbb{E}_{z \sim q_\phi(Z|X)} [\log p_\theta(X|Z)] \\ & + \alpha \sum |A| \\ & + \beta D_{\text{KL}}(q_\phi(Z|X) || p_\theta(Z)) \\ & + \gamma L_{\text{BCE}}(M_{\text{DA}}, \hat{M}_{\text{DA}}). \end{aligned} \quad (3.10)$$

We made several additional model design and training choices that further distinguish our model from that of DeepSEM. First, we improved the stability of the model by delaying the introduction of the sparse loss term by a customizable number of epochs. Another difference is that, to estimate the prior, DeepSEM estimates a separate latent variable while DAZZLE uses a closed-form Normal distribution. These changes lead to reduced model sizes and computational time, especially for

large networks. Finally, while DeepSEM is trained with two separate optimizers in an alternating manner (one on the adjacency matrix and the other on the rest of the neural networks), DAZZLE is trained using a single optimizer with different learning rates. This improvement helps DAZZLE stay modular, so it can be integrated with other network components more easily in the future.

3.2.3 Experiment Design

To test the effectiveness of DA and DAZZLE, we tested our methods on both standard benchmark datasets for GRN inference and a large real-world datasets on microglia with multiple time points. In terms of scales, the benchmark datasets typically have a few hundred of cells and roughly 1,500 gene. The microglia datasets have 49,972 cells in total and roughly 15,000 genes at each time point.

3.2.3.1 Datasets Used

BEELINE single-cell benchmarks The BEELINE benchmarks consist of both synthetic expression data based on curated ground truth networks, as well as seven pre-processed real single-cell RNA-seq datasets (Pratapa et al., 2020). These scRNA-seq datasets come from both human and mouse samples and have undergone different pre-processing steps, including normalization, depending on the original data format (e.g. raw counts or processed data). In some aspects, this variety reflects the wide array of differences we encounter in real-world data.

Next, BEELINE combines the scRNA-seq data with three different sources of “ground truth” data about regulatory relationships, including the functional interaction network represented in the STRING database v11 (Szklarczyk et al., 2019), non cell-type specific transcriptional networks, and cell-type specific networks. The Non-specific ChIPSeq network combines links from DoRothEA (Garcia-Alonso et al., 2019), RegNetwork (Liu et al., 2015), and TRRUST v2 (Han et al., 2018). The cell-type specific networks were created by the BEELINE authors for each dataset by searching through the ENCODE (ENCODE Project Consortium, 2012), ChIP-Atlas (Zou et al., 2022), and ESCAPE (Xu et al., 2013) databases. To generate a

benchmarking dataset, BEELINE identifies highly variable transcription factors and genes and randomly samples from this pool to create a benchmark of the desired size.

Hammond microglial data To assess DAZZLE’s performance in a more practical context, we use a published data set from (Hammond et al., 2019) (data available from NCBI’s Gene Expression Omnibus database (Edgar et al., 2002) under accession number GSE121654). The Hammond mouse microglial dataset includes RNA sequencing counts for cells underlying several possible comparisons. In our analysis, we selected the data from five mouse developmental stages, each of which includes single cell data from four healthy male mice. To preprocess the data, following suggestions from (Green et al., 2022) for the same data, we filtered out cells with fewer than 400 or more than 3,000 unique genes, cells with more than 10,000 UMIs, and cells with over 3% of reads mapping to mitochondrial genes.

Many standard analysis approaches further reduce the data set size by filtering the gene set, keeping only the most variable genes. However, here, we only remove genes with a raw count of zero transcripts detected in all cells. We further removed all gene models, mitochondrial genes, and ribosomal genes from this pool to simplify the interpretation of the resulting networks. Roughly 10% of genes are removed in this step for each time point. The expression values were normalized using natural log transformation.

After this cell and gene filtering, the final data set includes 49,972 cells from five time points across the mouse lifespan: Embryonic (embryonic day E14, 11,262 cells and 15,673 genes), Early Postnatal (postnatal day P4/5, 13,316 cells and 15,039 genes), Late Juvenile (P30, 9,431 cells and 13,929 genes), Adulthood (P100, 8,259 cells and 13,998 genes), and Old Age (P540, 7,704 cells and 14,140 genes). Note that compared to the original paper (Hammond et al., 2019), here we are using a very different approach, analyzing changes in potential regulatory links across time, rather than attempting to identify microglial subpopulations defined by specific injury-responsive cell clusters.

3.2.3.2 Evaluation metrics

In this report, we follow the recommendations from the BEELINE paper and use Area Under the Precision Recall Curve Ratio (AUPRC Ratio) and Early Precision Ratio as the main evaluation metrics. Compared to the Area Under the Receiver Operating Characteristic (AUROC), AUPRC is a better metric when there is a class imbalance between the positive and negative groups (Saito and Rehmsmeier, 2015).

In this thesis, AUPRC is approximated using its discretized form without interpolation, as shown in Equation 3.11:

$$\text{AUPRC} = \int_0^1 P(R) dR \approx \sum_{n=1}^N P_n \cdot (R_n - R_{n-1}), \quad (3.11)$$

where $P(R)$ is the Precision at Recall level R , N is the number of unique predictions, and R_n and P_n are the Recall and Precision scores at item n . This approximation is often referred to as average precision (Manning et al., 2008) and is the same analysis performed in the DeepSEM paper. The AUPRC Ratio is calculated by dividing the calculated AUPRC score with the theoretical AUPRC score of a random predictor. In this case, the expected precision of a random predictor is equal to the proportion of positive cases. On the Precision-Recall plot, the performance of a random predictor forms a horizontal straight line (see Figure 5D in (Saito and Rehmsmeier, 2015)). Therefore, the AUPRC of a random predictor is simply the proportional of positive cases and we can calculate AUPRC Ratio with Equation 3.12.

$$\text{AUPRC Ratio} \approx \frac{\text{Positives}}{\text{Total Instances}} \cdot \sum_{n=1}^N P_n \cdot (R_n - R_{n-1}) \quad (3.12)$$

Early Precision (EP) is the fraction of true positives within the top- k candidates, where k is the number of edges in the ground truth network. Early Precision Ratio further divides that fraction by the expected Early Precision of a random predictor, which is the edge density of the ground truth network (Pratapa et al., 2020).

Since the number of all possible edges and the number of true edges are usually very large in actual GRNs, the values of the AUPRC and Early Precision themselves tend to be small. As claimed in the BEELINE paper, converting them to ratios make it easier to understand the performance across benchmarks.

3.3 Experiments

3.3.1 DAZZLE improves GRN inference on BEELINE benchmarks

We performed a benchmark comparison of DAZZLE to DeepSEM (Shu et al., 2021), GENIE3 (Huynh-Thu et al., 2010), and GRNBoost2 (Moerman et al., 2019) on the BEELINE single cell benchmarks, which include seven datasets (two from human and five from mouse) and three sets of ground truth networks. The details of the BEELINE benchmarks are described in Section 3.2.3.1 above. We chose to compare to GENIE3 and GRNBoost2 as representatives of non-deep-learning methods because these two decision-tree-based methods are among the most widely used GRN inference algorithms. Recent benchmarks on single cell data (Kang et al., 2021), (Pratapa et al., 2020) also showed that these methods generally outperformed SCODE (Matsumoto et al., 2017), ppcor (Kim, 2015), and SINCERITIES (Papili Gao et al., 2018). We also included DeepSEM, the model that inspired this work and the previous front-runner.

Note that since a single run of DeepSEM is not stable (a point that is discussed further in later Results sections). Thus DeepSEM was proposed as an ensemble algorithm, combining results from a set of 10 runs. Here, we include both single runs (1x) and the ensemble model (10x) in the comparison for both DAZZLE and DeepSEM. For fair comparison, both DAZZLE and DeepSEM use the same size neural networks, and both are trained for 120 iterations as suggested by the DeepSEM paper. The hyperparameter settings are identical except for the changes discussed in section 3.2.2.

The main benchmark results are provided in Table 3.1. The numbers reported are the average area under the precision recall curve (AUPRC) ratios over

Table 3.1: DAZZLE shows improved GRN inference capacity on BEELINE benchmarks

	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
# of Genes	1410	1448	1321	1620	1204	1132	692
# of Cells	758	425	383	421	1071	889	847
Ground Truth: STRING							
# of True Edges	5,149	9,000	5,898	8,479	1,826	1,311	154
GENIE3	1.98 (0.01)	<i>1.86 (0.01)</i>	<i>1.72 (0.01)</i>	2.05 (0.01)	4.19 (0.05)	6.27 (0.04)	7.02 (0.08)
GRNBoost2	1.67 (0.01)	1.50 (0.01)	1.43 (0.01)	1.88 (0.02)	3.65 (0.05)	5.16 (0.11)	7.07 (0.08)
DeepSEM-1x	2.00 (0.06)	1.66 (0.04)	1.55 (0.06)	2.20 (0.02)	5.27 (0.16)	5.93 (0.38)	7.28 (0.60)
DeepSEM-10x	2.10 (0.03)	1.82 (0.05)	<i>1.68 (0.02)</i>	<i>2.33 (0.03)</i>	5.68 (0.11)	<i>6.91 (0.12)</i>	7.47 (0.14)
DAZZLE-1x	<i>2.39 (0.05)</i>	1.80 (0.02)	1.58 (0.03)	2.27 (0.04)	5.87 (0.21)	6.56 (0.10)	<i>7.50 (0.07)</i>
DAZZLE-10x	<i>2.44 (0.02)</i>	1.82 (0.01)	1.62 (0.01)	<i>2.34 (0.03)</i>	<i>6.07 (0.03)</i>	6.63 (0.04)	<i>7.53 (0.05)</i>
Ground Truth: Non-celltype-specific ChIP-Seq							
# of True Edges	4,597	5,335	3,918	8,030	1,960	1,358	317
GENIE3	0.97 (0.00)	1.01 (0.01)	1.56 (0.01)	<i>1.65 (0.01)</i>	2.54 (0.02)	<i>3.56 (0.03)</i>	2.78 (0.04)
GRNBoost2	1.01 (0.01)	1.09 (0.01)	1.36 (0.02)	1.49 (0.02)	2.39 (0.04)	2.84 (0.03)	2.45 (0.05)
DeepSEM-1x	1.22 (0.05)	1.29 (0.08)	1.73 (0.07)	1.58 (0.05)	2.94 (0.32)	3.27 (0.15)	2.50 (0.56)
DeepSEM-10x	1.24 (0.01)	1.41 (0.03)	<i>1.93 (0.02)</i>	<i>1.66 (0.02)</i>	3.18 (0.06)	3.48 (0.12)	<i>3.01 (0.20)</i>
DAZZLE-1x	<i>1.27 (0.05)</i>	<i>1.44 (0.04)</i>	1.85 (0.04)	1.62 (0.05)	<i>3.30 (0.05)</i>	3.45 (0.06)	3.00 (0.12)
DAZZLE-10x	<i>1.29 (0.01)</i>	1.44 (0.01)	1.89 (0.02)	1.64 (0.01)	<i>3.35 (0.02)</i>	3.51 (0.03)	<i>3.21 (0.11)</i>
Ground Truth: Celltype-specific ChIP-Seq							
# of True Edges	7,050	15,410	1,193	42,795	21,975	14,135	5,180
GENIE3	0.96 (0.00)	<i>1.10 (0.00)</i>	1.01 (0.01)	1.06 (0.00)	0.99 (0.00)	0.98 (0.00)	1.05 (0.00)
GRNBoost2	1.01 (0.00)	1.02 (0.00)	1.00 (0.01)	1.02 (0.00)	0.99 (0.00)	0.99 (0.00)	1.01 (0.00)
DeepSEM-1x	1.06 (0.02)	1.03 (0.01)	1.04 (0.03)	1.03 (0.01)	1.01 (0.00)	1.02 (0.01)	1.06 (0.01)
DeepSEM-10x	1.07 (0.01)	1.03 (0.00)	<i>1.05 (0.01)</i>	1.03 (0.00)	1.01 (0.00)	<i>1.02 (0.00)</i>	1.06 (0.00)
DAZZLE-1x	<i>1.10 (0.02)</i>	1.02 (0.01)	1.03 (0.03)	1.03 (0.01)	1.01 (0.01)	1.01 (0.01)	1.08 (0.01)
DAZZLE-10x	<i>1.10 (0.01)</i>	1.03 (0.00)	1.05 (0.01)	<i>1.03 (0.01)</i>	<i>1.01 (0.00)</i>	1.08 (0.00)	<i>1.08 (0.00)</i>

Metric: AUPRC Ratio; Number of target genes: 1000.

Numbers reported are mean and std. of AUPRC Ratio compared with random guess over 10 runs. Higher ratios indicate better performance. Here, italicized text in dark shading indicate the best algorithms and lightly shaded cells indicate the 2nd best algorithms. In this experiment, we ran each algorithm on all available data with the default settings (for cell-specific ones, we followed the suggestions from BEELINE and used very small L1 regulation on the adjacency matrix). The 10x models ensemble inferred networks from 10 runs and the 1x models are simply single-run models.

10 runs, where higher values represent better performance. In Table 3.1, the evaluation is done separately for the STRING network, the Non-celltype-specific ChIP-Seq network, and the Celltype-specific ChIP-Seq network. Note that for celltype-specific ChIP-Seq, we followed the recommendations of the DeepSEM paper and applied a very small L1 sparsity regulation to the adjacency matrix for better performance.

Among all the evaluations on the STRING and Non-celltype specific benchmarks, the ensemble version of DAZZLE (DAZZLE-10x) has the best performance of all methods in over half the cases, and in all the other cases it ranks either 2nd or 3rd, with a result that is within 6% of the top score. The DAZZLE-10x results also

Table 3.2: Performance on additional benchmarks

	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
# of Genes	1410	1448	1321	1620	1204	1132	692
# of Cells	758	425	383	421	1071	889	847
Metric: Early Precision Ratio; Ground Truth: STRING							
# of True Edges	5,149	9,000	5,898	8,479	1,826	1,311	154
GENIE3	3.66 (0.07)	3.17 (0.03)	2.19 (0.04)	3.48 (0.03)	7.00 (0.06)	8.53 (0.09)	7.34 (0.14)
GRNBoost2	3.46 (0.09)	2.78 (0.04)	2.08 (0.04)	3.69 (0.05)	6.21 (0.06)	7.32 (0.12)	7.12 (0.24)
DeepSEM-1x	4.23 (0.12)	3.12 (0.18)	2.06 (0.09)	3.89 (0.10)	7.36 (0.22)	7.91 (0.33)	7.21 (0.47)
DeepSEM-10x	4.54 (0.11)	<i>3.58 (0.18)</i>	<i>2.20 (0.07)</i>	<i>4.15 (0.07)</i>	7.86 (0.07)	<i>8.96 (0.19)</i>	<i>7.43 (0.20)</i>
DAZZLE-1x	5.12 (0.14)	3.25 (0.08)	2.02 (0.05)	3.90 (0.08)	7.79 (0.17)	8.54 (0.15)	7.37 (0.13)
DAZZLE-10x	<i>5.25 (0.05)</i>	3.33 (0.03)	2.01 (0.03)	3.96 (0.07)	<i>8.06 (0.06)</i>	8.53 (0.08)	7.40 (0.07)
Metric: Early Precision Ratio; Ground Truth: Non-celltype-specific ChIP-Seq							
# of True Edges	4,597	5,335	3,918	8,030	1,960	1,358	317
GENIE3	1.68 (0.03)	1.85 (0.08)	2.93 (0.08)	3.47 (0.05)	4.64 (0.07)	5.86 (0.09)	3.05 (0.09)
GRNBoost2	1.49 (0.05)	1.70 (0.05)	2.58 (0.07)	<i>3.51 (0.05)</i>	4.71 (0.11)	5.03 (0.15)	3.02 (0.11)
DeepSEM-1x	2.20 (0.19)	2.56 (0.12)	3.29 (0.14)	3.31 (0.12)	5.55 (0.39)	5.66 (0.32)	3.14 (0.28)
DeepSEM-10x	2.18 (0.05)	2.67 (0.07)	<i>3.73 (0.08)</i>	3.50 (0.06)	6.04 (0.14)	6.13 (0.22)	<i>3.35 (0.16)</i>
DAZZLE-1x	2.31 (0.12)	2.77 (0.16)	3.51 (0.11)	3.21 (0.14)	6.29 (0.13)	6.11 (0.11)	3.30 (0.17)
DAZZLE-10x	<i>2.40 (0.04)</i>	<i>2.83 (0.05)</i>	3.64 (0.07)	3.26 (0.04)	<i>6.36 (0.06)</i>	<i>6.21 (0.05)</i>	3.34 (0.06)
Metric: Early Precision Ratio; Ground Truth: Celltype-specific ChIP-Seq							
# of True Edges	7,050	15,410	1,193	42,795	21,975	14,135	5,180
GENIE3	0.95 (0.01)	<i>1.12 (0.01)</i>	0.99 (0.06)	<i>1.06 (0.00)</i>	<i>1.02 (0.00)</i>	1.03 (0.09)	<i>1.07 (0.00)</i>
GRNBoost2	0.78 (0.02)	0.27 (0.00)	0.97 (0.08)	0.33 (0.00)	0.21 (0.00)	0.23 (0.01)	0.44 (0.01)
DeepSEM-1x	1.13 (0.04)	1.03 (0.01)	1.11 (0.07)	1.02 (0.01)	1.01 (0.00)	1.02 (0.00)	1.04 (0.01)
DeepSEM-10x	1.15 (0.02)	1.04 (0.00)	1.15 (0.03)	1.02 (0.01)	1.01 (0.00)	<i>1.03 (0.00)</i>	1.04 (0.00)
DAZZLE-1x	1.20 (0.03)	1.03 (0.01)	1.11 (0.06)	1.02 (0.01)	1.01 (0.00)	1.02 (0.00)	1.06 (0.01)
DAZZLE-10x	<i>1.20 (0.01)</i>	1.03 (0.00)	<i>1.17 (0.02)</i>	1.02 (0.00)	1.01 (0.00)	1.03 (0.00)	1.06 (0.00)

Number of target genes: 1000.

Higher ratios indicate better performance. Here, italicized cell in dark shades indicate the best algorithms and lightly shaded cell indicate the 2nd best algorithm.

typically have lower variance compared to the DeepSEM-10x results, illustrating the stability of the DAZZLE model. Further, across nearly all benchmarks, a single run of DAZZLE, DAZZLE-1x, has a higher AUPRC ratio than the comparable single-run DeepSEM-1x. Yet, DAZZLE-1x is typically also more stable. Some DeepSEM-1x results in Table 3.1 have substantial variance, but this unwanted model behavior is not shown by DAZZLE-1x; the reasons for this are explained in Section 3.3.3. Also, as shown in Table 3.2, we also see similar findings when we use Early Precision Ratio (EPR) as an evaluation metric.

For the cell-type specific data, the highest observed AUPRC ratio for any data set or method is 1.10, and the highest observed EPR is 1.20 (just slightly better than random performance), suggesting that for all methods, reproducing these cell-specific networks is nearly impossible. A possible explanation of this effect appears in Figure 3.2 row 3, which shows that the results for this ground-truth network

are highly variable across training iterations. We confirmed that using very low L1 regularization yields better performance on this ground truth network; the reason for this is still unclear.

In summary, under similar training conditions, DAZZLE outperforms DeepSEM and demonstrates more stable performance. We found that while a single run of DeepSEM can produce unstable results, the ensemble strategy proposed in the DeepSEM paper significantly enhances its performance and reduces variance. However, this improvement comes at the cost of a tenfold increase in execution time. For DAZZLE, the ensemble strategy also yields excellent results, making DAZZLE-10x the best performer in our comparison. Note however that the performance of DAZZLE-1x itself approaches that of an ensemble method, in essence because it functions as an ensemble method, as described in the Methods section above. For datasets with a large number of genes and cells, a single pass of DeepSEM or DAZZLE can take several hours to compute, even on modern GPUs. In such cases, a single pass of DAZZLE offers a sufficiently-good solution at a more reasonable cost.

3.3.2 Dropout Augmentation contributes to model robustness

In this section, we discuss the effects of dropout augmentation alone on GRN inference. Figure 3.2 shows the quality of the inferred networks from DeepSEM-1x as a function of the number of training steps. Specifically, it shows that the quality of the inferred networks from DeepSEM tends to drop quickly if the model was over-trained, suggesting that the model may be overfitting to unwanted patterns in the training data. One possible solution is to stop training early, as suggested in the DeepSEM paper, which ends model training at 120 iterations. The tricky part is that the point of peak performance is very difficult to capture and may depend highly on the number of genes and cells in the dataset. In Figure 3.2, the dashed line indicates the recommended point to stop training for DeepSEM. As you can see, in most cases, it is not on the "optimal" point and the prediction performance may quickly downgrade after the model passes the optimal point. A In practice, when ground-truth networks are not available, it is very difficult to identify a good con-

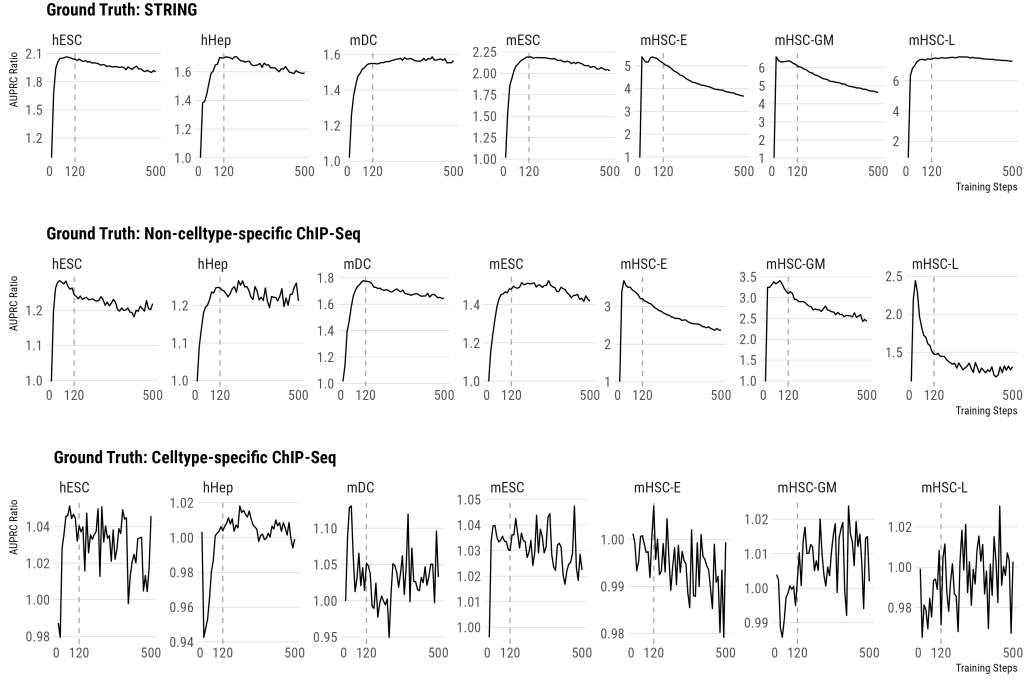


Figure 3.2: Quality of inferred GRN from DeepSEM-1x: AUPRC Ratio as a function of the number of training iterations. Quality may quickly downgrade after the performance peak. Dashed line is the recommended stopping point from DeepSEM. Note that for the celltype-specific data sets, performance is particularly volatile.

vergence point for the model. It would be ideal if inferred GRN accuracy remained robust, or at least if it dropped slowly and consistently after the performance peak, so that the cost of picking a sub-optimal stopping point would be low. We believe that dropout augmentation is potentially an effective solution to this problem.

For fairness, all the comparisons shown in Figure 3.3 were performed on DAZ-ZLE. We varied only the percentage of augmented dropout values while keeping all the other hyper-parameters the same. Without DA (thick black lines, representing a DA probability of 0%), a common pattern is that the AUPRC Ratios drop after the performance peaks, similar to what we observed with the DeepSEM method in Figure 3.2. However, when we train the model with some amount of DA, in most cases the AUPRC Ratios either stay flat or decrease at a slower pace. For some data sets, such as hESC and hHep, DA also improves accuracy significantly. However, we observed that different datasets and ground truth networks seem to require different optimal levels of dropout augmentation. For hESC and hHep on the STRING net-

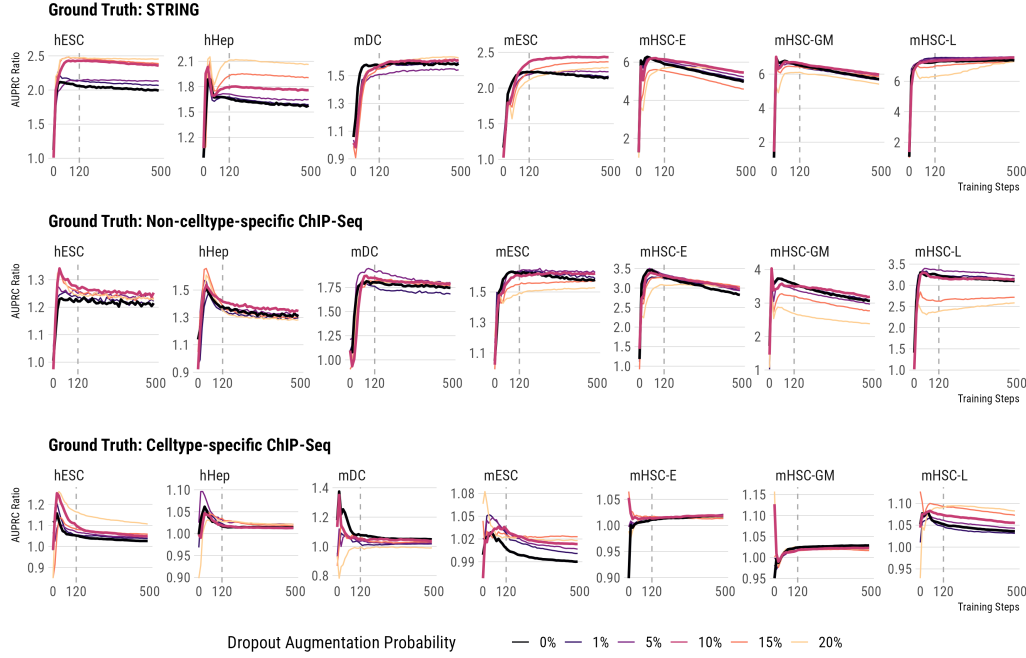


Figure 3.3: Appropriate amount of augmented dropout helps maintain model robustness and may contribute to better performance. Color reflects the probability of dropout augmentation. The two thick lines represent two important conditions, 0% - no augmented dropout, and 10% - the default dropout augmentation level we recommend. Dashed lines shows the default number of training iterations used in DeepSEM and DAZZLE.

work, 20% dropout augmentation yields the best AUPRC scores, but that amount of DA is too high for mHSC-E, mHSC-GM, and mHSC-L since it either slows down the model’s convergence or leads to lower accuracy. After reviewing all cases, we recommend using 10% DA probability as the default for DAZZLE.

3.3.3 Sparsity control strategy improves DAZZLE’s stability

In addition to increased robustness, DAZZLE also generates more stable GRN inference results thanks to its improved sparsity control strategy. As mentioned in Section 3.3.1, results from DeepSEM-1x are not stable. Figure 3.4 examines this issue using 100 runs of DeepSEM-1x and DAZZLE-1x on the hESC dataset evaluated using the STRING network. In Figure 3.4a, we show histograms of the AUPRC ratios for both methods. The accuracy of the 100 runs of DeepSEM-1x can be separated

rated into two groups. As shown in Figure 3.5, other benchmark data sets produced similar results. Further investigation suggested that the L1 regularization on the adjacency matrix might be the main cause of those less ideal results.

Both DAZZLE and DeepSEM are trained with L1 regularization on the adjacency matrix. This regularization term ensures that the model doesn't add unnecessary weights to the adjacency matrix. It also ensures the singularity of the inferred adjacency matrix. Experiments have shown that a large enough coefficient for this sparse loss is required to generate a meaningful adjacency matrix prediction in GRN inference. However, as shown in Figure 3.4b, for DeepSEM-1x, where this L1 sparse loss is introduced at the very beginning of the training process, the optimization of this loss may be prioritized such that the values in the adjacency matrix quickly drop to near zero within the first several training steps. Most runs that do this end up in the low performance group. In DAZZLE, we propose a simple solution to overcome this limitation by delaying the introduction of the L1 sparse loss by a few training steps. As shown in Figure 3.4c, after 5 steps of training without the L1 sparse loss, the values and the gradients on the adjacency matrix parameters are stabilized sufficiently so the introduction of the sparse loss is less likely to destabilize the training process. We also see similar findings on other benchmark datasets; detailed results are included in Supplement Figure S2.

3.3.4 Case study: Using DAZZLE to predict temporal changes in GRNs in mouse microglia

To test the effectiveness of running DAZZLE on actual single-cell data, we applied the method to published data characterizing mouse microglia at different developmental stages (Hammond et al., 2019). For each time point, DAZZLE inferred an adjacency matrix for all the input genes. All edges whose weights have an absolute value over 0.001 are extracted and analyzed.

These GRN inference results confirm that gene regulation is a dynamic process that changes across the lifespan. Figure 3.6a lists the top ten regulated genes at each time point, ranked by the summed edge weights of regulating relationships on

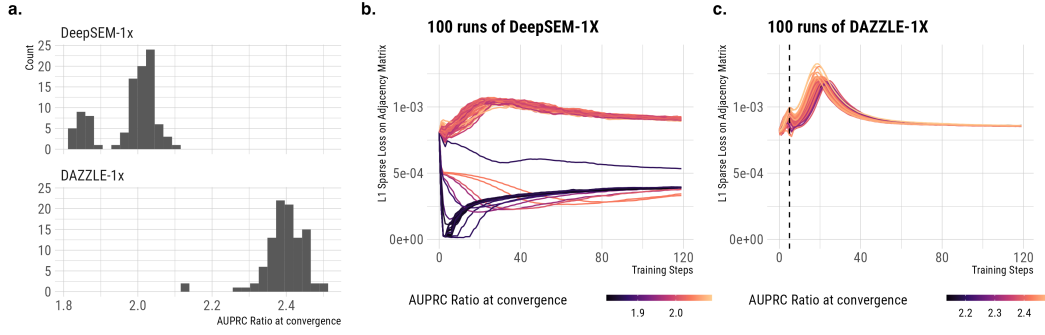


Figure 3.4: Comparison of 100 runs of DeepSEM-1x and DAZZLE-1x on hESC evaluated using the STRING network. In DeepSEM-1x, the prioritization of the L1 sparsity control at early stage is the main cause of unstable GRN inference performances. DAZZLE solved this issue by delaying the introduction of L1 sparse loss by 5 iterations. a. Histogram of AUPRC ratios. b. Sparse loss over time for DeepSEM-1X, colored by AUPRR at convergence. c. Sparse loss over time for DAZZLE-1x, colored by AUPRR at convergence.

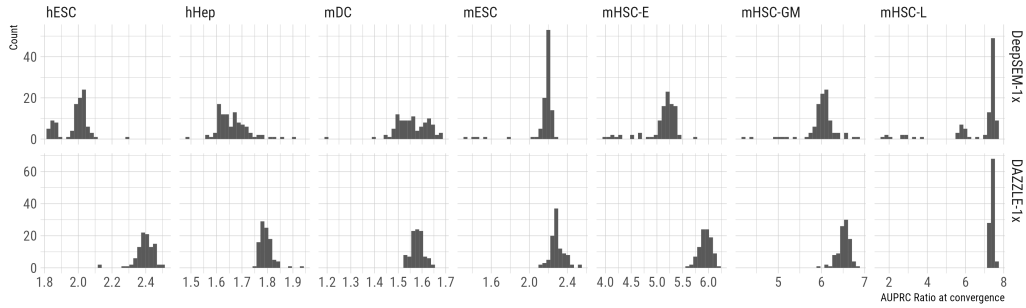


Figure 3.5: Distribution of 100 runs of DAZZLE-1x and DeepSEM-1x on BEELINE evaluated using the STRING network. Results from DAZZLE-1x tend to be more stable than results from DeepSEM-1x.

each gene. At the earliest life stages, most of the top regulated genes are associated with cell proliferation and differentiation. For example, *Tuba1a* (Tubulin alpha 1a) encodes proteins in microtubules, which form the mitotic spindle for cell division and motility structures that move cells to their correct positions. At later ages, however, we see more regulation of immune response genes. Many of these genes, such as *Tmem176B* (transmembrane protein 176B), *H2.D1* (histocompatibility 2, D region locus 1), and *PISD* (phosphatidylserine decarboxylase), encode key proteins, receptors, and enzymes of microglial immune response.

Figure 3.6b shows a closer view of two specific genes of interest. First, since

a. Top regulated genes at different mouse developmental stages

Embryonic Day (E14)	Early Postnatal Day (P4/5)	Late Juvenile Stage (P30)	Adulthood (P100)	Old Age (P540)
Tuba1a	Rsrp1	Serinc3	Serinc3	Lyz2
Malat1	Lpcat2	Tgfbr1	Rsrp1	Hmha1
P2ry12	P2ry12	Srrm2	Tgfbr1	PISD
H3f3b	Ltc4s	Actb	H2.D1	Ly6e
Tubb5	Serinc3	Rbm39	Frmd4a	Rbm39
Apoe	Glul	Apoe	Rbm39	Mbnl1
Ywhae	H3f3b	Frmd4e	Ssh2	H2.D1
Fth1	Asah1	Mbnl1	Ctsh	Sepp1
C1qc	Crybb1	Adap2	Cd14	Gpr34
Trap1a	Malat1	Rrbp1	Tmem176b	Eef1a1

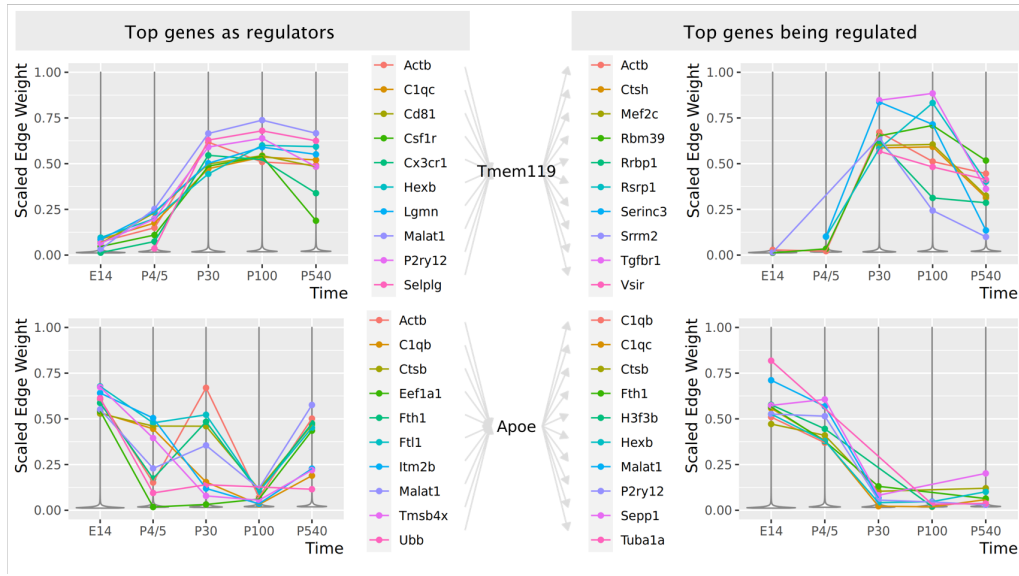
b. Closer look at local networks around specific genes at different developmental stages

Figure 3.6: a. Top 10 regulated genes at each life stage in mouse microglia. b. Predicted local networks around *Tmem119* and *Apoe* in mouse microglia. Here edge weights are min-max scaled at each time point. Top genes are selected according to the maximum weight at all time points.

Tmem119 (transmembrane protein 119) is often used as a biomarker to differentiate microglia from other immune cells in the brain (Bennett et al., 2016), we chose it as the center of the local network to analyze. Figure 3.6b shows that regulation of and by *Tmem119* is only heavily active starting in the late juvenile stage. Among the top 10 predicted regulators, *C1qc*, *Cd81*, *Cx3cr1*, *Hexb*, *Lgmn*, *P2ry12*, and *Selplg* are commonly considered as part of the microglia transcriptional “signature,” as they are generally expressed at low levels in other immune cells (Masuda et al., 2020; Holtman et al., 2015; Boche and Gordon, 2022; Schwabenland et al., 2021; Pettas et al., 2022). In addition, *Csf1r* has been recently identified as a regulator of

pathogenesis for microglia and macrophages (Hagan et al., 2020). It is reasonable to hypothesize that this surrounding local neighborhood includes much of the core functionality of healthy microglia.

Apoe (apolipoprotein E) is another well-studied gene that encodes a protein playing a central role in lipid metabolism, neurobiology, and neurodegenerative diseases. Unlike *Tmem119*, *Apoe* is highly regulated at embryonic day E14. As the mice mature, the relative impact of *Apoe* drops, but it increases again in old age. The list of top predicted links by DAZZLE is consistent with recent studies showing the variety of *Apoe*'s roles in cellular activities. For example, at E14, the top three genes regulating *Apoe* are *Ftl1* (ferritin light polypeptide 1), *Tmsb4x* (thymosin beta 4 X-linked), and *Itm2b* (integral membrane protein 2B). Although the connection between *Apoe* and *Ftl1* is not well established, a recent study (Ma et al., 2021) shows that *Apoe* deficiency leads to increased iron levels. Another study (Kenkhuis et al., 2021) suggests iron loading is a prominent marker of activated microglia in Alzheimer's disease patients. Further, both genes are located on mouse chromosome 7, about 20cM apart. This evidence suggests a plausible regulatory connection between *Ftl1* and *Apoe* and further reflects the important role of iron in early brain development.

Figure 3.6 illustrates DAZZLE's predicted regulation patterns for two typical well expressed genes. However, we have found that DAZZLE's regulatory predictions make sense even for genes whose overall expression levels are low. (See, e.g., comparable images for *Ifit3* (Interferon Induced Protein With Tetratricopeptide Repeats 3) on the project web site.)

Another gene worth mentioning is *Malat1* (Metastasis Associated Lung Adenocarcinoma Transcript 1), which appears in our predicted networks as one of the top regulators for many microglia core genes, including (Figure 3.6) *Tmem119*, *Apoe*, and *H2.D1*. As a long non-coding RNA (lncRNA), *Malat1* has been identified in many pathological processes with immunological components, including several types of cancer (Amodio et al., 2018) and diabetes (Gordon et al., 2018). It has also been identified as a key regulator in the microglial inflammatory response (Zhou

et al., 2018; Cai et al., 2020), but beyond that its function in microglia is mostly unexplored.

Overall, our analysis of this data set confirms that DAZZLE can handle typical real-world single-cell data with more than 10,000 genes and thousands of cells. On examination, the predicted networks appear generally consistent with current research on gene regulation in microglia. Beyond previously identified links, DAZZLE also suggests novel yet plausible links that may be confirmed through future experiments.

3.4 Discussion

In this study, we tackle the dropout problem in real-world single-cell data paradoxically by adding more dropout. To our knowledge, this is the first time a noise-injection approach has been applied for this purpose. Our experiments show that for the GRN inference task, DA improves model performance.

In terms of GRN inference, our proposed method DAZZLE not only stabilizes the output but also produces better predictions. On the BEELINE benchmarks, a single-run of DAZZLE yields comparable results to an ensemble containing of 10 repeated trials with the previously most-accurate method.

Finally, our experiment with the microglia dataset shows that DAZZLE has the capacity to run on large single-cell data with minimal gene filtration. While we do not have ground-truth networks suitable for direct comparison in these cases, the predicted networks are consistent with current understanding and include plausible novel links. These novel links may be good candidates for future investigations of key regulatory relationships.

One major limitation of the current model architecture is that the space complexity of this model also scales quadratically. With 15,000 genes, the model requires 30 Gb GPU memory, which still fits in a single modern GPU. However, for even larger use cases, the method may require multiple GPUs. Another limitation is that the current version is designed to be applied to each individual dataset (time

point or cell cluster). Thus, it does not help us develop a universal understanding of gene interactions. How to relax these restrictions, how to learn these connections in a more efficient way, and how to interpret the inferred networks biologically are important questions to consider in future work.

Chapter 4

Diffusion Probabilistic

Model-Based GRN Inference

4.1 Introduction

The noise injection and prediction idea that we used in DAZZLE further inspired us to further explore the possibility of utilizing noise in network inference. One potential learning framework is diffusion probabilistic models (Ho et al., 2020; Sohl-Dickstein et al., 2015). In Chapter 2, we provided an overview of the structure of diffusion probabilistic models but in short, it is doing something very similar to what we are doing in DAZZLE. There is a non-parameterized forward process, which adds small amounts of noise to the data for each of a number of steps, and a parameterized reverse process, which reconstructs a less-noisy version of the data.

Here, we introduce RegDiffusion, a novel diffusion probabilistic model focusing on the interactions among variables. To our knowledge, this is the first time a diffusion probabilistic model has been applied for this purpose. In contrast with other diffusion models, since the main objective of RegDiffusion is to obtain the adjacency matrix from the trained model, it is typically not necessary to use it as a generative model. At the same time, since RegDiffusion does not use any known/existing edges as training labels, it should be considered an unsupervised algorithm.

In our experiment, we demonstrate that GRNs can be learned effectively using the objective of predicting added noise in a diffusion process. The network architecture of our model is surprisingly simple, yet it outperforms several benchmark GRN inference methods. Overall, RegDiffusion has the following advantages:

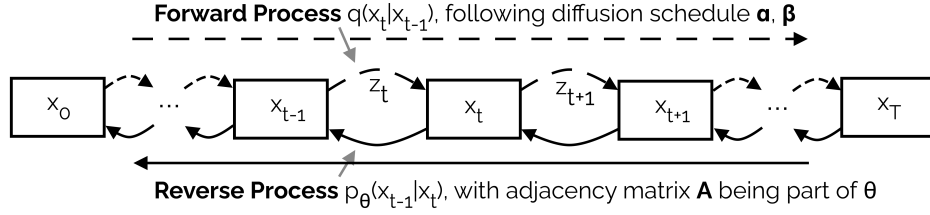
- Compared to previous VAE-based models, the runtime of RegDiffusion improves from $O(m^3n)$ to $O(m^2)$, where m is the number of genes and n is the number of cells. One of the most important theoretical breakthroughs is the utilization of a more restrictive assumption allowing elimination of the costly adjacency matrix inversion step. Inference of networks with more than 15,000 genes now takes less than 5 minutes.
- RegDiffusion combines high-performance GRN inference with high stability. Compared to VAE-based deep learning models, RegDiffusion enforces a trajectory to normality by its diffusion process, which helps stabilize the learning process.
- Although presented as a deep learning method, RegDiffusion can be viewed as a form of Bayesian Network and is based on widely-used assumptions in gene regulatory research (Sanchez-Castillo et al., 2018).
- The success of RegDiffusion also illustrates the potential of applying diffusion models to single-cell and other kinds of noisy tabular data.

4.2 Methods

4.2.1 RegDiffusion

In normal diffusion models, we are creating a model $p_\theta(\mathbf{z}|\mathbf{x}_t, t)$ to estimate the standard Gaussian noise $\mathbf{z}$ used in the forward pass (Eq. 2.3). In computer vision, this challenging problem is usually resolved using image segmentation models, such as U-Net (Ronneberger et al., 2015), to ‘segment’ the noise. Such an approach is not applicable in single cell data, where the data is tabular and the columns are

Diffusion Process in RegDiffusion



Example with 4 genes

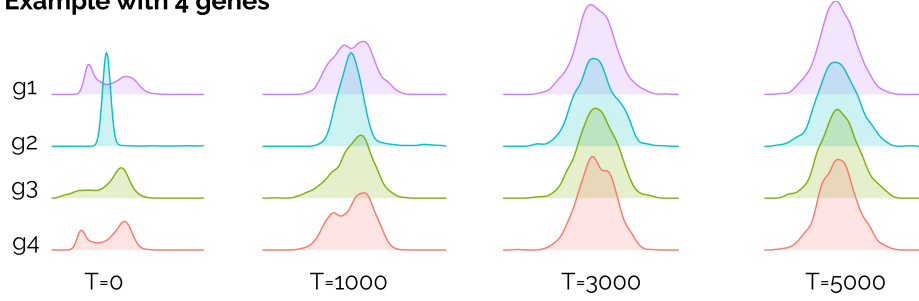


Figure 4.1: The Diffusion Process is an iterative process that consists of both the forward diffusion pass and the reverse pass. The top panel (based on (Ho et al., 2020)) shows the forward diffusion process transforming all data into Gaussian noise in T steps following a diffusion schedule, while the reverse process aims to recover the original distribution from the near-Normal distribution. We are modeling the reverse process with the adjacency matrix being part of the model. The bottom panel shows four representative genes and how their distributions change over time. Each gene begins with a unique distribution over all cells, but by the final time step the distribution is nearly Normal.

unordered genes. Here, to resolve this problem, we propose a new way of calculating the noise using the linear additive assumption in Bayesian Networks.

4.2.1.1 Graph noise estimating model

Our proposed RegDiffusion method relies on the linear additive assumption, which is commonly used in many GRN inference methods, especially in those based on Bayesian Networks (Friedman et al., 2000; Sanchez-Castillo et al., 2018). This assumption requires that the log probability of a particular gene is a linear combination of all of the regulator genes.

$$\mathbf{X} = \mathbf{X}\mathbf{A} \quad (4.1)$$

Note that this assumption is more restrictive than the one used in DeepSEM, which includes an addition term for random noise. The benefit of removing that term will become apparent later.

During the diffusion process, the gene expression matrix $\mathbf{X}$ is under slight Gaussian perturbation. The perturbed $\mathbf{X}\mathbf{A}$ term on the RHS of Equation 4.1 will become a good approximation of the unperturbed $\mathbf{X}$ since the sum of the noise on all neighboring nodes will most likely trend back to zero with a much tighter variance.

$$\mathbf{X} \approx \mathbf{X}_t \mathbf{A}$$

This means that the perturbed gene expression $\mathbf{X}_t$ can be written in the following form:

$$\mathbf{X}_t \approx \mathbf{X}_t \mathbf{A} + \lambda_t \mathbf{Z},$$

where $\mathbf{Z}$ is the Gaussian noise and λ_t is some constant for time step t . This equation can be easily transformed into a conceptual equation to estimate $\mathbf{Z}$:

$$\begin{aligned} \mathbf{Z} &\propto \mathbf{X}_t - \mathbf{X}_t \mathbf{A} \\ &\propto \mathbf{X}_t (\mathbf{I} - \mathbf{A}) \end{aligned} \tag{4.2}$$

Equation 4.2 is exactly what we need to replace U-Net in the reverse modeling part of a diffusion model. This simple form describes a more straightforward network design compared with VAE-based models such as DeepSEM and DAZZLE. Most importantly, it is no longer necessary to perform matrix inversion on the $(\mathbf{I} - \mathbf{A})$ matrix. Matrix inversion, which is often solved by Gaussian elimination, runs in cubic time, and is one of the major bottlenecks in VAE-based models for large numbers of genes. At the same time, there is no need for separate encoders and decoders. All we need to do is to parameterize $\mathbf{X}_t$ and integrate the time step t .

The detailed structure of our proposed network, RegDiffusion, is shown in

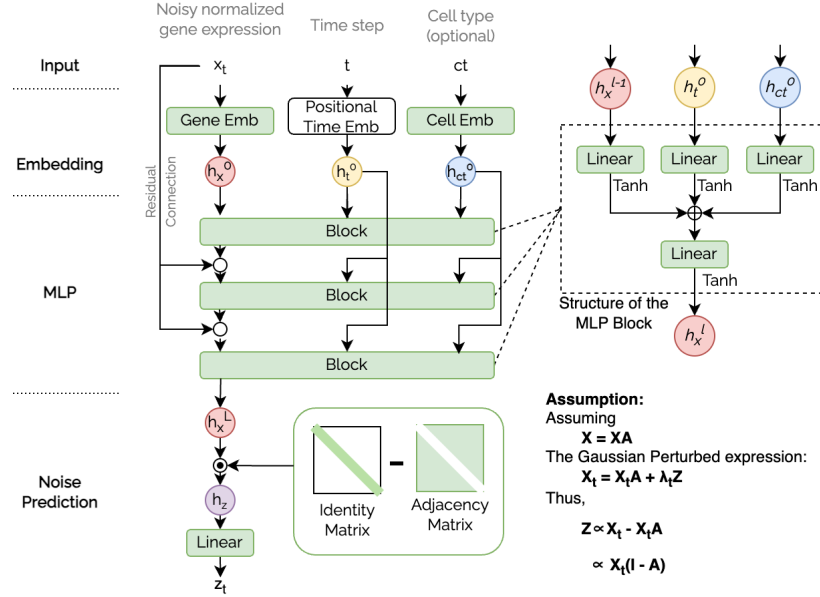


Figure 4.2: Graphic illustration of the RegDiffusion model. Green areas represent trainable parameters within the model. RegDiffusion starts with embedding the model inputs (gene expression x_t , diffusion time step t , and cell type ct) into corresponding feature matrices (h_x^0 , h_t^0 , and h_{ct}^0). These feature vectors are then integrated through 3 layers of MLP training blocks. In each training block, the integrated gene features h_x^{l-1} are mixed with the original embeddings of the time step and the cell type feature. The final integrated feature h_x^L then multiplies with the $(\mathbf{I} - \mathbf{A})$ matrix and the outcome is used to predict the added noise z_t .

Figure 4.2. We explain each section in the following paragraphs.

4.2.1.2 Data Preprocessing and Normalization

RegDiffusion requires a single-cell count table as input. We suggest users first follow common data preprocessing steps to address their specific needs. In experiments described here, we simply transform the data using the $\log_2(\text{count-plus-one})$ transformation and apply a standard quality assurance workflow (removing cells with high mitochondrial content or low/high UMI or gene counts). In terms of gene filtering, we removed genes that are not expressed at all. There is no need to restrict consideration to only the most variable genes, because RegDiffusion with all detected genes runs fairly quickly on modern GPUs.

We first normalize the expression table by cells using min-max normalization to balance inter-cell differences. Then, we z-score normalize the genes.

4.2.1.3 Embedding

As shown in Figure 4.2, our model starts with an embedding process that transforms the input gene expression vector x_t , diffusion time step t , and (optionally designated) cell type ct to corresponding embeddings h_x^0 , h_t^0 , and h_{ct}^0 . For each cell, the gene embedding h_x^0 is a 2D array in the shape of $\mathbb{R}^{m \times d_{gene}}$, where m is the number of genes and d_{gene} is the size of the gene embedding. The first column is always the normalized expression values. The other columns are trainable gene features that are shared across cells. Next, following the standard practice for Diffusion Probabilistic Models, we use a Sinusoidal Positional Embedding to embed the time steps t . This process transforms an ordinal time step to a meaningful vector $h_t^0 \in \mathbb{R}^{d_{time}}$ using sine and cosine transformation. Finally, RegDiffusion works well without additional cell type labels. However, if such information is available in existing data, users have the option to supply it to the model. In this case, the cell type embedding h_{ct}^0 is another 2D array in the shape of $\mathbb{R}^{c \times d_{celltype}}$, where c is the number of cell types and $d_{celltype}$ is the dimension of the cell type embedding. Note that here, all the feature dimensions (d_{gene} , d_{time} , and $d_{celltype}$) are hyperparameters of the model. In the current implementation, we set d_{gene} to be the same as the size of the first block.

4.2.1.4 Feature Learning MLPs

In RegDiffusion, we use a set of multilayer perceptrons (MLPs) to aggregate feature matrices (h_x , h_t , and h_{ct}) into a representation that is ready for noise prediction, as in Equation 4.2. To achieve this goal, we design an MLP building block as shown in the inset in Figure 4.2. This building block consumes the previous gene feature h_x^{l-1} , the original gene feature h_x^0 , the time step embedding h_t^0 , and the cell type embedding h_{ct}^0 . These features are converted into the same dimension using three separate linear models, and the activated values are summed together followed by another linear layer to build the new gene feature h_x^l . After L layers of the learning block, we are able to build a combined feature h_x^L .

4.2.1.5 Adjacency Matrix Initialization and Regularization

The adjacency matrix in RegDiffusion is an $m \times m$ matrix of trainable model parameters, where m is the number of genes. Here we assume the average effect of a random gene to another, or the “regulation norm”, is $\frac{1}{m-1}$. The initialization value of the adjacency matrix is set to five times the regulation norm to help the model explore both potential links and non-links. Here we set the default to 5 because in our experiment, a value of 1 to 10 typically gave us similar results. During the training process, unless otherwise specified, the learning rate on the adjacency matrix is set to one fiftieth of the regulation norm to ensure the learning steps on the adjacency matrix scale with the number of genes. To help the model converge faster, we also employ a soft threshold function (as shown in Equation 4.3) of half the regulation norm on the absolute values of the edge weights, so tiny edge weights are clipped during the training process. At the same time, the diagonal entries of the adjacency matrix are set to zeros to discourage self-regulation. To increase the sparsity of the adjacency matrix, we applied a combination of L1 and L2 losses on the matrix (weights controlled by two hyper parameters). To prevent overfitting and increase model robustness, we also apply dropout to the adjacency matrix; specifically, a proportion of edge weights are randomly set to zeros while the rest are scaled accordingly during training.

$$S_{\tau}(x) = \text{sign}(x) \cdot \max(0, |x| - \tau) \quad (4.3)$$

4.2.1.6 Noise Prediction

Following equation 4.2, with the learned feature matrix h_x coming out of the stacked blocks and the adjacency matrix, we can calculate the feature of the added noise h_z using matrix multiplication. This feature is then converted into the prediction of noise z using a linear layer.

4.2.1.7 Data Loading and Model Convergence

RegDiffusion considers data from each cell as an observation. In reality, the number of cells in different single-cell data varies significantly. For different datasets, if the model sees the entire dataset in every training iteration, it would take different numbers of iterations to converge. To overcome this issue, in each training iteration, we sample a fixed number of cells with replacement. By default, we set this number to 128. If the batch size is too small, the differences among batches may lead to instable model training performances. You can also set this number higher so the model will batch process more sample at a time. However, when the number of genes is large, you may run out of GPU memory. At the same time, since batch size is directly linked to learning rate, if you significantly adjust the batch size, you may need to adjust the defaults for learning rate as well. Overall, this bootstrap resampling strategy also helps make RegDiffusion more robust.

The convergence point of RegDiffusion could be identified using the “amount of change” of the adjacency matrix with the help of the iteration training loss. The “amount of change” score is defined as the average change in value of the adjacency matrix after one iteration of training, in units of the regulation norm ($\frac{1}{m-1}$). Eventually, this score will approach 0, reflecting that at this point in the training process, the optimizer is only making tiny updates on the adjacency matrix, showing that the model has converged.

As shown later in the result section, with a fixed sampling size per iteration, the convergence of RegDiffusion on different datasets follows a similar pattern, regardless of the number of genes and cells. The default number of training iteration is selected based on these observations. When users are using this model, they can also use the reported “amount of change” to help them determine if the model has converged. Note that convergence does not guarantee the correctness of the inferred edges, which requires further downstream assessment and validations.

4.2.1.8 Network Interpretation and Neighbor/Neighborhood Extraction

After convergence, the next task is to interpret the inferred network. However, the difficulty of this task increases quadratically with the number of genes. Here, for the purpose of validating the inferred networks in their biological contexts, we propose visualizing the 2-hop local neighborhoods of specific interesting genes. For a given gene, we extract its top-k neighbors (sorted by the absolute values of their edge weights) as the first hop; then, for each gene in that set, we extract its top-k neighbors and add them to the “second hop” set. Finally, we scan through the neighbors of the second-hop genes, to add any remaining edges to nodes in either the first hop or second hop node sets. Thus, multiple genes that work together as a group are pulled closer in the visualization space, making their biological relationship more apparent.

4.2.2 Experiment Design

Experiments were run on the BEELINE benchmark data sets, where there are multiple, different putative “ground truth” networks to which to compare inferred regulatory edges. In addition, we inferred networks from two published sets of single-cell data in microglia, which can be compared to our knowledge about regulation in these cell types and which demonstrate the method’s feasibility in a more realistic context. Experimental details appear in the following subsections.

4.2.2.1 Data Sets - BEELINE Benchmark

Similar to what we used in the previous chapter, here we also use the BEELINE single-cell benchmarks (Pratapa et al., 2020) for systematic evaluation. For details of these benchmarks, please refer to section 3.2.3.1.

We compared the performance of RegDiffusion on these seven benchmark data sets to that of four other methods: GENIE3, GRNBoost2, DeepSEM, and our previously proposed method DAZZLE. We have already discussed GENIE3 and GRNBoost2 in section 2.2. DeepSEM and DAZZLE are both variational autoen-

coder based neural networks. For GENIE3 and GRNBoost2, we used the implementation from the Python package *arboreto* v0.1.6 (Moerman et al., 2019). For DeepSEM and DAZZLE, we used our implementation from the Python package *GRN-dazzle* v0.0.2 (Zhu and Slonim, 2023). Note that as described by the authors, DeepSEM is an ensemble algorithm that combines results from 10 runs. Because of its speed, we include the single-run version, denoted by DeepSEM*, in our comparisons as well as the ensemble version.

4.2.2.2 Data Sets - Hammond Microglia Dataset

To assess RegDiffusion’s capacity on real-world single-cell data, we tested it on two different mouse brain microglia datasets. The first one is a subset of the Hammond Microglia Dataset we used in Chapter 3. The data was processed in a similar way but this time we only focus on the four healthy male adult mice (P100). The final processed data consists of 8,258 cells and 14,065 genes. Note that this data set potentially includes many different microglia subtypes.

4.2.2.3 Data Sets - Cerebellum Atlas Microglia Dataset

The second real-world microglia single-cell data is derived from a published transcriptomic atlas of mouse cerebellar cortex consisting of 611,034 cells of many types (Kozareva et al., 2021). The authors assigned cells to into known cell types based on specific gene markers in cell clusters. For microglia cells, the marker genes used were *Prkcd* and *C1qa*. From this dataset, we didn’t remove any cells from those designated as microglia, but we applied the same criteria as above to filter genes. We then log-plus-one transformed the count data. The final processed data set consists of 1,296 cells and 15,547 genes.

4.2.2.4 Evaluation Metrics

Following the BEELINE paper, we use Area Under the Precision Recall Curve Ratio (AUPRCR) and the ratio value for Early Precision (EP) as the main evaluation metrics. The details of these two metrics have been described in section 3.2.3.2.

	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
# of Genes	1410	1448	1321	1620	1204	1132	692
# of Cells	758	425	383	421	1071	889	847
# of True Edges	5,149	9,000	5,898	8,479	1,826	1,311	154
GENIE3	1.98 (0.01)	1.86 (0.01)	1.72 (0.01)	2.05 (0.01)	4.19 (0.05)	6.27 (0.04)	7.02 (0.08)
GRNBoost2	1.67 (0.01)	1.50 (0.01)	1.43 (0.01)	1.88 (0.02)	3.65 (0.05)	5.16 (0.11)	7.07 (0.08)
DeepSEM* (single-run)	1.95 (0.08)	1.66 (0.10)	1.57 (0.05)	2.20 (0.03)	5.00 (0.49)	5.83 (0.81)	6.69 (1.81)
DeepSEM (ensemble)	2.11 (0.02)	1.85 (0.02)	<i>1.67 (0.02)</i>	<i>2.31 (0.03)</i>	5.67 (0.13)	<i>6.76 (0.17)</i>	<i>7.50 (0.12)</i>
DAZZLE	2.50 (0.05)	<i>1.87 (0.02)</i>	1.60 (0.02)	2.26 (0.06)	6.02 (0.15)	6.39 (0.16)	7.65 (0.12)
RegDiffusion	<i>2.48 (0.01)</i>	1.99 (0.01)	1.66 (0.01)	2.45 (0.02)	<i>5.96 (0.07)</i>	7.14 (0.07)	7.28 (0.07)

Table 4.1: RegDiffusion outperforms most GRN inference methods on BEELINE benchmark datasets (Metric: AUPRC Ratio; Ground truth: STRING; Number of target genes: 1000). Numbers reported are mean and std. of AUPRC Ratio compared with random guess over 10 runs. Higher ratios indicate better performance. Here, bold **Red** and italicized *orange* text indicate the **best** and *2nd best* algorithms. In this experiment, we ran each algorithm on all available data with the default settings.

	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
# of Genes	1410	1448	1321	1620	1204	1132	692
# of Cells	758	425	383	421	1071	889	847
# of True Edges	5,149	9,000	5,898	8,479	1,826	1,311	154
GENIE3	3.66 (0.07)	3.17 (0.03)	<i>2.19 (0.04)</i>	3.48 (0.03)	7.00 (0.06)	8.53 (0.09)	7.34 (0.14)
GRNBoost2	3.46 (0.09)	2.78 (0.04)	2.08 (0.04)	3.69 (0.05)	6.21 (0.06)	7.32 (0.12)	7.12 (0.24)
DeepSEM*	4.12 (0.17)	3.00 (0.23)	2.13 (0.16)	3.89 (0.13)	7.27 (0.36)	7.79 (0.65)	6.92 (1.11)
DeepSEM	4.55 (0.08)	3.69 (0.05)	2.16 (0.07)	<i>4.12 (0.07)</i>	7.87 (0.17)	<i>8.73 (0.24)</i>	7.48 (0.14)
DAZZLE	<i>5.31 (0.12)</i>	3.50 (0.13)	2.04 (0.06)	3.89 (0.09)	<i>7.99 (0.13)</i>	8.39 (0.12)	7.57 (0.18)
RegDiffusion	5.38 (0.05)	<i>3.51 (0.05)</i>	2.23 (0.04)	4.18 (0.04)	8.41 (0.09)	9.19 (0.07)	<i>7.51 (0.30)</i>

Table 4.2: BEELINE Benchmarking Results (**STRING network**, 1,000 target genes) evaluated on Early Precision Ratio (**EPR**). Results are from the same experiment shown in Table 1 in the main text.

4.3 Experiments

4.3.1 RegDiffusion is Highly Accurate on BEELINE Benchmarks

Table 4.1-4.6 present comparisons of GRN inference methods evaluated on different benchmarking datasets, ground truth networks, and evaluation metrics. Numbers reported are means and standard deviations over 10 runs. The seven data sets differ in terms of the number of cells, genes, and edges in the ground truth networks - details appear in the table header.

The proposed RegDiffusion method generally outperforms most other GRN inference techniques across the different datasets, as indicated by higher AUPRC Ratios or Early Precision Ratios. It achieves top results for the hHep, mESC, and

	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
# of Genes	1410	1448	1321	1620	1204	1132	692
# of Cells	758	425	383	421	1071	889	847
# of True Edges	4,597	5,335	3,918	8,030	1,960	1,358	317
GENIE3	0.97 (0.00)	1.01 (0.01)	1.56 (0.01)	1.65 (0.01)	2.54 (0.02)	3.56 (0.03)	2.78 (0.04)
GRNBoost2	1.01 (0.01)	1.09 (0.01)	1.36 (0.02)	1.49 (0.02)	2.39 (0.04)	2.84 (0.03)	2.45 (0.05)
DeepSEM*	1.20 (0.06)	1.35 (0.07)	1.71 (0.06)	1.57 (0.05)	2.95 (0.19)	3.05 (0.24)	3.06 (0.51)
DeepSEM	1.24 (0.01)	1.41 (0.01)	1.94 (0.03)	1.65 (0.02)	3.18 (0.03)	3.52 (0.14)	3.15 (0.08)
DAZZLE	1.27 (0.01)	1.42 (0.02)	1.80 (0.05)	1.64 (0.03)	3.39 (0.06)	3.52 (0.16)	3.32 (0.11)
RegDiffusion	1.18 (0.01)	1.38 (0.01)	1.82 (0.02)	1.72 (0.01)	3.17 (0.03)	3.69 (0.05)	3.02 (0.08)

Table 4.3: BEELINE Benchmarking Results (**Non-celltype-specific ChIP-Seq** network, 1,000 target genes) evaluated on AUPRC Ratio (**AUPRR**). Results are from a similar experiment to that shown in Table 1 in the main text, but with a different ground truth network.

	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
# of Genes	1410	1448	1321	1620	1204	1132	692
# of Cells	758	425	383	421	1071	889	847
# of True Edges	4,597	5,335	3,918	8,030	1,960	1,358	317
GENIE3	1.68 (0.03)	1.85 (0.08)	2.93 (0.08)	3.47 (0.05)	4.64 (0.07)	5.86 (0.09)	3.05 (0.09)
GRNBoost2	1.49 (0.05)	1.70 (0.05)	2.58 (0.07)	3.51 (0.05)	4.71 (0.11)	5.03 (0.15)	3.02 (0.11)
DeepSEM*	2.15 (0.16)	2.55 (0.12)	3.32 (0.19)	3.31 (0.10)	5.71 (0.14)	5.50 (0.24)	3.31 (0.16)
DeepSEM	2.17 (0.07)	2.68 (0.08)	3.74 (0.06)	3.49 (0.06)	6.01 (0.08)	6.31 (0.26)	3.40 (0.13)
DAZZLE	2.26 (0.07)	2.77 (0.16)	3.53 (0.19)	3.24 (0.11)	6.56 (0.15)	6.26 (0.20)	3.65 (0.18)
RegDiffusion	2.12 (0.06)	2.84 (0.06)	3.61 (0.08)	3.91 (0.05)	6.98 (0.08)	6.94 (0.11)	3.48 (0.13)

Table 4.4: BEELINE Benchmarking Results (**Non-celltype-specific ChIP-Seq** network, 1,000 target genes) evaluated on Early Precision Ratio (**EPR**). Results are from a similar experiment to that shown in Table 1 in the main text, with a different ground truth network.

mHSC-GM data sets, and it is either the second or third best for the others, with performance above 95% of the top score in all cases.

Furthermore, the performance of RegDiffusion is very stable across runs, as indicated by the small standard deviations of AUPRC Ratios in 10 repeated runs. It has even better stability than the ensemble version of DeepSEM and is almost as stable as GENIE3 and GRNBoost2.

4.3.2 RegDiffusion Runs Faster than Competitors

As mentioned in Section 4.2, one of the most important breakthroughs in RegDiffusion is the elimination of the matrix inversion step on the adjacency matrix, which is required by previous VAE-based models. Matrix inversion requires cubic time and is the key computational bottleneck when the number of genes is large. By incorpo-

	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
# of Genes	1410	1448	1321	1620	1204	1132	692
# of Cells	758	425	383	421	1071	889	847
# of True Edges	7,050	15,410	1,193	42,795	21,975	14,135	5,180
GENIE3	0.96 (0.00)	1.10 (0.00)	1.01 (0.01)	1.06 (0.00)	0.99 (0.00)	0.98 (0.00)	1.05 (0.00)
GRNBoost2	1.01 (0.00)	1.02 (0.00)	1.00 (0.01)	1.02 (0.00)	0.99 (0.00)	0.99 (0.00)	1.01 (0.00)
DeepSEM*	1.37 (0.05)	1.13 (0.01)	0.99 (0.07)	1.03 (0.01)	1.00 (0.01)	0.98 (0.01)	1.09 (0.02)
DeepSEM	1.38 (0.01)	1.14 (0.01)	1.02 (0.02)	1.03 (0.00)	1.00 (0.00)	0.98 (0.00)	1.08 (0.01)
DAZZLE	1.30 (0.12)	1.13 (0.03)	1.03 (0.09)	1.02 (0.01)	1.02 (0.01)	0.99 (0.06)	1.10 (0.03)
RegDiffusion	1.37 (0.02)	1.15 (0.01)	1.04 (0.01)	1.05 (0.00)	1.02 (0.00)	0.95 (0.01)	1.08 (0.01)

Table 4.5: BEELINE Benchmarking Results (**Celltype-specific ChIP-Seq** network, 1,000 target genes) evaluated on AUPRC Ratio (**AUPRR**). DeepSEM*, DeepSEM, DAZZLE are all run using a different set of hyperparameters recommended by the DeepSEM paper only for use on this benchmark (L1 Loss = 1% of default; KL divergence = 1% of default). RegDiffusion also uses a different set of hyperparameters (L1 loss = 1% of default; L2 Loss = 1% of default; Adjacency matrix initialization = 10 * regulation norm). Because of the low performance ratios and unique hyperparameters here, plus a lack of understanding for why performance with the original hyperparameters is much worse, we do not recommend evaluating on this set of ground truth.

	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
# of Genes	1410	1448	1321	1620	1204	1132	692
# of Cells	758	425	383	421	1071	889	847
# of True Edges	7,050	15,410	1,193	42,795	21,975	14,135	5,180
GENIE3	0.95 (0.01)	1.12 (0.01)	0.99 (0.06)	1.06 (0.00)	1.02 (0.00)	1.03 (0.09)	1.07 (0.00)
GRNBoost2	0.78 (0.02)	0.27 (0.00)	0.97 (0.08)	0.33 (0.00)	0.21 (0.00)	0.23 (0.01)	0.44 (0.01)
DeepSEM*	1.62 (0.05)	1.12 (0.02)	0.97 (0.19)	1.01 (0.01)	1.05 (0.00)	1.06 (0.01)	1.07 (0.01)
DeepSEM	1.64 (0.01)	1.12 (0.01)	1.02 (0.10)	1.01 (0.00)	1.05 (0.00)	1.07 (0.00)	1.07 (0.00)
DAZZLE	1.51 (0.17)	1.12 (0.03)	1.04 (0.15)	1.02 (0.02)	1.04 (0.01)	1.02 (0.06)	1.08 (0.03)
RegDiffusion	1.65 (0.03)	1.14 (0.01)	1.10 (0.05)	1.04 (0.00)	1.02 (0.01)	1.03 (0.01)	1.05 (0.01)

Table 4.6: BBEELINE Benchmarking Results (**Celltype-specific ChIP-Seq** network, 1,000 target genes) evaluated on Early Precision Ratio (**EPR**). DeepSEM*, DeepSEM, DAZZLE are all run using a different set of hyperparameters recommended by the DeepSEM paper only for use on this benchmark (L1 Loss = 1% of default; KL divergence = 1% of default). RegDiffusion also uses a different set of hyperparameters (L1 loss = 1% of default; L2 Loss = 1% of default; Adjacency matrix initialization = 10 * regulation norm). Because of the low performance ratios and unique hyperparameters here, plus a lack of understanding for why performance with the original hyperparameters is much worse, we do not recommend evaluating on this set of ground truth.

	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L	Hammond	Atlas
# of Genes	1,410	1,448	1,321	1,620	1,204	1,132	692	14,065	15,547
# of Cells	758	425	383	421	1,071	889	847	8,258	1,296
GENIE3	6,462	2,931	2,709	4,173	7,723	5,674	2,493	-	-
GRNBoost2	2,055	992	962	2,171	2,442	1,382	379	-	-
DeepSEM* (single run)	29	15	12	17	30	23	15	4hr 21min	57min
DeepSEM (ensemble)	251	147	116	168	298	229	139	43hr 33min	9hr 34min
DAZZLE	31	16	13	18	34	27	19	4hr 47min	1hr 10min
RegDiffusion	7	8	7	9	6	5	5	3min 28s	4min 10s

Table 4.7: Algorithm execution time (reported in seconds unless noted). Time cost on BEELINE data sets are averages over 10 runs. All code was executed on HPC computing nodes with 4 cores and 12GB memory. DeepSEM, DeepSEM*, DAZZLE, and RegDiffusion are executed with an additional NVIDIA A100 card.

rating bootstrap sampling with a fixed pool size, the running time of RegDiffusion is independent of the number of cells. In section 3.5, we further show that this leads all models to converge in a fixed number of steps.

Compared to previous VAE-based models, the run time of RegDiffusion drops from $O(m^3n)$ to $O(m^2)$. A comparison of algorithm execution time is provided in Table 4.7. Across all benchmarks, RegDiffusion runs in a fraction of the time needed by any previous algorithm. This speedup has greater impact on larger data sets. For the two real-world single-cell data sets, each with tens of thousands of genes and thousands of cells, RegDiffusion takes only minutes, while previous VAE-based methods require hours of computation.

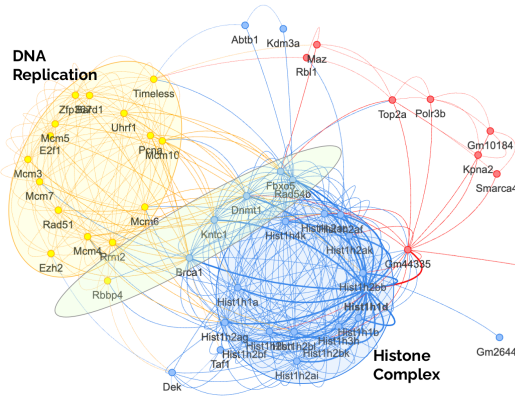
4.3.3 An Example of Interpreting the Inferred Networks

Here we demonstrate how to interpret the inferred network with an example: the network inferred from BEELINE’s mESC data (mouse embryonic stem cells) using the method described in section 2.3.8. For the choice of target gene, we picked *Hist1h1d* (or *H1.3*), a histone encoding transcription factor based on RegNetwork (Liu et al., 2015), as the target gene simply because it has the highest edge weight in the inferred adjacency matrix. Substantial expression and regulation of histone proteins in embryonic stem cells is unsurprising (Luger et al., 1997).

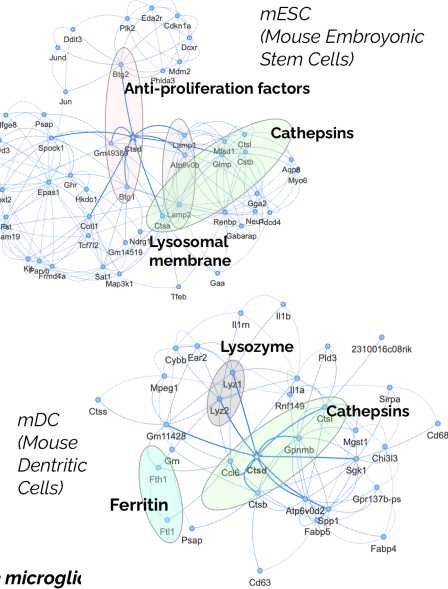
In Figure 4.3a, the inferred 2-hop neighborhood around *Hist1h1d* consists largely of two highly interconnected regions (“hairballs”, colored blue and yellow)

a. Inferred regulatory neighborhood around top regulator in mouse embryonic stem cells (mESC) (k=40) reveals connections between histones and DNA replication.

- This is 2-hop neighborhood centered around *Histh1d*.
- Direct links to *Histh1d* are marked with thick edges.
- Functional groups are marked with ovals.
- Edge directions are ignored for visual simplicity.



b. Inferred regulatory neighborhoods around cathepsin D gene (k=15) are cell-type specific.



c. Inferred regulatory neighborhoods around *ApoE* (k=20) in mouse microglia

Ovals represent functional regions consistent with literature

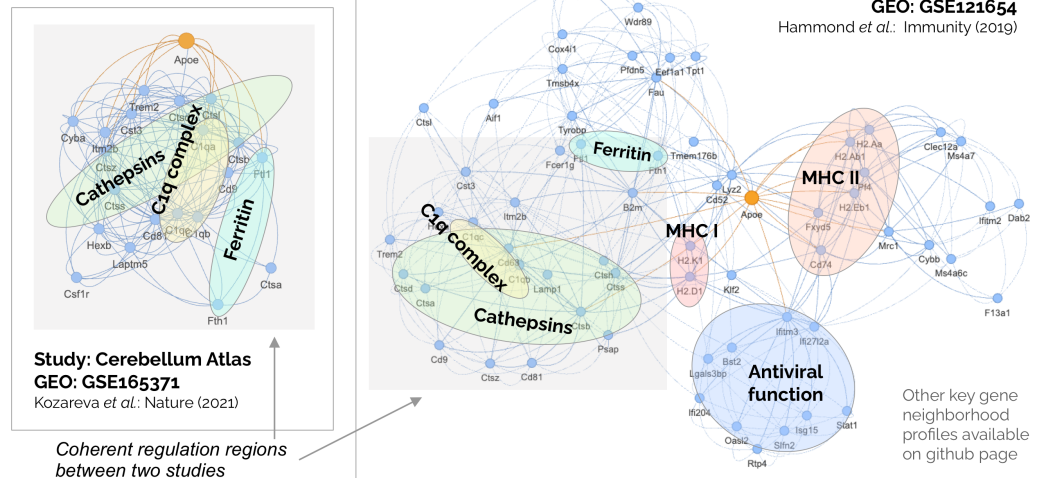


Figure 4.3: Local regulatory neighborhood analysis shows that the inferred regulatory networks generated by RegDiffusion are biologically interpretable. All the networks visualized here are 2-hop neighborhood around target genes at indicated top-k levels.

with a layer of connecting genes. The blue histone complex includes 15 histone genes, comprising the entire set of histone genes available in the mESC dataset. RegDiffusion effectively retrieved all of them as direct neighbors using only expression data. The yellow cluster, which includes 5 minichromosome maintenance protein complex (MCM) genes among others, is enriched for links to DNA replication. Separating

the blue and yellow gene clusters is a layer of connecting genes, including *Rbbp4*, *Brca1*, *Kntc1*, *Dnmt1*, *Fbxo5*, and *Rad54b*. Among them, *Rbbp4* is a histone binding protein (Balboula et al., 2015). *Brca1* is a well-studied tumor suppressor gene that, in conjunction with histone h1, plays a crucial role in DNA repair (Ozgencil et al., 2023). The connection between *Dnmt1* and histone deacetylase has also been reported (Fuks et al., 2000). *Rad54b* deficiency has been reported to be associated with chromosome breakage (Russo et al., 2018).

4.3.4 Inferred regulatory neighborhoods are cell-type specific

As mentioned in Section 2.4, we are suspicious of the cell-specific ChIPSeq ground truth from BEELINE for a number of reasons. First, all of the unsupervised GRN inference methods we tested report performance near that of random guessing when compared to the cell-type specific data for the corresponding network. Second, although DeepSEM reports slightly better results than we found with most other methods (though still effectively no better than random guessing), that performance required a different set of hyperparameters than was otherwise suggested, one that minimizes the regularization on the adjacency matrix and minimizes the regulations of the KL divergence. The underlying cause of this pattern is still unclear, so we do not feel comfortable with the comparisons on these networks. Nonetheless, the comparable performance of RegDiffusion (under minimal adjacency matrix regularization) on this set of ground truth is reported in Supplement Table 4 and 5, to be interpreted with these caveats in mind.

To further address the challenge of cell-type-specific analysis, we compared the inferred regulatory neighborhoods around genes that are expressed across various cell types. For example, we noted that cathepsin genes are expressed in multiple datasets including mESC (mouse embryonic stem cells), mDC (mouse dendritic cells) from BEELINE, and both microglia datasets. As the most abundant lysosomal proteases, cathepsins play an important role in intracellular protein degradation and immune response (Yadati et al., 2020). As shown in Figure 3b, in the inferred networks in mESC, the top regulators of *Ctsd* include B Cell Translocation Gene

(*Btg*) 1 and 2. *Btg1* and *Btg2* are anti-proliferation factors that regulate cell growth and differentiation. It is not surprising to see they trigger the expression of cathepsin to complete their function. In the inferred networks in mDC, one of the biggest changes is the disappearance of *Lamp1* and *Lamp2* from the neighborhood of *Ctsd* even though these two genes are expressed and measured in the mDC data. One possible explanation is that Lamp proteins in dendritic cells, or DC-LAMP, are biochemically different from the other *Lamp* proteins (Arruda et al., 2006; de Saint-Vis et al., 1998). Studies have shown that they are immediately fused into MHC II compartment and that this is a very cell-type-specific process for dendritic cells. In mouse microglia cells, as shown in Figure 3c, cathepsin genes (light green ovals) interact with many microglia homeostatic genes as part of the immune response. The RegDiffusion networks shown are consistent with these cell-specific changes.

Overall, the specificity of the inferred GRNs depends on the cell-type and cell state of the input data. When the input data has many cell types, the inferred GRN could be considered as overlaying many cell-specific GRNs but it also provides an overview of the cellular processes. With cell-specific data, the inferred GRNs might also be more specific, and there might be fewer pathways going through hub genes. We recommend that users design their project workflows based on their specific research questions.

4.3.5 Characteristics of RegDiffusion

The speedup obtained by RegDiffusion, in addition to relying on the avoidance of the matrix inversion step, comes from using bootstrapping to ensure convergence in time independent of the number of cells. In Figure 4a, by inspecting the amount of change in the adjacency matrix per iteration in the unit of the regulation norm, we show that across all of our experiments, whether on benchmark or real data sets, the training processes follow a similar pattern and the models converge after a fixed number of steps.

Since RegDiffusion is trained only on the objective of predicting added noise, the ground truth GRN network never leaks into the training process. It works in

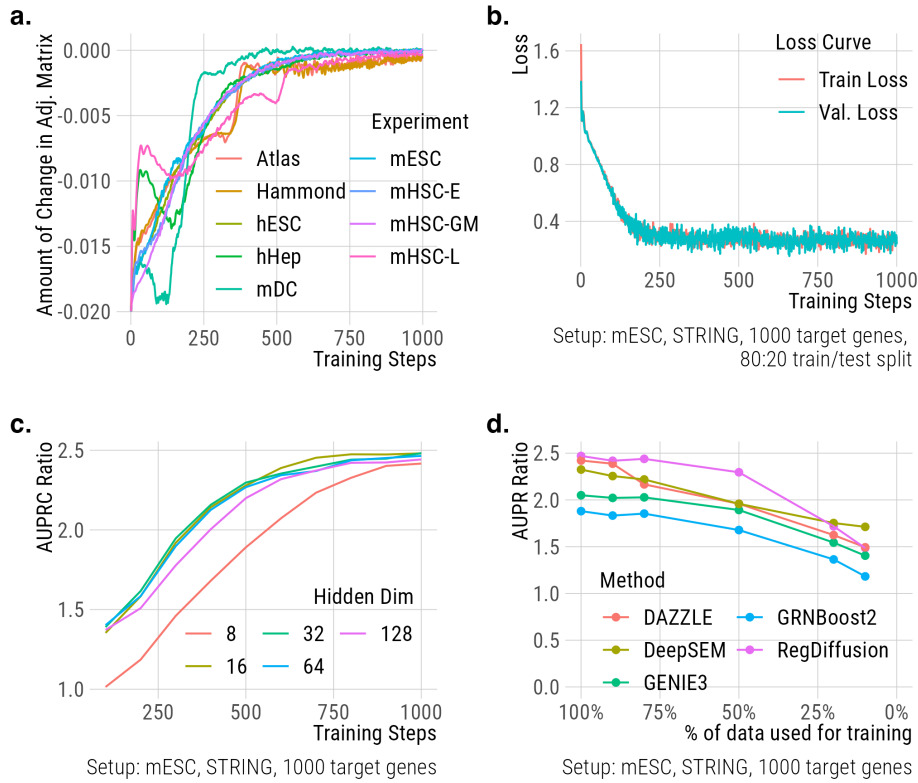


Figure 4.4: Properties of RegDiffusion. **a.** By measuring the amount of change in the adjacency matrix, we show RegDiffusion converges after a fixed number of steps in all experiments; **b.** Loss curves on train/test splits (4:1) show that the model was not overfitted. Also, it's expected to see the training loss oscillating even after the model converges; **c.** Model performance is rather stable regardless of the hidden dimensions in the neural networks; **d.** Downsampling shows that RegDiffusion performs better than bulk RNAseq methods even when the sample size is limited.

an unsupervised manner in the same way as GENIE3 or GRNBoost2. To ensure there was no overfitting, we further investigated the training behavior by applying a train/test split on cells. Figure 4b illustrates the training losses of a typical training process on the mESC data set when we applied a 4:1 train/test split. As shown in the figure, the loss curves for the train and test splits are never separated too much, showing that the model was not overfitted. Also, note that since the diffusion steps are stochastic and it is usually harder to learn the fine details, it is expected to see training loss oscillating even after the model converges.

In terms of the number of parameters used in the model, we tested a number

of choices for the dimension of the hidden layer in the MLP building block; results are shown in Figure 4c. For all dimensions greater than 8, the performance of the model was rather stable. Since we would like to reduce the number of parameters in the model as much as possible, we set the default model dimension to 16.

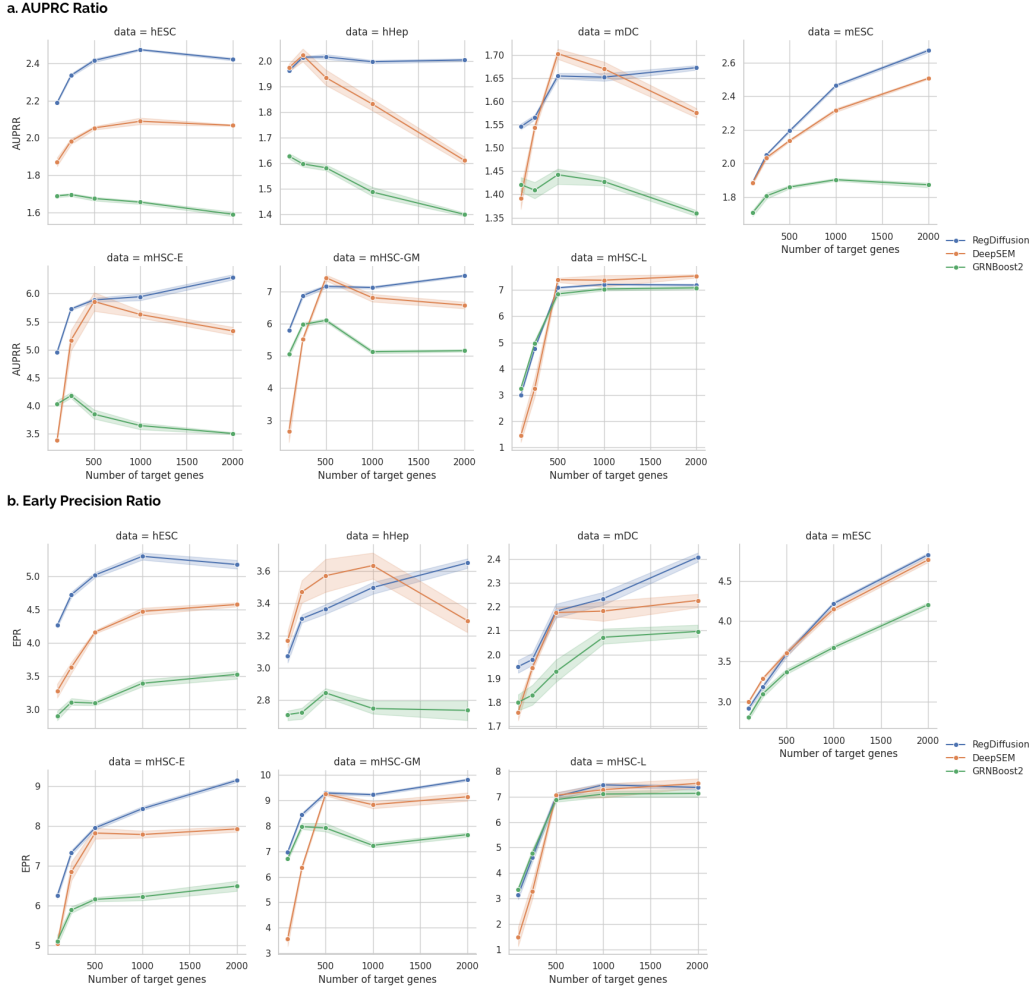


Figure 4.5: Comparison of Inference Performance (a. AUPRC Ratio; b. Early Precision Ratio) with Limited or Abundant data. Ground truth network is STRING. Line and bands represent mean and standard deviations of 10 runs. Each run of DeepSEM ensembles 10 single runs.

To assess RegDiffusion’s performance on networks with different sizes, we varied the number of target genes in BEELINE and assessed model performance for 100, 250, 500, 1000, and 2000 target genes. The results are shown in Figure 4.5. Overall, RegDiffusion appears to maintain a more stable performance across

different numbers of target genes, as compared to DeepSEM. In most cases, training with more genes is helpful for RegDiffusion but it hurts DeepSEM’s performance in many cases. When the number of target genes is very limited (for example, 100), the performance of RegDiffusion seems to be slightly downgraded but is still better than the performance of DeepSEM in the same situation.

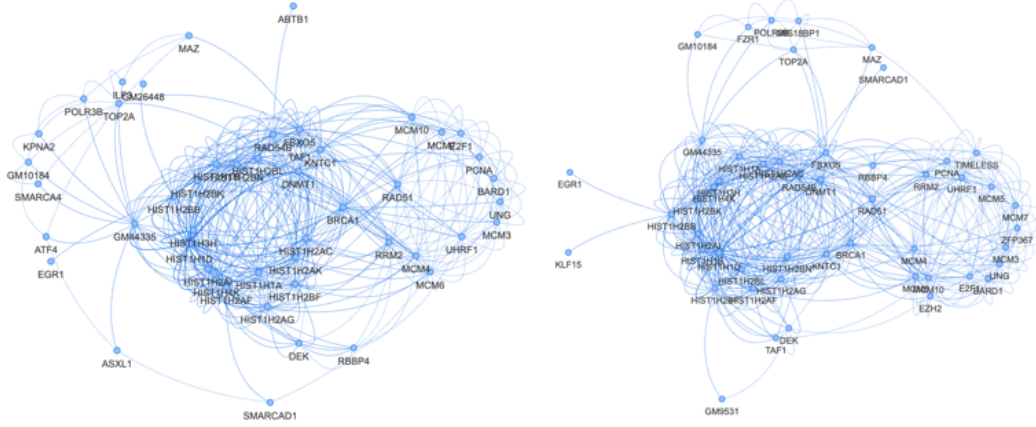


Figure 4.6: Inferred Local Neighborhood around Hist1h1d when inferred with two different batches of 90% sampled cells. These two neighborhoods are very similar with only some small variances (84-86% overlapping nodes and 65-71% overlapping edges).

Finally, we investigated the performance of RegDiffusion with limited numbers of cells. We gradually down-sampled the training data and then compared the AUPRC Ratios of results from RegDiffusion, GENIE3, and GRNBoost2. Figure 4.4d shows that even with 10% of the mESC data, around 42 cells, RegDiffusion still works thanks to the bootstrap resampling methods. It also outperforms tree-based methods, which are usually considered to be robust, but it seems to have lower performance compared with DeepSEM in this case. In practice, some cell subtypes may only have a few sampled cells. This analysis shows RegDiffusion can be effective even in such cases. In Figure 4.6, we further include a visual comparison of the inferred local neighborhood from two different random batches of 90% cell samples. The results are very similar.

4.3.6 Regulatory Networks in Microglia

In both large microglia experiments, we systematically evaluated performance using both the full STRING v11 network and the combined non-cell-type-specific network provided in BEELINE as ground truth. In both experiments, RegDiffusion yields better AUPRC Ratio and Early Precision Ratio compared with DeepSEM according to both benchmarks. For example, in the Hammond experiment evaluated on the STRING network, the inferred network from RegDiffusion has an AUPRC ratio of 1.22 and Early Precision Ratio of 2.23 while the network from DeepSEM reports an AUPRC ratio of 1.17 and Early Precision Ratio of 1.68. The full results are provided in Supplement Table 4.8.

	STRING				Non-cell-type-specific ChIP-Seq			
	Hammond		Atlas		Hammond		Atlas	
	AUPRR	EPR	AUPRR	EPR	AUPRR	EPR	AUPRR	EPR
DeepSEM	1.17	1.68	1.11	1.39	1.06	1.53	1.01	1.09
RegDiffusion	1.22	2.23	1.12	1.51	1.08	1.74	1.04	1.64

Table 4.8: RegDiffusion outperforms DeepSEM in two large microglia experiments on all metrics. Both of the ground truth networks (STRING and Non-cell-type-specific ChIP-Seq) are the full networks from BEELINE.

In addition to benchmark testing, we further examined the biological interpretability of the inferred GRNs in mouse microglia using data from the two complementary studies described in Section 4.2.2. For each study, we inferred a GRN and analyzed the local regulation neighborhood around particular genes. For this discussion, we focus on *ApoE* (apolipoprotein E) and visualize the gene’s RegDiffusion-inferred network neighborhoods from both studies’ data in Figure 4.3c. *ApoE* is a well studied lipid regulator that plays a role in several diseases and disorders, but it is probably best known for the E4 allele’s association with Alzheimer’s Disease (AD) (Fernández-Calle et al., 2022).

The ApoE neighborhood inferred from the Hammond dataset is essentially a superset of the neighborhood inferred from the Cerebellum Atlas data. This is possibly because the microglia from the Atlas dataset were selected from clusters

marked by expression of marker genes *C1qa* and *Prkcd*. Some microglial subtypes may have been filtered out in this process, or there might have been fewer microglial cells because of the focus on cerebellum and the large number of other cell types considered in the atlas.

Nonetheless, many key relationships are shared across the data sets in Figure 4.3c, and many inferred links have previously been experimentally confirmed. For example, the regulatory relationship between *ApoE* and MHC (major histocompatibility complex) type I genes, such as *H2.K1* and *H2.D1* has been recently confirmed in (Zalocusky et al., 2021) via an ApoE perturbation experiment. Activated Response Microglia (ARMs), a microglia subtype discovered in (Frigerio et al., 2019), have been found to have overexpressed MHC type II genes, including *H2-Aa*, *H2-Ab1*, and *Cd74*. RegDiffusion even further identified *H2.Eb1* as another potential biomarker for this cell type. On the right side of the MHC II complex in Figure 4.3c, we have another pair of genes *Mrc1* and *Ms4a7*. These two genes were identified in (Hammond et al., 2019) as biomarkers of a unique microglial state during the embryonic phase. While the data we analyzed is from adult mice, the topological position of these two groups suggests a possible relationship between the ARMs cell type and that embryonic cell state. We further identified connections between *ApoE* and a set of antiviral response genes, including *Ifitm3*, which connects directly to *ApoE* based on our model. *Ifitm3* has previously been identified as a γ -secretase modulatory protein that increases the production of amyloid- β in AD ((Hur et al., 2020)).

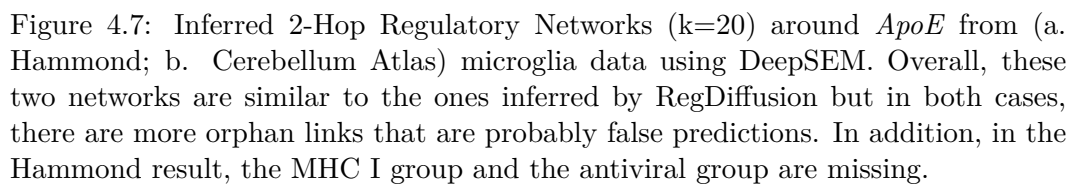
If we compare the inferred *ApoE* neighborhoods between the two studies, we can identify coherent shared patterns. These recapitulate previously identified links among *ApoE* and several functional gene groups, including the *C1q* (Complement component 1q) complex (*C1qa*, *C1qb*, and *C1qc* (Habenicht et al., 2022)), Cathepsins (such as *Ctsd*, *Ctsb*, and *Ctss* (Zhou et al., 2006), (Samokhin et al., 2010), (Li et al., 2024)), Ferritins (*Ftl1* and *fth1* (Ayton et al., 2015)), and several microglia homeostatic genes (such as *Trem2* (Wolfe et al., 2018), *Itm2b* (Biundo et al., 2015), and *Hexb*. While we didn't find prior evidence explicitly showing ApoE regulating

β -hexosaminidase (*Hexa*, *Hexb*), there is evidence that they are both implicated in, and have correlated expression in, AD pathology (Tiribuzi et al., 2011; Sierksma et al., 2020), suggesting such a relationship is plausible. Further, β -hexosaminidase plays a role in lysosomal function and autophagy. Our recent work examining fetal microglia in the setting of maternal immune activation (Batorsky et al., 2024) shows that the CLEAR (Coordinated Lysosomal Expression and Regulation) pathway, particularly *Hexa*, *Hexb*, and several cathepsins, is differentially regulated in microglia exposed to inflammation. From the consistency and plausibility of these networks, we conclude that the results from running RegDiffusion on current single-cell data sets are reasonable and consistent.

For comparison purposes, the local networks inferred by DeepSEM are provided in Figure 4.7. Overall, DeepSEM yields similar findings on the Hammond dataset. However, the MHC I group and the antiviral group are missing. There are more microglia homeostatic genes but there are also more “orphan links”, where one of the nodes is not connected with any other nodes in the graph. Since these links are not connected with others, they seem to more likely to be false positive predictions. For the Atlas dataset, DeepSEM generates a lot of orphan links but it still captures the *C1q* complex and the cathepsins among the noise. In addition to the local neighborhood around *ApoE*, we also provide visualizations for 10 other microglia signature genes on our project website.

4.4 Discussion and Conclusion

We have presented RegDiffusion, a novel diffusion probabilistic model designed to infer GRNs from single-cell RNAseq data. RegDiffusion simulates the information diffusion process by iteratively adding Gaussian noise following a fixed schedule. Based on the benchmarking results, compared to other top methods, RegDiffusion demonstrates consistent superiority in GRN inference capacity while remain stable across runs. It also achieves a significant runtime reduction over comparable methods. Even with limited data, RegDiffusion still outperforms competitors. At the



same time, by visualizing a local gene neighborhood in the inferred GRNs, we illustrate that GRNs inferred by RegDiffusion are consistent with current knowledge of

molecular function.

Following the idea of Dropout Augmentation (Zhu and Slonim, 2023), which simulates the dropout phenomenon in single cell data by inserting zero values into expression data using Bernoulli sampling, RegDiffusion is our second attempt at leveraging noise injection in single cell analysis. The success of RegDiffusion in GRN inference also suggests several potential future research paths. First, it would be very interesting to explore the effects of RegDiffusion and its learned embeddings on other single-cell analysis tasks. Another important future research direction is understanding the inferred networks. Our proposed visualization method works well for inspecting individual genes, but it would be nice to investigate ways to understand the graphs more systematically. Finally, it would be helpful to find a solution to learn detailed features in single-cell data given the existence of background noise. In our experiments, we observed that the models are good at learning crude features (at high diffusion steps) but still have some difficulty learning fine features at low diffusion steps, a phenomenon that has been observed many times in other diffusion models.

Diffusion models in general have sparked a lot of research interest in recent years thanks to their huge success in computer vision. Together with a few recent pre-prints in 2023 (Tang et al., 2023), RegDiffusion presents as one of the initial attempts to solve the problems in single-cell data with diffusion models and is the first one that considers gene regulatory relationships. We anticipate that RegDiffusion may not only offer a reliable and convenient tool for GRN inference, but may also advance our understanding of regulatory genomics and single-cell analysis.

Chapter 5

Denoising Diffusion Causal Discovery

5.1 Introduction

Learning a network structure that represents underlying causal dependencies between variables in observational data has long been a crucial goal. It plays an important role in multiple disciplines, including genetics, epidemiology, and economics (Slonim, 2002; Pearl, 2009; Uhler et al., 2013; Spirtes and Zhang, 2016; Pingault et al., 2018; Glymour et al., 2019). Such networks, where nodes represent feature variables and edges capture potential relationships, are often represented as Directed Acyclic Graphs (DAGs), which allow no cycles.

There has been a great deal of prior work developing methods to address this problem. Score-based approaches, such as PC (Spirtes and Glymour, 1991), GES (Chickering, 2002), and LiNGAM (Shimizu et al., 2012), often optimize the network structure using an objective function under the restriction of some combinatorial constraint to describe the acyclicity of the graph. However, they scale poorly as the size of data increases; in fact, the learning inference problem has been proven to be NP-hard (Chickering et al., 2004). To address this challenge, Zheng et al. (2018) proposed a method called NOTEARS that introduces a continuous

acyclicity score, which can be solved by a regular numerical optimizer.

Many others have since extended this formulation, focusing on scalability (Yu et al., 2019; Lee et al., 2019; Ng et al., 2020; Yu et al., 2021), convexity (Bello et al., 2022; Deng et al., 2023; Ng et al., 2024), sparsity control (Wei et al., 2020; Ng et al., 2020), and nonlinearity (Yu et al., 2019; Ng et al., 2019; Zheng et al., 2020; Yang et al., 2021; Shen et al., 2022; Ng et al., 2022; Kalainathan et al., 2022). Its variations also have been applied to a wide range of settings, such as time series (Pamfil et al., 2020; Sun et al., 2021; Shang et al., 2021) and gene networks (Shu et al., 2021; Agamah et al., 2022; Zhu and Slonim, 2024). There are also analyses and discussions of the application of such models to datasets with unequal variances and different data types (Reisach et al., 2021; Kaiser and Sipos, 2021; Ng et al., 2024).

In this paper, we propose a set of Denoising Diffusion Causal Discovery (DDCD) models that offer significant improvements on scalability and the models’ ability to capture nonlinear transformations. DDCD was inspired by the designs of Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020) and Latent Diffusion Models (LDMs) (Rombach et al., 2022). In DDCD, we introduce noise progressively into the data during a forward diffusion process and then reverse the process by predicting the added noise under the constraint of the learned adjacency matrix. By replacing the least squares loss in NOTEARS with a denoising objective, DDCD allows the model to explore a broader range of noise, which enhances its ability to capture both linear and nonlinear relationships. We demonstrate the effectiveness of the proposed method on both synthetic data and large scale real-world datasets. In summary, our contributions are as follows:

- We prove the validity of the denoising objective, which paves the way to using diffusion models for causal structural learning.
- We push the boundary of structural learning on nonlinear data by showing that the nonlinear transformation function can be approximated together with the adjacency matrix.

- We introduce a k-hop DAG constraint, which approximates acyclicity within a k-hop neighborhood. It is a relaxation of the acyclicity constraint in NOTEARS and has improved complexity. We also applied gradient clipping to avoid gradient explosion when the graph is large. We discuss when acyclicity is helpful and when it may not be.
- We propose a fixed-size bootstrap sampling technique so the learning process proceeds similarly for data sets of different sizes.

5.2 Methods

Inspired by the denoising diffusion framework in DDPMs, here we propose an alternative training objective to learn the adjacency matrix $\mathbf{W}$ for a SEM. In this proposal, we will augment each sample in the same way as the forward diffusion process in Equation 2.1. Then, we will optimize a reverse model with the parameterized $\mathbf{W}$ to predict the added noise under the constraint of $\mathbf{W}$. We start with the assumptions for linear SEMs and then move to nonlinear cases.

5.2.1 Denoising Diffusion Models for Linear SEMs

As a quick recap, Denoising Diffusion Probabilistic Models (DDPMs) are a class of generative model that shows strong performance in modeling complex data distributions (Ho et al., 2020). A typical DDPM starts from a non-parameterized forward diffusion process. Given an unperturbed input $\mathbf{x}_0$, the forward process aims to generate a series of noisy samples $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_T$ over T steps, where $\mathbf{x}_T$ usually stands for pure noise. In each step, a small amount Gaussian noise is gradually introduced following a diffusion schedule β as shown in Equation 5.1.

$$\mathbf{x}_t = \sqrt{1 - \beta_t} \mathbf{x}_{t-1} + \sqrt{\beta_t} \mathbf{z}_{t-1} \quad (5.1)$$

Here, $\mathbf{z}_{t-1} \sim \mathcal{N}(0, \mathbf{I})$ so Equation 5.1 is essentially trying to reduce the means to $\mathbf{0}$ while increasing the variances to $\mathbf{I}$. With the reparameterization trick,

Equation 5.1 can be rewritten into Equation 5.2 as follows,

$$\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \mathbf{z}, \quad (5.2)$$

where $\alpha_t = \prod_{i=0}^t (1 - \beta_i)$ and $\mathbf{z} \sim \mathcal{N}(0, 1)$. In this way, the noisy data x_t can be generated in one step.

The actual modeling piece of DDPM is the reverse model, which is trained to predict the added noise $\mathbf{z}$ and to denoise the data. The choice of model for the reverse process depends on the input data. Recent studies have suggested similarity between diffusion models and a generalized form of variational autoencoder (VAE) (Kingma and Welling, 2013) with infinite latent spaces (Luo, 2022).

For linear SEMs, the reverse model is trying to minimize the following denoising objective:

$$\min_{\mathbf{W}} \frac{1}{2n} \|(\mathbf{X}_{\mathbf{t}} - \mathbf{X}_{\mathbf{t}} \mathbf{W}) - \text{diag}(\sqrt{1 - \alpha_{\mathbf{t}}}) \mathbf{Z} (\mathbf{I} - \mathbf{W})\|_F^2 + \lambda_1 \|\mathbf{W}\|_1 + \lambda_2 \|\mathbf{W}\|_2, \quad (5.3)$$

where $\mathbf{t}$ is a vector of diffusion time steps for all samples in $\mathbf{X}$, $\mathbf{X}_{\mathbf{t}}$ is the perturbed observational data $\mathbf{X}$, and $\text{diag}(\mathbf{v})$ is the diagonal operator that converts vector $\mathbf{v}$ to a diagonal matrix.

Theorem 5.2.1.1 *For linear SEMs, the objective functions in Equation 5.3 and Equation 2.6 are equivalent.*

Proof Consider the case when each sample in $\mathbf{X}$ is perturbed using the forward diffusion process in 5.2. The perturbed observational data could be written as Equation 5.4. Here we use $\mathbf{t}$ to denote the different diffusion time steps for all samples in $\mathbf{X}$ and the $\text{diag}()$ operator scales each row of $\mathbf{X}_0$ and $\mathbf{Z}$ accordingly based on the diffusion schedule.

$$\mathbf{X}_{\mathbf{t}} = \text{diag}(\sqrt{\alpha_{\mathbf{t}}}) \mathbf{X}_0 + \text{diag}(\sqrt{1 - \alpha_{\mathbf{t}}}) \mathbf{Z}, \quad (5.4)$$

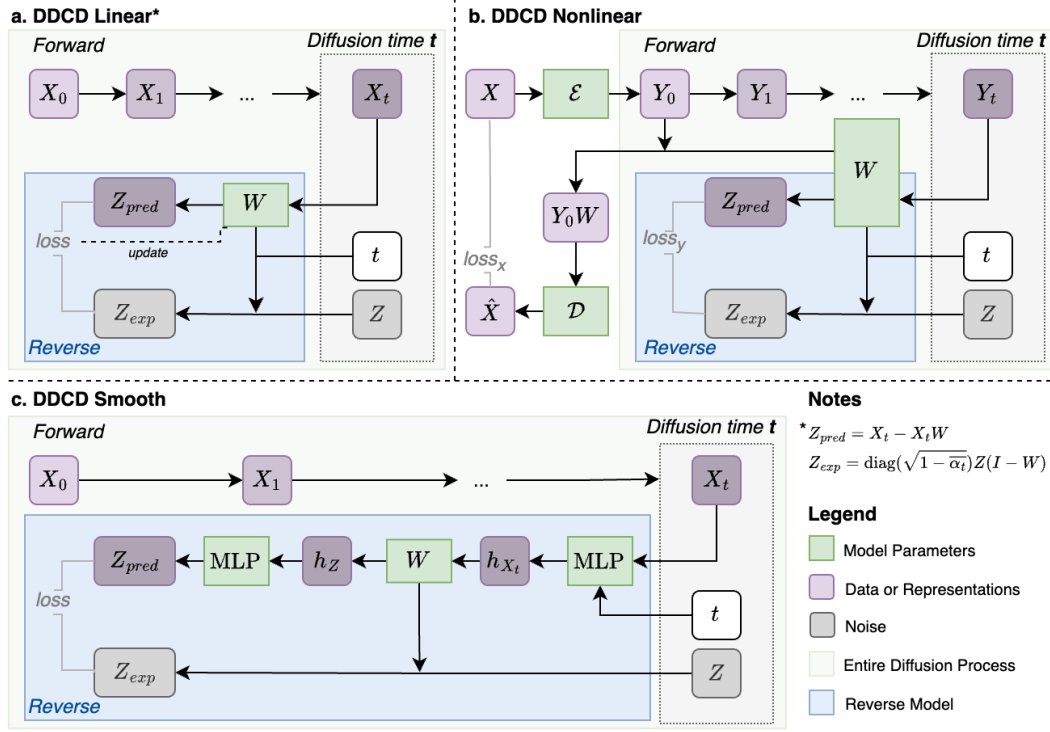


Figure 5.1: Model architectures of proposed models in this chapter.

With Equations 5.4 and 2.5, we can easily develop Equations 5.5-5.7

$$\mathbf{X}_t \mathbf{W} = \text{diag}(\sqrt{\bar{\alpha}_t}) \mathbf{X}_0 \mathbf{W} + \text{diag}(\sqrt{1 - \bar{\alpha}_t}) \mathbf{Z} \mathbf{W}, \quad (5.5)$$

$$\mathbf{X}_t - \mathbf{X}_t \mathbf{W} = \text{diag}(\sqrt{\bar{\alpha}_t}) (\mathbf{X}_0 - \mathbf{X}_0 \mathbf{W}) + \text{diag}(\sqrt{1 - \bar{\alpha}_t}) (\mathbf{Z} - \mathbf{Z} \mathbf{W}) \quad (5.6)$$

$$\text{diag}(\sqrt{\bar{\alpha}_t}) (\mathbf{X}_0 - \mathbf{X}_0 \mathbf{W}) = (\mathbf{X}_t - \mathbf{X}_t \mathbf{W}) - \text{diag}(\sqrt{1 - \bar{\alpha}_t}) \mathbf{Z} (\mathbf{I} - \mathbf{W}) \quad (5.7)$$

Recall that in Equation 2.6, the main objective is to minimize $\mathbf{X}_0 - \mathbf{X}_0 \mathbf{W}$. Equation 5.7 shows that minimizing $\mathbf{X}_0 - \mathbf{X}_0 \mathbf{W}$ is equivalent to minimizing the right-hand side, which measures the distance between $(\mathbf{X}_t - \mathbf{X}_t \mathbf{W})$ and $\text{diag}(\sqrt{1 - \bar{\alpha}_t}) \mathbf{Z} (\mathbf{I} - \mathbf{W})$.

Although we just showed that Equation 5.3 is equivalent to 2.6, in practice the denoising objective smooths out the gradients by adding a noise component that regularizes the learning process. This prevents large changes that could impair convergence, especially when the number of samples is limited. At the same time, by introducing noise, this new objective encourages the model to find solutions that generalize better by avoiding sharp minima. These two characteristics can help the model to converge to an optimal solution more efficiently. To illustrate this point, we implemented a toy model called NOTEARS-Denoising with the denoising diffusion process, while everything else is the same as the original implementation of NOTEARS-Linear.

With the new objective in mind, we also propose a linear DDCD model, where the only trainable parameters are values in the adjacency matrix $\mathbf{W}$. The input of the model includes the perturbed $\mathbf{X}_t$, the diffusion variance schedule $\sqrt{1 - \bar{\alpha}_t}$, and the noise term $\mathbf{Z}$. All three inputs are generated during the forward process given $\mathbf{X}_0$. The model is trained with both L1 and L2 regularization on the adjacency matrix together with a k-hop DAG constraint, which will be explained in section 5.2.4. In terms of optimization, the model is optimized using the Adam optimizer with increasing weights on the DAG constraint. We will explain that in detail in section 5.2.6.

5.2.2 Denoising Diffusion Models for Nonlinear SEMs

The main challenge of applying the de-noising objective to non-linear cases is that, without the linear assumption on the transformation functions (mainly in f_1 in Equation 2.8), we cannot simply separate the noise term $\mathbf{Z}$ from the signals of $\mathbf{X}$. That makes Theorem 1 not applicable to nonlinear SEMs. To overcome this challenge, we introduce an intermediate variable $\mathbf{Y}$ and rewrite the nonlinear SEM in Equation 2.8 in the following form.

$$\mathbf{Y} = f_1(\mathbf{X}) \tag{5.8}$$

$$\mathbf{X} = f_2(\mathbf{Y}\mathbf{W}) + \mathbf{E}_1 \quad (5.9)$$

In addition, we further assume $\mathbf{Y}$ can be modeled with a linear SEM model that satisfies Equation 5.10, allowing us to use the linear denoising diffusion models we discussed previously.

$$\mathbf{Y}_0 = \mathbf{Y}_0\mathbf{W} + \mathbf{E}_2 \quad (5.10)$$

Together, the model will be trained with two main objectives. First of all, we would like to minimize the nonlinear SEM reconstruction loss on $\mathbf{X}$. Secondly, we would like to minimize the linear denoising diffusion loss on $\mathbf{Y}$. The full loss function is,

$$\min_{\mathbf{W}} \frac{1}{2n} (\|\mathbf{X} - f_2(f_1(\mathbf{X})\mathbf{W})\|_F^2 + \|(\mathbf{Y}_t - \mathbf{Y}_t\mathbf{W}) - \text{diag}(\sqrt{1 - \bar{\alpha}_t})\mathbf{Z}(\mathbf{I} - \mathbf{W})\|_F^2) + \lambda_1 \|\mathbf{W}\|_1 + \lambda_2 \|\mathbf{W}\|_2, \quad (5.11)$$

From the deep learning point of view, this architecture, as shown in Figure 5.1b, is very similar to a Latent Diffusion Model (LDM) (Rombach et al., 2022), where noise is added to the latent representation learned by an autoencoder. In this case, we can treat $\mathbf{Y}$ as the unobserved latent variable, and the learned nonlinear transformation functions f_1 and f_2 could be viewed as the encoder and decoder. Furthermore, the learned adjacency matrix $\mathbf{W}$ for the SEM could be considered as a form of attention or graph neural network.

5.2.3 Scale handling with smoothed features and adjacency matrix

While continuous optimization causal discovery algorithms have yielded great success on synthetic SEM datasets, recent studies (Reisach et al., 2021; Kaiser and Sipos, 2021) have shown that they often don't work that well on datasets with standardized features or on real-world datasets where the scales are unknown. The root cause, as pointed out in Kaiser and Sipos (2021), is the use of SEM equations in

Equations 2.5 and 2.7. An underlying assumption of these two SEM equations is that the variables in $\mathbf{X}$ must be on the same scale. If the scales of the features are altered differently due to data standardization, the equation will no longer hold. The DDCCD Linear and DDCCD Nonlinear methods we propose here have the same problem. To overcome this issue, we propose a smoothed version of DDCCD. Instead of trying to learn the exact values in the adjacency matrix of the SEM, DDCCD Smooth tries to learn a normalized adjacency, where the expected values of the edge weights are $\frac{1}{d}$. This normalized adjacency matrix is conceptually similar to the one used in graph convolution (Kipf and Welling, 2016). To do this, we first use an MLP with Tanh activation to normalize all features to the range of -1 to 1 regardless. An illustration of the architecture of this model is provided in Figure 5.1c.

In DDCCD Smooth, because of the normalization on the features, the learned adjacency matrix $\hat{\mathbf{W}}$ is also normalized, and the expected value is $\frac{1}{d}$. This normalized adjacency matrix would be conceptually similar to the normalized adjacency matrix in graph convolution (Kipf and Welling, 2016). Under these assumptions, from the following theorem, instead of estimating the $\mathbf{Z}(\mathbf{I} - \mathbf{A})$ term as used in Equation 5.7, we can reach a special form of conclusion that allows us to predict the added noise $\mathbf{Z}$ directly.

Theorem 5.2.3.1 *With a normalized adjacency matrix, we can directly infer the added noise $\mathbf{Z}$.*

Proof Starting from Equation 5.5, let's pick a random sample $\mathbf{x}$ and perturb that with a Gaussian noise vector $\mathbf{z} \in \mathcal{N}(0, 1)$ to build $\mathbf{x}_t$.

$$\mathbf{W}^T \mathbf{x}_t = \sqrt{\alpha_t} \mathbf{W}^T \mathbf{x}_0 + \sqrt{1 - \alpha_t} \mathbf{W}^T \mathbf{z}, \quad (5.12)$$

We can in fact write each element in $\mathbf{W}^T \mathbf{z}$ as a form of weighted Gaussian mixtures. Since all values in $\mathbf{z}$ are standard Gaussian noise with a mean of 0 and variance of 1, the weighted sum of such a mixture will also be centered at 0. Given

that the expected value of edge weight in $\mathbf{W}$ is $\frac{1}{d-1}$ and there are $d-1$ entries, the expected value for the entire variance is $\sum_{i=0}^d \frac{1-\bar{\alpha}_t}{d^2} = \frac{1-\bar{\alpha}_t}{d}$. When d is large, this variance of $\mathbf{W}^T \mathbf{x}_t$ will be much smaller than the variance term in $\mathbf{x}_t$, which is $1 - \bar{\alpha}_t$. When d is really large and the diffusion coefficient is small, we can therefore use $\mathbf{W}^T \mathbf{x}_t$ to approximate the unperturbed $\mathbf{x}_0$. This argument is very similar to the Central Limit Theorem, but on noise. As a result, $\mathbf{W}^T \mathbf{x}_t - \mathbf{x}_t$ will give us an close estimate of the added noise z .

5.2.4 k-hop DAG Constraint with gradient clipping

In practice, the $\mathcal{O}(d^3)$ runtime of NOTEARS' DAG constraint and its risk of gradient explosion on larger networks is restrictive. Furthermore, many real world networks, such as gene regulatory networks, include cycles and feedback loops. Thus, we believe that starting with the DAG assumption may be incorrect, although we do want to prevent the network from being symmetric. Based on these considerations, by transforming the matrix exponential in NOTEARS to its power expansion form, we propose an alternative "k-hop DAG constraint" that only checks the acyclicity score within k hops. By keeping a running sum, we can reduce the runtime to $\mathcal{O}(k \cdot d^2)$.

The NOTEARS DAG constraint is in the form of $h_{\text{NOTEARS}}(\mathbf{W}) = \text{tr}(e^{\mathbf{W} \circ \mathbf{W}}) - d$, where $\circ$ is the Hadamard product, $e^{\mathbf{W}}$ is the matrix exponential of $\mathbf{W}$, and $\text{tr}()$ is the trace of a matrix. In this case, the matrix exponential is the sum of a weighted power series as shown below.

$$e^{\mathbf{W}} = \sum_{j=0}^{\infty} \frac{1}{j!} \mathbf{W}^j. \quad (5.13)$$

The trace of the summed matrix and the sum of all the traces are equivalent. At the same time, since $\mathbf{W}^0$ is simply the identity matrix, whose trace equals to the value of d , we can rewrite the NOTEARS DAG function in the following form:

$$h_{\text{NOTEARS}}(\mathbf{W}) = \sum_{j=0}^{\infty} \frac{1}{j!} \text{tr}((\mathbf{W} \circ \mathbf{W})^j) - d = \sum_{j=1}^{\infty} \frac{1}{j!} \text{tr}((\mathbf{W} \circ \mathbf{W})^j). \quad (5.14)$$

In this case, if we want to account for all the cycles within k hops, we can do the following calculation:

$$h_{\text{k-hop}}(\mathbf{W}, k) = \sum_{j=1}^{k+1} \frac{1}{j!} \text{tr}((\mathbf{W} \circ \mathbf{W})^j) \quad (5.15)$$

In the case when values in the weighted adjacency matrix are tiny, it might be helpful to multiply the values in the adjacency matrix by a constant multiplier γ and then remove it after the trace calculation. Then the equation becomes to the following form:

$$h_{\text{k-hop}}(\mathbf{W}, k, \gamma) = \sum_{j=1}^{k+1} \frac{1}{j! \gamma^{2j}} \text{tr}((\gamma \mathbf{W} \circ \gamma \mathbf{W})^j) \quad (5.16)$$

If we keep a running product for $j!$, γ^{2j} , and $(\gamma \mathbf{W} \circ \gamma \mathbf{W})^j$, we can keep the complexity within $\mathcal{O}(d^2)$.

In addition, we apply gradient clipping on the model parameters to prevent gradient explosion on large networks.

5.2.5 Fixed-size Bootstrap Sampling

In this chapter, we use the fixed-sized bootstrap sampling design as previously described in Chapter 4 for RegDiffusion. Basically, in each training iteration, we sample a fixed size batch of samples with replacement and add different amounts of noise to the samples. By doing so, we remove the dependency on the number of samples n from the algorithm's runtime and gain more similar behavior on data of different sizes.

5.2.6 Optimization

In this experiment, the models are optimized using the Adam optimizer since it has fewer restrictions. However, the denoising diffusion objective could be applied to the NOTEARS model directly and optimized with L-BFGS without any issues. Another experiment design we tested involves replacing the dual-ascending augmented Lagrangian optimization used in many methods, such as NOTEARS and DAG-GNN, with a simple linear multiplier using training epoch steps. Our justification for this is that our training pattern is much smoother with the fixed-size bootstrap sampling, so we can replace the automatic scaled Lagrangian multiplier with a simple linear multiplier. Another motivation is that in the case when edge weights are mostly smaller than 1 (for example, when features are normalized due to the use of neural networks), it no longer makes sense to test on the quadratic term in the augmented Lagrangian as a heavy penalty.

5.2.7 Experiment Design

Our first experiment is a comparison of results from NOTEARS-Denoising and NOTEARS-Linear. The only differences between these two models are the noise perturbation process and the denoising objective. Both models are optimized using dual-ascending augmented Lagrangian with the L-BFGS-B optimizer. The performance is evaluated on a synthetic Scale-Free graph with 20 nodes and degree 10. We use this example to demonstrate how the denoising objective can smooth out gradients and help the model converge faster.

Then, we evaluate the performance of the proposed linear and nonlinear models on synthetic and real data. The results for synthetic data are compared to a range of well-known causal inference methods: NOTEARS (Zheng et al., 2018), NOTEARS-MLP (Zheng et al., 2018), DAG-GNN (Yu et al., 2019), GOLEM (Ng et al., 2020), and GAE (Ng et al., 2019). We tested the performance of the models on various numbers of nodes, numbers of observations, graph types, and SEM types.

For the synthetic data, we pre-generated a set of Scale-Free (SF) and Erdős-

Rényi (ER) random graphs with a wide range of node counts (20 - 5,000) and edge degrees (10 - 500). Edge weights can be fully positive (ranging from 0.5 to 1.5) or both negative and positive (ranging from -1 to 1). Observational data were generated following additive noise models (ANMs) with both linear and nonlinear transformations. For nonlinear transformations, we follow the examples of DAG-GNN(Yu et al., 2019) and tested the following nonlinear SEM: $\mathbf{x} = \mathbf{W}^T \cos(\mathbf{x} + \mathbf{1}) + \epsilon$. In both linear and nonlinear cases, data was generated with Gaussian noise

To assess the quality of the inferred structures, we report the True Positive Rate (TPR), False Discovery Rate (FDR), False Positive Rate (FPR), Structural Hamming Distance (SHD), and algorithm execution time. We focus on SHD as the main metric. A detailed description of these metrics is provided here.

1. True Positive Rate (TPR) is defined as

$$\text{TPR} = \frac{\text{True Positive}}{\text{Condition Positive}} \quad (5.17)$$

True positive is the number of cases when predicted association exists in condition in correct direction. Condition positive is the total number of true edges in the ground truth graph.

2. False Discovery Rate (FDR) is defined as

$$\text{FDR} = \frac{\text{False Positive} + \text{Reverse}}{\text{Prediction Positive}} \quad (5.18)$$

False positive is the number of cases when predicted association does not exist in condition. Reverse is the number of cases when predicted association exists in condition in opposite direction. Prediction Positive is the total number of positive predictions in the inferred graph.

3. False Positive Rate (FPR) is defined as

$$\text{FPR} = \frac{\text{False Positive} + \text{Reverse}}{\text{Condition Negative}} \quad (5.19)$$

Condition negative is the total number of edges that do not exist.

4. Structural Hamming Distance (SHD) is a measure used to quantify the difference between two directed acyclic graphs (DAGs). It counts the number of operations, including adding an edge, removing an edge, and reversing an edge, required to transform one graph into another. Here, the SHD is the sum of reversed positive predictions, false positive predictions regardless of direction, and false negative predictions regardless of direction.

For real data, we evaluated the proposed methods using the Myocardial Infarction (MI) Complications dataset (Golovenkin et al., 2020) from the UCI data repository, as well as single cell yeast gene expression data (Tjärnberg et al., 2024) (available in GEO with accession number GSE218089 (Edgar et al., 2002)). The MI dataset includes 124 variables and 1,700 observations. We removed the ‘ID’ column and treated all missing values as 0. Since most categorical variables are leveled with increasing severity, we treated them as continuous variables for simplicity. For the yeast gene expression data, we followed the data preprocessing steps described in the original paper; these details are described in the Appendix. We further removed all ribosomal genes. The final data set has 4,980 genes and 1,428 samples. In both cases, since ground truth is not available, the inferred structures are evaluated using domain knowledge.

5.3 Experiments

5.3.1 Denoising objective leads to smoother gradients

An interesting observation about NOTEARS-Linear (Zheng et al., 2018) is that when the number of samples is very limited, it takes a long time to converge (as shown in Figure 5.2a). The reason is that the gradient of the L2 loss used in NOTEARS-Linear is in fact not that smooth, so it will take many steps for the L-BFGS-B optimizer to converge in each iteration (as shown in Figure 5.2b and c). In contrast, with the denoising objective, the gradient is much smoother, so the

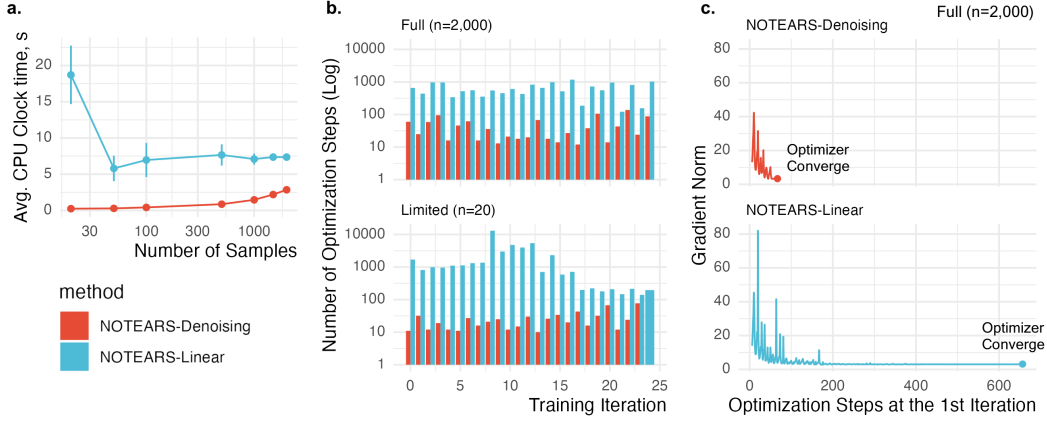


Figure 5.2: NOTEARS-Denoising runs faster than NOTEARS-Linear because it has smoother gradient. a. Average runtime over 10 runs with various number of samples; b. The L-BFGS-B optimizer converges with fewer number of steps on NOTEARS-Denoising; c. Gradient norm during the first iteration on the Full data ($n=2,000$). First 5 steps are removed for visualization purpose.

optimizer only explores a fraction of local minima.

5.3.2 Linear Synthetic Data

Figure 5.3a offers a visual comparison of the inferred networks by DDCD Linear and NOTEARS on a SF graph with 100 nodes. With sufficient numbers of samples ($n = 2,000$), the inferred network from DDCD Linear is nearly identical to the ground truth network ($\text{SHD} = 12$). In the case where the number of samples is extremely insufficient ($n = 20$), the main structures of the network can still be visually identified by DDCD Linear while the results from NOTEARS are more limited.

Figure 5.3b provides a broader comparison of the SHD metrics between different methods on different test cases. In general, DDCD Linear, GOLEM, and DAG-GNN (for SF graphs only) are the most competitive methods on these linear cases. Nonlinear methods tend to do worse on these linear cases, which is expected. It is harder to recover ER graphs compared to SF graphs since the signals are weaker and it's more challenging to model on conditional probability.

In terms of algorithm runtime, all three DDCD models finish execution at a fraction of the cost of other algorithms. The only comparable baseline method is

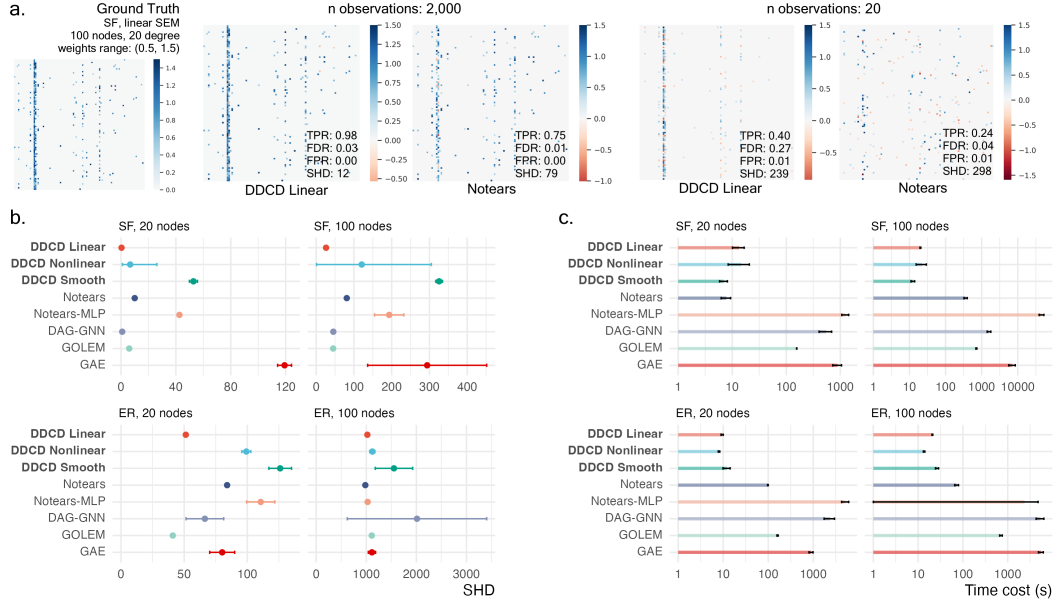


Figure 5.3: a. Example heatmap of weight estimates on a 100-node scale free graph with different numbers of observed samples using DDCCD Linear. b. Benchmark Results on 2,000 observations on linear synthetic data over 10 runs, evaluated using Structural Hamming Distance (SHD). Lower scores indicate better performance. (SF: scale-free; ER: Erdős-Rényi) c. Algorithm execution time on 2,000 observations on CPUs over 10 repeated runs. Time axis is displayed in log scale.

NOTEARS, which finished in an average of 8 seconds and is the fastest algorithm on SF graphs with 20 nodes. However, the time cost quickly scales up to 6 minutes on SF graphs with 100 nodes. In contrast, for DDCCDs, the execution time only extends from around 10 seconds to around 20 seconds.

5.3.3 Nonlinear Synthetic Data

The results for the nonlinear synthetic data are included in Figure 5.4. This time, the DDCCD Nonlinear model demonstrates strong performance in recovering the causal structures from nonlinear data. In the example provided in Figure 5.4a, DDCCD Nonlinear not only generates an accurate weight estimate of the graph (TPR: 0.91, SHD: 126), but also provides an approximation to the underlying nonlinear transformation function. After training is complete, we can send the input data $\mathbf{X}$ through the trained encoder and decoder and obtain the predicted values for $\mathbf{Y}$ and $\hat{\mathbf{X}}$. The relationship between $\mathbf{X}$ and $\mathbf{Y}$ will be the transformation function f_1 and

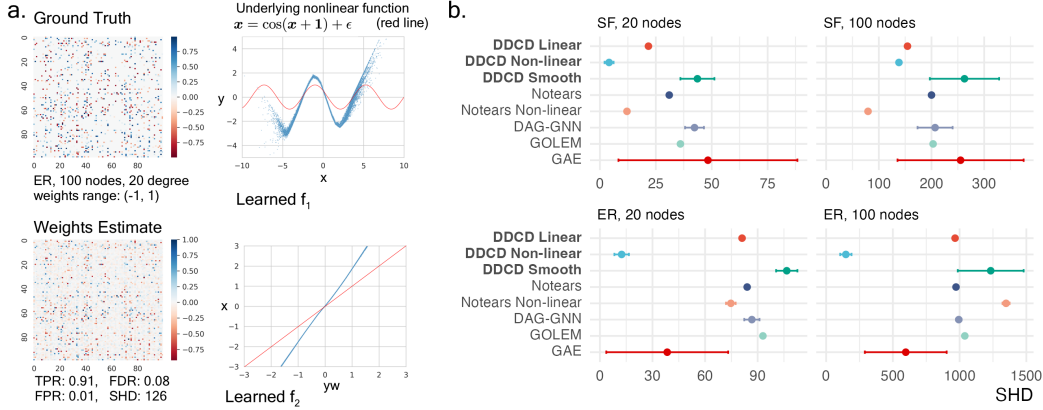


Figure 5.4: a. Example Weight Estimates on an ER graph with 100 nodes using DDCCD Nonlinear. DDCCD Nonlinear not only infers the causal structure but also offers a way to approximate the underlying nonlinear transformation function. (Approximation indicated by blue dots in the second column. The ground truth transformation function is indicated by the red line). b. Benchmark Results with 2,000 observations over 10 runs evaluated using SHD.

the relationship between $\mathbf{Y}\mathbf{W}$ and $\hat{\mathbf{X}}$ will be f_2 . In addition to DDCCD Nonlinear, DDCCD Linear also presents competitive benchmark performance despite being a linear model.

5.3.4 Real World Observational Data: Myocardial Infarction

In this section, we assess the performance of DDCCD Smooth on the real-world myocardial infarction dataset. After training was complete, we extracted all edges with weights above the cut-off threshold (0.2) in the inferred normalized adjacency matrix and examined a graph of the 2-hop neighbors around the Lethal Outcome (LET_IS) node. This cut-off threshold was set by inspecting the distribution of edge weights and selecting the right tail. As shown in Figure 5.5, we can identify several meaningful node clusters in this graph, including lethal outcome with its primary causes, critical conditions and interventions, hospital pain control, emergency cardiology pain control, and blood test results, purely based on the topological relationships among the nodes.

The three most important direct causes of lethal outcome include myocardial rupture, cardiogenic shock, and complete Right Bundle Branch Block (RBBB) on

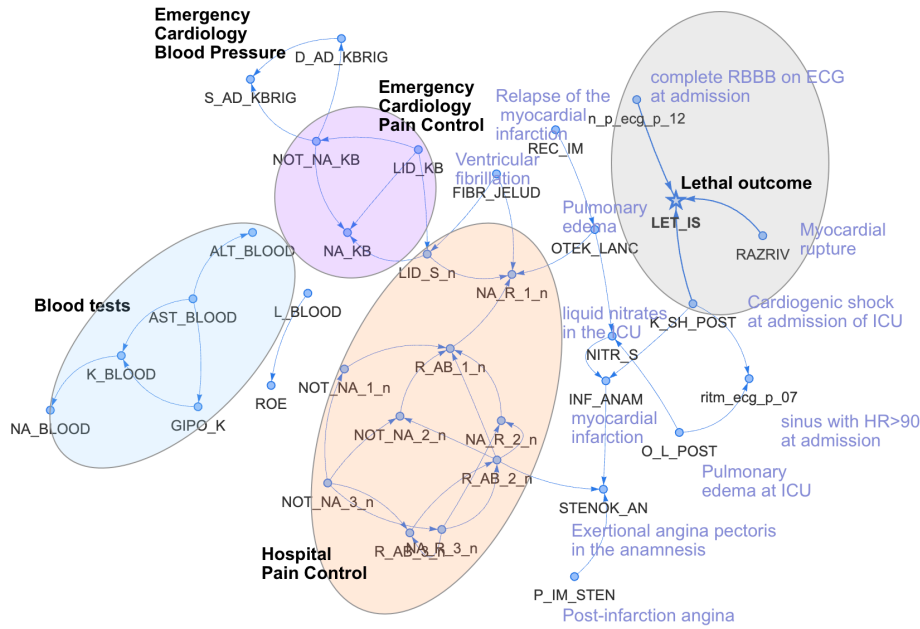


Figure 5.5: Inferred Causal Network around Lethal Outcome in Myocardial Infarction

ECG at admission; all of these are known to be conditions with a poor prognosis. Cardiogenic shock (K_SH_POST) is further shown to cause “sinus ECG rhythm with heart rate > 90 ” ($ritm_ecg_p_07$), consistent with a high heart rate (tachycardia) being a symptom of cardiogenic shock. Pulmonary edema ($OTEK_LANC$) is shown to cause the use of liquid nitrates in the ICU ($NITR_S$); this is indeed a common practice for rapidly managing pulmonary edema. Other inferred edges include that NSAID drugs used by the emergency team (NOT_NA_KB) cause blood pressure to increase, and that relapsing pain in the 2nd hospitalization period causes NSAID use in the same period.

There are also some node pairs for which plausible edges are inferred, but in an implausible direction. For example, in the lower right of the figure, “post-infarction angina” (chest pain after the heart attack causing the current hospital admission) is shown to cause “exertional angina pectoris in the anamnesis” (e.g., a reported history of chest pain after exercise), when the former clearly occurs after the latter. Still, many of the directed edges appear consistent with known causal

relationships.

5.3.5 Real World Observational Data: Gene Regulatory Networks

Another real world experiment we conducted is inferring a gene regulatory network (GRN) on a yeast single cell dataset. We collected the *Saccharomyces cerevisiae* single cell expression data from NCBI’s GEO database (Edgar et al., 2002) with accession GSE218089 (Tjärnberg et al., 2024). We selected the yeast wild-type strain 2 cultured in nutrient-rich YPD media, as described in Jackson et al. (2020). We followed the standard raw count data pre-processing procedures, which include, (1) gene filtering, by removing genes with positive expression counts in fewer than 10 cells; (2) size factor calculation (Anders and Huber, 2010), which involves calculating the geometric mean of counts across all samples, calculating the ratio for each gene within a sample by dividing each sample’s count by the geometric mean for the genes, and calculating the size factor by computing the median of the ratios for each sample; (3) data normalization, normalizing the count by dividing by the size factors; and (4) log transformation, transforming (count-plus-one) using the natural logarithm. After these data processing steps, we performed gene ID conversion using g:Profiler (Raudvere et al., 2019), and we further removed all ribosomal genes (genes with “RPS” or “RPL” prefixes in their names). The processed yeast single cell dataset includes 4,980 genes from 1,428 samples.

We run RegDiffusion, which does not account for the DAG constraint; DDCCD Smooth with a 2-hop DAG constraint; and DDCCD Smooth with a 10-hop DAG constraint. All networks are constructed with similar sizes (16 hidden dimensions, 3 layers of MLP blocks) and are trained for 1000 iterations on GPUs. In terms of clock time, RegDiffusion completed in 20 seconds, DDCCD Smooth 2-hop DAG in 34 seconds, and DDCCD Smooth 10-hop DAG in 109 seconds.

Here we treat the STRING protein-protein interaction network (Szkarczyk et al., 2023) as a noisy and non-context-specific ground truth network. In contrast, we would consider the 3 inferred networks to be context-specific since they are inferred from context specific (wild type, cultured in nutrient-rich media) gene

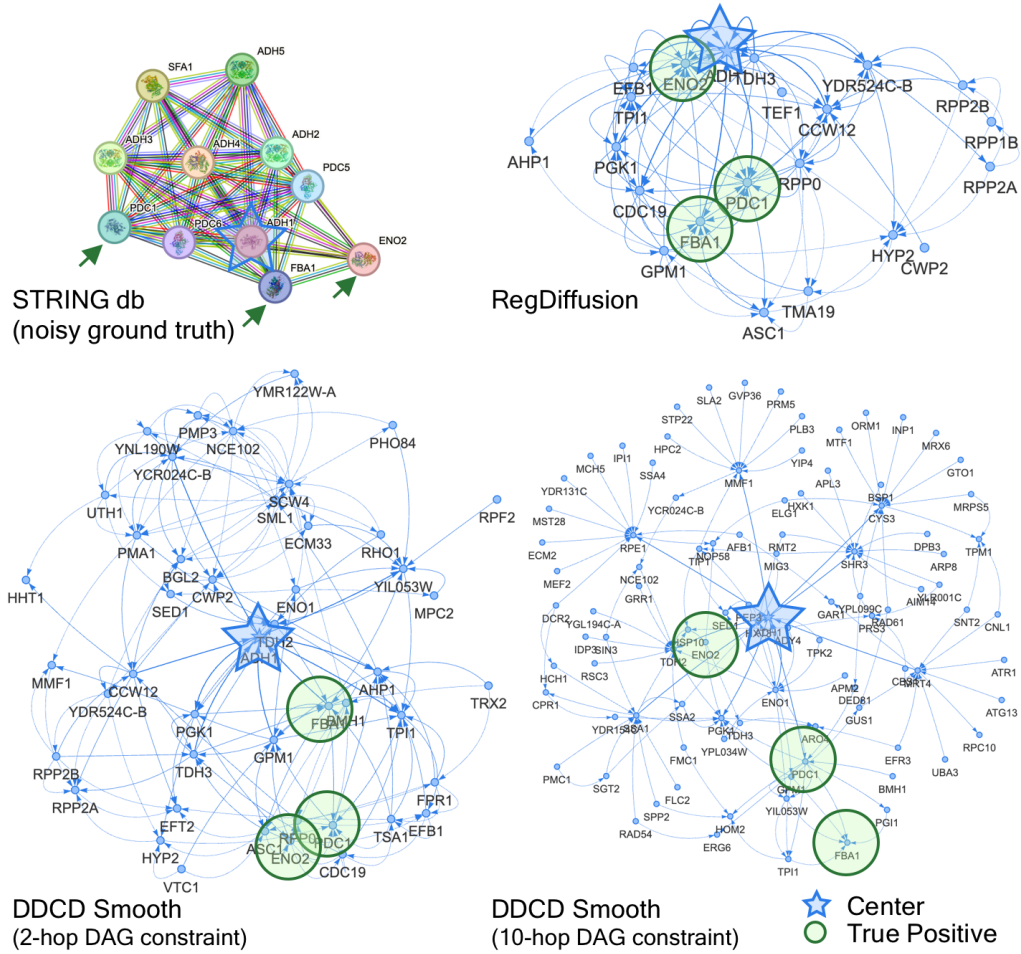


Figure 5.6: Comparison of the inferred gene regulatory neighborhood around *ADH1* using different models and settings. The network from STRING db is considered as a noisy and non-context-specific ground truth network.

expression data.

We compared the inferred networks by spot checking local neighborhoods together with a systematic evaluation using similar metrics as used in Chapter 4. In the local neighborhood visualization, we inspected the local 2-hop neighborhood around *AHD1*, a key player in ethanol fermentation that has been extensively studied in the context of alcohol metabolism and glycolysis (Raj et al., 2014). The inferred networks are shown in Figure 5.6; true positive predictions of neighboring nodes of *ADH1* are marked with green circles. In the STRING network, *ADH1* is shown to interact with *FBA*, *ENO*, *PDC*, *ADH*, and *SFA*. All three inferred networks successfully captured the links with *FBA*, *ENO*, and *PDC* but their network

Method	EP	EPR	AUPRC	AUPRC Ratio	AUROC
DDCD Smooth, k=2	11548	1.296	0.021	1.067	0.513
DDCD Smooth, k=10	10497	1.178	0.020	1.047	0.512
RegDiffusion	26119	2.931	0.028	1.434	0.534

Table 5.1: Systematic Evaluation Results on the Yeast Network using similar metrics as used in Chapter 4.

topologies are quite different.

In the network from RegDiffusion, where we include the top 15 connected candidates for each gene, all three genes are directly connected, forming a chain ($FBA1 \leftrightarrow ENO2$) $\rightarrow ADH1 \rightarrow PDC1$. DDCD Smooth with the 2-hop constraint forms two pathways starting from $FBA1$: $FBA1 \rightarrow PDC1$ and $FBA1 \rightarrow ADA1 \rightarrow PGK1 \rightarrow ENO2 \rightarrow PDC1$. DDCD Smooth with the 10-hop constraint forms several fragmented pathways, including $PDC1 \rightarrow FBA1$, $PDC1 \rightarrow ADH1$, and $ENO2, AHD1 \rightarrow PGK1$.

Although all three models identify proximity to these three neighboring genes, in DDCD Smooth with the 2-hop constraint, neither $ENO2$ nor PDC is directly connected to $ADH1$. In DDCD Smooth with the 10-hop constraint, neither $FBA1$ nor $ENO2$ is directly connected to $ADH1$. Further, most of the 2-hop neighboring nodes in the DDCD Smooth (10 hop) plots do not interact with each other, showing that the inferred candidate neighbors lack functional coherence. In contrast, in more accurate inferred networks, the 2-hop neighbors of most genes interact with each other extensively.

On the other hand, in the systematic evaluation, to avoid including too many edges, we only included experiment-based edges. The results are included in Table 5.1. Note that in the STRING network, the edges are undirectional so RegDiffusion, which didn't control the directionality of edges, is at advantage. However, this alone couldn't explain why the Early Precision of RegDiffusion is more than 2 times of DDCD Smooth (k=2).

Based on the local network inspections and the systematic evaluation, we conclude that the inferred network from RegDiffusion is more trustworthy. Intro-

ducing even limited acyclicity controls to gene networks, or to other networks that potentially include longer feedback loops, does not appear to improve inference results.

5.4 Discussion and Conclusion

Our benchmarks on synthetic data demonstrate the superior performance of the DDCCD models in both linear and nonlinear cases. The denoising nature allows the model to explore a broader range of noise and to yield better estimates, especially when the number of observed samples is limited. The capacity for recovering an accurate approximation of nonlinear transformation functions further assists the model in approximating the truth. Since it runs very quickly, it may even be used as a low-cost nonlinearity test in appropriate scenarios.

Compared to existing methods, the most similar model to DDCCD is DAG-GNN, which also models the noise term of the SEM equation. To some degree, both DDCCD and DAG-GNN train a decoder to reconstruct the original input from pure noise under the constraint of the weighted adjacency matrix. This is similar to the comparison of diffusion models to VAEs with infinite latent spaces (Luo, 2022). By modeling the noise prediction and expected noise under the constraint of the parameterized adjacency matrix at the same time, we eliminate the requirement of doing matrix inversion, which runs in $\mathcal{O}(d^3)$, in DAG-GNN. At the same time, DDCCD allows us to experiment with more flexible neural network architectures, addressing multiple types of assumptions.

Since both DDCCD Linear and DDCCD Nonlinear are based on the same assumptions that NOTEARS, GOLEM, and DAG-GNN make, they also suffer from the same problems pointed out in Reisach et al. (2021) and Kaiser and Sipos (2021). In our real-world experiments, the results from DDCCD Smooth are also much more explainable than the results from DDCCD Linear and Nonlinear. This once again marks the gap between real and current synthetic data in this field. However, there have been several proposed solutions (Ng et al., 2024; Nazaret et al., 2023) to solve

these problem with unequal variance. Incorporating these into our linear and non-linear models is an important topic for future research.

Overall, for real world use cases, we recommend using DDCCD Smooth. For synthetic data, we recommend using DDCCD non-linear first and check the shape of the inferred non linear transformations. If both transformations seem to be linear, the use of DDCCD linear could be applied with confidence. Causal discovery is still being actively researched with many remaining research questions. Applying this type of inference on real world applications, such as GRN inference, requires further validations on the directionality of the inferred edges. On the other hand, single cell data, especially the new perturb seq provides an excellent testing ground for developing new theories and practically useful learning methods in the causal discovery field. Integrating these two world may yield many fruitful results in the next decades.

Chapter 6

Discussion and Future Works

In this thesis, I made several contributions to the field of network inference. Looking back from this point, I think RegDiffusion, as a tool, has the potential to make a big impact in the GRN inference community because of its high performance in speed and inference accuracy. It has been tested in a number of cases beyond the experiments we did for the paper and it provided very good results. The inferred networks could be easily built into existing GRN analysis pipelines, including regulon analysis or GRN based cell type clustering. At the same time, the entire Dropout Augmentation idea provides a new perspective to consider noise in single cell data, and it may inspire other works in single cell RNA-seq, ATAC-seq, and other forms of sparse data (unfortunately, zero inflation seems to be a prevalent problem in our field). DDGD, on the other hand, may be able to be extended to a generalized hypothesis generating tool. There is a growing awareness in the field of economics and epidemiology research that researchers should have a hypothetical DAG while carrying out their analyses. While expert knowledge is crucial for developing these DAGs, data-driven tools could also provide some fresh perspective.

As a researcher, it is also a very joyful experience to explore how to utilize noise, which is typically considered as unfavorable, in the learning process. Overall, the utilization of noise could be viewed from several learning perspectives.

- Noise injection could be viewed as a model regularization method that may

help the model come to a more generalized results. It is widely acknowledged that our training sample is sampled from the general population that it represents. By introducing these small variations, we will add some small variations to the input data, which will prevent the model being overfitted to the specific samples that we draw. In terms of the model optimization landscape, we are adding some “paddings” around the optimization minima so they are not sharp. This is particularly useful for noisy and incomplete data, such as single cell RNA-seq.

- As mentioned in Chapter 5 in Figure 5.2, noise injection and the denoising process may also help smooth out the gradient during training. With the added noising and the denoising objective, we are adding some extra room of error so we could avoid sharp gradient updates. The properties helps the model to converge faster.
- The elimination of the matrix inversion step from the VAE-based SEM is also interesting. In fact, VAE-based SEM models also models on the noise term in the SEM formula. However, they have both encoder and decoder and they need to use the adjacency matrix on both sides (one regular and one inverse). In RegDiffusion and DDCCD-Smooth, we are basically replacing this model structure with a decoder only design. By doing this, we eliminated the matrix inversion step and reduce the number of large matrix multiplication from two to one. When the number of genes/entities is large, it significantly reduces runtime of the algorithm.

In addition, while I was working on interpreting the inferred complex networks, I realized the power of network visualization in the process of understanding the results. I also realized the limitation of current tools in terms of scalability and functionality. Therefore, based on my experience, I developed a d3.js based network visualization tool on HTML canvas. It is scalable, fast, and interactive. At the same time, it offers capacity on manually label clusters by selected nodes. This work is still in progress and we have not done any quantitative research to study the gain.

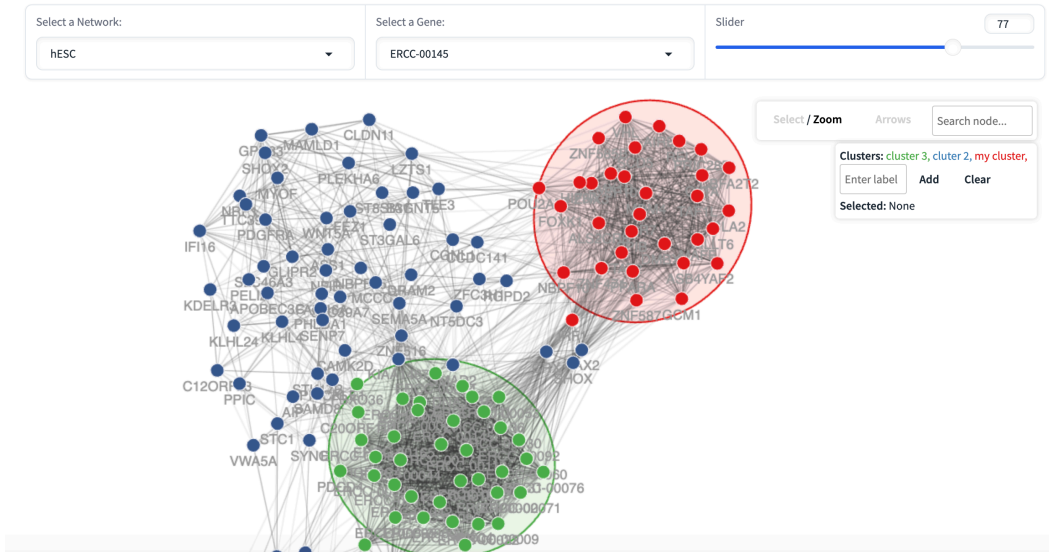


Figure 6.1: Screenshot of the user interface of lightGraph. In the Zoom mode, you can zoom and pan the interface. In the Select mode, you can also drag and select nodes and assign labels to them. An oval will be drawn around the cluster to improve visual clarity.

Therefore, here I'm including a screenshot in Figure 6.1.

While I'm excited about the current results, they also open up new questions to further investigate in the future. Here, I'm separating these directions into two tracks: GRN inference, and causal discovery.

For GRN inference:

- First of all, given several samples with different contexts (intervention vs. control, different time points), how can we compare the inferred networks? While we have made some progress on this topic, there is more to be done. One idea is to treat condition as a node on the graph. So by investigating the neighbor of the condition node, we may be able to identify genes or pathways that are significantly changed. In fact, we are already effectively doing this in DDGD by treating all features/conditions as nodes. But for gene networks, this might be an idea worth investigating. By doing so, we assume that most interactions on the graph are stable and the impact of the context on the rest of the graph is additive.
- Second, the directionality of the edges requires more investigation. While

tools such as RegDiffusion and DeepSEM do generate directional graphs and it's okay to have feedback loops in GRNs, there are just way too many feedback loops in the current results. We did experiment with the idea of k-hop acyclicity control, but it didn't provide us what we were expecting. We need to investigate more sophisticated ways to model the directionality of the edges.

- Next, right now the GRN inference tools I have been working on takes in a single-shot of single-cell RNA-seq and outputs a single-shot inference of GRN. How can we better utilize the huge amount of single-cell data researchers have generated so far? What's the best way to organize and model GRNs under different contexts?
- Can we discover cell types using the learned regulatory relationship? Cell type identification is a major task in single cell analysis and it has large impact on the downstream tasks (e.g. GRN inference). Right now, it is possible to feed the inferred network by RegDiffusion to the SCENIC (Aibar et al., 2017) pipeline and perform clustering based on regulon/interactome patterns. First of all, we need to do a more systematic comparison to quantify how much gain we can achieve by replacing GRNBoost2 with RegDiffusion in their package. Second, are there any other more explainable ways to describe this type of regulatory patterns? Can we create a database of this kind of "marker pathway" or "marker neighborhood" as alternatives to marker genes?
- Similar to the task discussed above, can we use RegDiffusion to do cell-cell interaction or trajectory analysis? This might be easier than we thought: if we transpose the input data so RegDiffusion will treat the cell axis as features, the network we are constructing might has some physical meaning. In fact, in a preliminary experiment, I simply run RegDiffusion on the transposed data of a single cell dataset (BEELINE hESC) with 4 different time points. The result, as shown in Figure 6.2, seems to make sense. However, this method currently only scale to 20,000 cells and requires further validation.

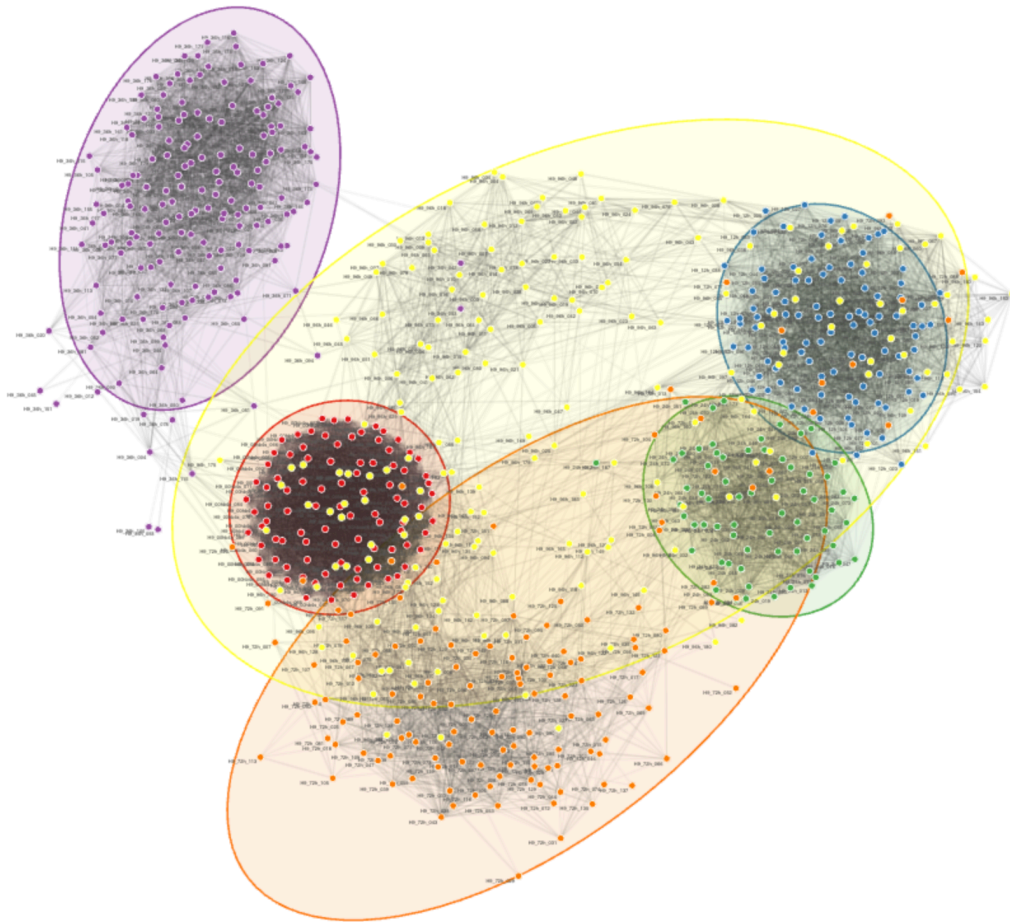


Figure 6.2: An example of running Cell-Cell Interaction using transposed single cell data.

- I would like to see how RegDiffusion could be used together with other popular tools in the community.
- I'm also curious to investigate more into the direction of perturb-seq analysis. With perturb-seq, it's much easier to discover which downstream genes are affected by perturbing the expression of one gene. It still seems to be difficult to separate direct or indirect cause but it has high potentials. I would like to do more research on what that means to GRN discovery and how we can better utilize the data.

For causal discovery:

- In the current experiments, while I did provide empirical evidence that adding

noise helps smooth out the gradients, I can't seem to provide a explanation on the theoretical part. I would like to have an extension on that in the future.

- Currently, there is a huge gap between synthetic data based on Additive Noise Models (ANM) and real-world data. What exactly causes the difference? How can we generate better synthetic data?
- The scaling part of current causal discovery models is very tricky. Many proposed methods works so well on synthetic data but not so much after the data is standardized. There have been discussions on the cause but how we can overcome this challenge is still an open question. In my experiments, I found that while the proposed DDCD-Smooth (similar to RegDiffusion) works well on real-world data, it doesn't work well enough on synthetic data. Is there a theoretical bound on what we can learn?
- Engineering wise, I would like to develop a DAG discovery tool where researchers could input a CSV and then obtain a data-driven DAG.

As I look ahead, I feel very optimistic for the future of this field. With advanced sequencing techniques, such as single-cell RNA-seq, spatial single-cell, and perturb-seq, researchers have unprecedented opportunities to study the state of cells under different conditions at reasonable costs. These new research techniques bring in new challenges and demand innovative computational methods. This is the time when developing novel methodologies can make a significant impact to the progress of biological and medical science. Moreover, these new data and evidence provide fresh opportunities to develop advanced causal and network inference methods. Such advancements can help us build a more robust causal understanding of the complex systems that govern our genetic mechanisms, our public health, and our society.

Bibliography

Francis E Agamah, Jumamurat R Bayjanov, Anna Niehues, Kelechi F Njoku, Michelle Skelton, Gaston K Mazandu, Thomas HA Ederveen, Nicola Mulder, Emile R Chimusa, and Peter AC 't Hoen. Computational approaches for network-based integrative multi-omics analysis. *Frontiers in Molecular Biosciences*, 9: 967205, 2022. doi: 10.5353/th_991044128172503414.

Sara Aibar, Carmen Bravo González-Blas, Thomas Moerman, Vân Anh Huynh-Thu, Hana Imrichova, Gert Hulselmans, Florian Rambow, Jean-Christophe Marine, Pierre Geurts, Jan Aerts, et al. Scenic: single-cell regulatory network inference and clustering. *Nature methods*, 14(11):1083–1086, 2017. doi: 10.3410/f.731968869.793540434.

Nicola Amodio, Lavinia Raimondi, Giada Juli, Maria Angelica Stamato, Daniele Caracciolo, Pierosandro Tagliaferri, and Pierfrancesco Tassone. Malat1: a drug-gable long non-coding rna for targeted anti-cancer approaches. *Journal of hematology & oncology*, 11:1–19, 2018. doi: 10.1186/s13045-018-0606-4.

Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Nature Precedings*, pages 1–1, 2010. doi: 10.1038/npre.2010.4282.2.

Cédric Arisdakessian, Olivier Poirion, Breck Yunits, Xun Zhu, and Lana X Garmire. Deepimpute: an accurate, fast, and scalable deep neural network method to impute single-cell rna-seq data. *Genome biology*, 20:1–14, 2019. doi: 10.1101/353607.

Luciana B Arruda, Del Sim, Priya R Chikhlikar, Milton Maciel, Kenji Akasaki, J Thomas August, and Ernesto TA Marques. Dendritic cell-lysosomal-associated

membrane protein (lamp) and lamp-1-hiv-1 gag chimeras have distinct cellular trafficking pathways and prime t and b cell responses to a diverse repertoire of epitopes. *The Journal of Immunology*, 177(4):2265–2275, 2006. doi: 10.4049/jimmunol.177.4.2265.

Scott Ayton, Noel G Faux, and Ashley I Bush. Ferritin levels in the cerebrospinal fluid predict alzheimer’s disease outcomes and are regulated by apoe. *Nature communications*, 6(1):6760, 2015. doi: 10.1016/j.nbd.2018.12.010.

Timothy Bailey, Pawel Krajewski, Istvan Ladunga, Celine Lefebvre, Qunhua Li, Tao Liu, Pedro Madrigal, Cenny Taslim, and Jie Zhang. Practical guidelines for the comprehensive analysis of chip-seq data. *PLoS computational biology*, 9(11): e1003326, 2013.

Suzanne J Baker, Eric R Fearon, Janice M Nigro, Stanley R Hamilton, Ann C Preisinger, J Milburn Jessup, Peter VanTuinen, David H Ledbetter, David F Barker, Yusuke Nakamura, et al. Chromosome 17 deletions and p53 gene mutations in colorectal carcinomas. *Science*, 244(4901):217–221, 1989.

Ahmed Z Balboula, Paula Stein, Richard M Schultz, and Karen Schindler. Rbbp4 regulates histone deacetylation and bipolar spindle assembly during oocyte maturation in the mouse. *Biology of reproduction*, 92(4):105–1, 2015. doi: 10.1095/biolreprod.115.128298.

R. Batorsky, A. M. Ceasrine, L. L. Shook, S. Kislal, E. A. Bordt, B. A. Devlin, R. H. Perlis, D. K. Slonim, S. D. Bilbo, and A. G. Edlow. Hofbauer cells and fetal brain microglia share transcriptional profiles and responses to maternal diet-induced obesity. *Cell Reports*, 43(6):114326, Jun 2024. doi: 10.1101/2023.12.16.571680.

Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. Dagma: Learning dags via m-matrices and a log-determinant acyclicity characterization. *Advances in Neural Information Processing Systems*, 35:8226–8239, 2022. doi: 10.1515/9781400835416.380.

Marouen Ben Guebila, Tian Wang, Camila M Lopes-Ramos, Viola Fanfani, Des Weighill, Rebekka Burkholz, Daniel Schlauch, Joseph N Paulson, Michael Altenbuchinger, Katherine H Shutta, et al. The network zoo: a multilingual package for the inference and analysis of gene regulatory networks. *Genome Biology*, 24(1):45, 2023.

Mariko L Bennett, F Chris Bennett, Shane A Liddelow, Bahareh Ajami, Jennifer L Zamanian, Nathaniel B Fernhoff, Sara B Mulinyawe, Christopher J Bohlen, Aykezar Adil, Andrew Tucker, et al. New tools for studying microglia in the mouse and human cns. *Proceedings of the National Academy of Sciences*, 113(12):E1738–E1746, 2016. doi: 10.3410/f.726151340.793515827.

Peter J Bickel, Eugene A Hammel, and J William O’Connell. Sex bias in graduate admissions: Data from berkeley: Measuring bias is harder than is usually assumed, and the evidence is sometimes contrary to expectation. *Science*, 187(4175):398–404, 1975.

Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995. doi: 10.1162/neco.1995.7.1.108.

Fabrizio Biundo, Keita Ishiwari, Dolores Del Prete, and Luciano D’Adamio. Interaction of apoe3 and apoe4 isoforms with an itm2b/bri2 mutation linked to the alzheimer disease-like danish dementia: Effects on learning and memory. *Neurobiology of learning and memory*, 126:18–30, 2015. doi: 10.1016/j.nlm.2015.10.009.

Delphine Boche and Marcia N Gordon. Diversity of transcriptomic microglial phenotypes in aging and alzheimer’s disease. *Alzheimer’s & Dementia*, 18(2):360–376, 2022. doi: 10.1002/alz.12389.

Carmen Bravo González-Blas, Seppe De Winter, Gert Hulselmans, Nikolai Hecker, Irina Matetovici, Valerie Christiaens, Suresh Poovathingal, Jasper Wouters, Sara Aibar, and Stein Aerts. Scenic+: single-cell multiomic inference of enhancers and gene regulatory networks. *Nature methods*, 20(9):1355–1367, 2023. doi: 10.1101/2022.08.19.504505.

Li-Jun Cai, Li Tu, Xiao-Mo Huang, Jia Huang, Nan Qiu, Guang-Hong Xie, Jian-Xiong Liao, Wei Du, Ying-Yue Zhang, and Jin-Yong Tian. Lncrna malat1 facilitates inflammasome activation via epigenetic suppression of nrf2 in parkinson's disease. *Molecular brain*, 13(1):1–15, 2020. doi: 10.1186/s13041-020-00656-8.

Thalia E Chan, Michael PH Stumpf, and Ann C Babbie. Gene regulatory network inference from single-cell data using multivariate information measures. *Cell systems*, 5(3):251–267, 2017. doi: 10.1101/082099.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002. doi: 10.7717/peerj-cs.464/table-3.

Max Chickering, David Heckerman, and Chris Meek. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004. doi: 10.1007/11821069_27.

Darren A Cusanovich, Bryan Pavlovic, Jonathan K Pritchard, and Yoav Gilad. The functional consequences of variation in transcription factor binding. *PLoS genetics*, 10(3):e1004226, 2014.

B de Saint-Vis, J Vincent, S Vandenabeele, B Vanbervliet, J-J Pin, S Ait-Yahia, S Patel, M-G Mattei, J Banchereau, S Zurawski, et al. A novel lysosome-associated membrane glycoprotein, dc-lamp, induced upon dc maturation, is transiently expressed in mhc class ii compartment. *Immunity*, 9(3):325–336, 1998. doi: 10.1016/s1074-7613(00)80615-9.

Albert B DeLeo, Gilbert Jay, Ettore Appella, Garrett C Dubois, Lloyd W Law, and Lloyd J Old. Detection of a transformation-related antigen in chemically induced sarcomas and other transformed cells of the mouse. *Proceedings of the National Academy of Sciences*, 76(5):2420–2424, 1979.

Chang Deng, Kevin Bello, Bryon Aragam, and Pradeep Kumar Ravikumar. Optimizing notears objectives via topological swaps. In *International Conference on*

Machine Learning, pages 7563–7595. PMLR, 2023. doi: 10.1109/icmla58977.2023.00118.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *cell*, 167(7):1853–1866, 2016.

Elie Dolgin. The most popular genes in the human genome. *Nature*, 551(7681):427–432, 2017.

R. Edgar, M. Domrachev, and A.E. Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.*, 30(1):207–10, Jan 2002. doi: 10.22541/au.158471624.47405051.

Daniel Eliyahu, Avraham Raz, Peter Gruss, David Givol, and Moshe Oren. Participation of p53 cellular tumour antigen in transformation of normal embryonic cells. *Nature*, 312(5995):646–649, 1984.

Daniel Eliyahu, Dan Michalovitz, Siona Eliyahu, Orit Pinhasi-Kimhi, and Moshe Oren. Wild-type p53 can inhibit oncogene-mediated focus formation. *Proceedings of the National Academy of Sciences*, 86(22):8763–8767, 1989.

Frank Emmert-Streib, Matthias Dehmer, and Benjamin Haibe-Kains. Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. *Frontiers in cell and developmental biology*, 2:38, 2014. doi: 10.3389/fcell.2014.00038.

ENCODE Project Consortium. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57, 2012. doi: 10.1038/nature11247.

Jeremiah J Faith, Boris Hayete, Joshua T Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J Collins, and Timothy S Gardner. Large-scale mapping and validation of escherichia coli transcriptional

regulation from a compendium of expression profiles. *PLoS biology*, 5(1):e8, 2007. doi: 10.1371/journal.pbio.0050008.

Rosalía Fernández-Calle, Sabine C Konings, Javier Frontiñán-Rubio, Juan García-Revilla, Lluís Camprubí-Ferrer, Martina Svensson, Isak Martinson, Antonio Boza-Serrano, José Luís Venero, Henrietta M Nielsen, et al. Apoe in the bullseye of neurodegenerative diseases: impact of the apoe genotype in alzheimer’s disease pathology and brain diseases. *Molecular Neurodegeneration*, 17(1):62, 2022. doi: 10.1186/s13024-022-00566-4.

Cathy A Finlay, Philip W Hinds, and Arnold J Levine. The p53 proto-oncogene can act as a suppressor of transformation. *Cell*, 57(7):1083–1093, 1989.

Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe’er. Using bayesian networks to analyze expression data. In *Proceedings of the fourth annual international conference on Computational molecular biology*, pages 127–135, 2000. doi: 10.3410/f.1009865.137307.

Carlo Sala Frigerio, Leen Wolfs, Nicola Fattorelli, Nicola Thrupp, Iryna Voytyuk, Inga Schmidt, Renzo Mancuso, Wei-Ting Chen, Maya E Woodbury, Gyan Srivastava, et al. The major risk factors for alzheimer’s disease: age, sex, and genes modulate the microglia response to a β plaques. *Cell reports*, 27(4):1293–1306, 2019. doi: 10.3410/f.735596218.793562726.

François Fuks, Wendy A Burgers, Alexander Brehm, Luke Hughes-Davies, and Tony Kouzarides. Dna methyltransferase dnmt1 associates with histone deacetylase activity. *Nature genetics*, 24(1):88–91, 2000. doi: 10.1038/71750.

Luz Garcia-Alonso, Christian H Holland, Mahmoud M Ibrahim, Denes Turei, and Julio Saez-Rodriguez. Benchmark and integration of resources for the estimation of human transcription factor activities. *Genome research*, 29(8):1363–1375, 2019. doi: 10.1101/337915.

Shila Ghazanfar, Adam J Bisogni, John T Ormerod, David M Lin, and Jean YH

- Yang. Integrated single cell data analysis reveals cell specific networks and novel coactivation markers. *BMC systems biology*, 10(5):11–24, 2016. doi: 10.1186/s12918-016-0370-4.
- Kimberly Glass, Curtis Huttenhower, John Quackenbush, and Guo-Cheng Yuan. Passing messages between biological networks to refine predicted interactions. *PloS one*, 8(5):e64832, 2013.
- Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019. doi: 10.3389/fgene.2019.00524.
- SE Golovenkin, AN Gorban, EM Mirkes, VA Shulman, DA Rossiev, PA Shesternya, S Yu Nikulina, Yu V Orlova, and MG Dorrer. Complications of myocardial infarction: a database for testing recognition and prediction systems. UCI Machine Learning Repository, 2020. DOI: <https://doi.org/10.24432/C53P5M>.
- Andrew Devon Gordon, Saumik Biswas, Biao Feng, and Subrata Chakrabarti. Malat1: a regulator of inflammatory cytokines in diabetic complications. *Endocrinology, diabetes & metabolism*, 1(2):e00010, 2018. doi: 10.1002/edm2.10.
- James B Grace. *Structural equation modeling and natural systems*. Cambridge University Press, 2006.
- Lauren A Green, Michael R O’Dea, Camden A Hoover, Dana F DeSantis, and Cody J Smith. The embryonic zebrafish brain is seeded by a lymphatic-dependent population of mrc1+ microglia precursors. *Nature Neuroscience*, 25(7):849–864, 2022. doi: 10.1038/s41593-022-01091-9.
- Alex Greenfield, Aviv Madar, Harry Ostrer, and Richard Bonneau. Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PloS one*, 5(10):e13397, 2010. doi: 10.1371/journal.pone.0013397.

- Christopher Heje Grønbech, Maximillian Fornitz Vording, Pascal N Timshel, Casper Kaae Sønderby, Tune H Pers, and Ole Winther. scvae: variational auto-encoders for single-cell gene expression data. *Bioinformatics*, 36(16):4415–4422, 2020. doi: 10.1101/318295.
- Livia KL Habenicht, Zhihua Wang, Xi Zhang, Yuanfang Li, Carolin Mogler, Julia Slotta Huspenina, Roland M Schmid, Christian Weber, Sarajo K Mohanta, Zhe Ma, et al. The c1q-apoe complex: A new hallmark pathology of viral hepatitis and nonalcoholic fatty liver disease. *Frontiers in Immunology*, 13:970938, 2022. doi: 10.3389/fimmu.2022.970938.
- Nellwyn Hagan, John L Kane, Deepak Grover, Lisa Woodworth, Charlotte Madore, Jacqueline Saleh, Jose Sancho, Jinyu Liu, Yi Li, Jonathan Proto, et al. Csf1r signaling is a regulator of pathogenesis in progressive ms. *Cell Death & Disease*, 11(10):904, 2020. doi: 10.2210/pdb6wxj/pdb.
- Timothy R Hammond, Connor Dufort, Lasse Dissing-Olesen, Stefanie Giera, Adam Young, Alec Wysoker, Alec J Walker, Frederick Gergits, Michael Segel, James Nemesh, et al. Single-cell rna sequencing of microglia throughout the mouse lifespan and in the injured brain reveals complex cell-state changes. *Immunity*, 50(1):253–271, 2019. doi: 10.3410/f.734472557.793582092.
- Heonjong Han, Hongseok Shim, Donghyun Shin, Jung Eun Shim, Yunhee Ko, Junha Shin, Hanhae Kim, Ara Cho, Eiru Kim, Tak Lee, et al. Trrust: a reference database of human transcriptional regulatory interactions. *Scientific reports*, 5(1):11432, 2015.
- Heonjong Han, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, et al. Tr-rust v2: an expanded reference database of human and mouse transcriptional regulatory interactions. *Nucleic acids research*, 46(D1):D380–D386, 2018. doi: 10.1093/nar/gkx1013.

Michael Hecker, Sandro Lambeck, Susanne Toepfer, Eugene Van Someren, and Reinhard Guthke. Gene regulatory network inference: data integration in dynamic models—a review. *Biosystems*, 96(1):86–103, 2009. doi: 10.1016/j.biosystems.2008.12.004.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. doi: 10.1016/j.neunet.2019.02.007.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. doi: 10.1109/powertech55446.2023.10202713.

Inge R Holtman, Divya D Raj, Jeremy A Miller, Wandert Schaafsma, Zhuoran Yin, Nieske Brouwer, Paul D Wes, Thomas Möller, Marie Orre, Willem Kamphuis, et al. Induction of a common microglia gene expression signature by aging and neurodegenerative conditions: a co-expression meta-analysis. *Acta neuropathologica communications*, 3:1–18, 2015. doi: 10.1186/s40478-015-0203-5.

Ji-Yeun Hur, Georgia R Frost, Xianzhong Wu, Christina Crump, Si Jia Pan, Eitan Wong, Marilia Barros, Thomas Li, Pengju Nie, Yujia Zhai, et al. The innate immunity protein iftm3 modulates γ -secretase in alzheimer’s disease. *Nature*, 586(7831):735–740, 2020. doi: 10.3410/f.738593874.793578232.

Vân Anh Huynh-Thu and Pierre Geurts. dyngenie3: dynamical genie3 for the inference of gene networks from time series expression data. *Scientific reports*, 8(1):1–12, 2018. doi: 10.1038/s41598-018-21715-0.

Vân Anh Huynh-Thu, Alexandre Irrthum, Louis Wehenkel, and Pierre Geurts. Inferring regulatory networks from expression data using tree-based methods. *PloS one*, 5(9):e12776, 2010. doi: 10.1371/journal.pone.0012776.

Christopher A Jackson, Dayanne M Castro, Giuseppe-Antonio Saldi, Richard Bon-

neau, and David Gresham. Gene regulatory network reconstruction using single-cell rna sequencing of barcoded genotypes in diverse environments. *eLife*, 9: e51254, jan 2020. ISSN 2050-084X. doi: 10.7554/eLife.51254. URL <https://doi.org/10.7554/eLife.51254>.

Philipp Janssen, Zane Kliesmete, Beate Vieth, Xian Adiconis, Sean Simmons, Jamie Marshall, Cristin McCabe, Holger Heyn, Joshua Z Levin, Wolfgang Enard, et al. The effect of background noise and its removal on the analysis of single-cell expression data. *Genome Biology*, 24(1):140, 2023. doi: 10.1101/2022.11.16.516780.

JR Jenkins, K Rudge, and GA Currie. Cellular immortalization by a cDNA clone encoding the transformation-associated phosphoprotein p53. *Nature*, 312(5995): 651–654, 1984.

Marcus Kaiser and Maksim Sipos. Unsuitability of notears for causal graph discovery. *arXiv preprint arXiv:2104.05441*, 2021. doi: 10.1007/s11063-021-10694-5.

Diviyan Kalainathan, Olivier Goudet, Isabelle Guyon, David Lopez-Paz, and Michèle Sebag. Structural agnostic modeling: Adversarial learning of causal graphs. *Journal of Machine Learning Research*, 23(219):1–62, 2022. doi: 10.1109/access.2022.3188395.

Yoonjee Kang, Denis Thieffry, and Laura Cantini. Evaluating the reproducibility of single-cell gene regulatory network inference algorithms. *Frontiers in genetics*, 12:362, 2021. doi: 10.1101/2020.11.10.375923.

Yue Kang, Hongyu Zhang, and Jinting Guan. scinrb: single-cell gene expression imputation with network regularization and bulk rna-seq data. *Briefings in Bioinformatics*, 25(3):bbae148, 2024. doi: 10.1093/bib/bbae148.

Boyd Kenkhuis, Antonios Somarakis, Lorraine de Haan, Oleh Dzyubachyk, Marieke E IJsselsteijn, Noel FCC de Miranda, Boudewijn PF Lelieveldt, Jouke Dijkstra, Willeke van Roon-Mom, Thomas Höllt, et al. Iron loading is a prominent

- feature of activated microglia in alzheimer's disease patients. *Acta neuropathologica communications*, 9(1):1–15, 2021. doi: 10.1101/2021.01.27.428379.
- Seongho Kim. ppcor: an r package for a fast calculation to semi-partial correlation coefficients. *Communications for statistical applications and methods*, 22(6):665, 2015. doi: 10.5351/csam.2015.22.6.665.
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. doi: 10.18653/v1/2020.coling-main.458.
- Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. doi: 10.18178/wcse.2019.06.016.
- Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, 2015. doi: 10.3410/f.725508825.793543120.
- Rex B Kline. *Principles and practice of structural equation modeling*. Guilford publications, 2023. doi: 10.1207/s15328007sem0703_8.
- Velina Kozareva, Caroline Martin, Tomas Osorno, Stephanie Rudolph, Chong Guo, Charles Vanderburg, Naeem Nadaf, Aviv Regev, Wade G Regehr, and Evan Macosko. A transcriptomic atlas of mouse cerebellar cortex comprehensively defines cell types. *Nature*, 598(7879):214–219, 2021. doi: 10.1038/s41586-021-03220-z.
- Karsten Kreis, Ruiqi Gao, and Arash Vahdat. Denoising diffusion-based generative modeling: Foundations and applications. In *CVPR*, 2022.
- Michel Kress, Evelyne May, Roland Cassingena, and Pierre May. Simian virus 40-transformed cells express new species of proteins precipitable by anti-simian virus 40 tumor serum. *Journal of virology*, 31(2):472–483, 1979.

- Teemu D Laajala, Sunil Raghav, Soile Tuomela, Riitta Lahesmaa, Tero Aittokallio, and Laura L Elo. A practical comparison of methods for detecting transcription factor binding sites in chip-seq experiments. *BMC genomics*, 10:1–15, 2009.
- David P Lane and Lionel V Crawford. T antigen is bound to a host protein in sy40-transformed cells. *Nature*, 278(5701):261–263, 1979.
- Sophie Lèbre. Inferring dynamic genetic networks with low order independencies. *Statistical applications in genetics and molecular biology*, 8(1), 2009. doi: 10.2202/1544-6115.1294.
- Hao-Chih Lee, Matteo Danieletto, Riccardo Miotto, Sarah T Cherng, and Joel T Dudley. Scaling structural learning with no-bears to infer causal transcriptome networks. In *Pacific Symposium on Biocomputing 2020*, pages 391–402. World Scientific, 2019. doi: 10.1142/9789811215636_0035.
- Arnold J Levine. p53: 800 million years of evolution and 40 years of discovery. *Nature Reviews Cancer*, 20(8):471–480, 2020.
- Hui Li, Quanwei Zhao, Danan Liu, Bo Zhou, Fujun Liao, and Long Chen. Cathepsin b aggravates atherosclerosis in apoe-deficient mice by modulating vascular smooth muscle cell pyroptosis through $\text{nf-}\kappa\text{b/nlrp3}$ signaling pathway. *Plos one*, 19(1):e0294514, 2024. doi: 10.1371/journal.pone.0294514.
- Wei Vivian Li and Jingyi Jessica Li. An accurate and robust imputation method scimpute for single-cell rna-seq data. *Nature communications*, 9(1):997, 2018. doi: 10.1101/141598.
- Zhi-Ping Liu, Canglin Wu, Hongyu Miao, and Hulin Wu. Regnetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. *Database*, 2015, 2015. doi: 10.1093/database/bav095.
- Po-Ling Loh and Peter Bühlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *The Journal of Machine Learning Research*, 15(1):3065–3105, 2014. doi: 10.1007/978-94-015-9468-4_6.

Karolin Luger, Armin W Mäder, Robin K Richmond, David F Sargent, and Timothy J Richmond. Crystal structure of the nucleosome core particle at 2.8 Å resolution. *Nature*, 389(6648):251–260, 1997. doi: 10.2210/pdb3x1s/pdb.

Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022. doi: 10.1021/acsnano.4c00226.s001.

Juan Ma, Christopher Qian, Yong Bao, Meng-Yue Liu, Hui-Min Ma, Meng-Qi Shen, Wei Li, Jiao-Jiao Wang, Yu-Xin Bao, Yong Liu, et al. Apolipoprotein e deficiency induces a progressive increase in tissue iron contents with age in mice. *Redox Biology*, 40:101865, 2021. doi: 10.1016/j.redox.2021.101865.

James J Manfredi. Mdm2 and mdmx: partners in p53 destruction. *Cancer Research*, 81(7):1633–1634, 2021.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. doi: 10.1007/0-306-47031-4_1. URL <http://www-csli.stanford.edu/~hinrich/information-retrieval-book.html>.

Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. In *BMC bioinformatics*, volume 7-1, pages 1–15. BioMed Central, 2006. doi: 10.3410/f.1030704.368878.

Takahiro Masuda, Lukas Amann, Roman Sankowski, Ori Staszewski, Maximilian Lenz, Paolo d’Errico, Nicolas Snaidero, Marta Joana Costa Jordao, Chotima Böttcher, Katrin Kierdorf, et al. Novel hexb-based tools for studying microglia in the cns. *Nature immunology*, 21(7):802–815, 2020. doi: 10.1038/s41590-020-0707-4.

Hiroataka Matsumoto, Hisanori Kiryu, Chikara Furusawa, Minoru SH Ko, Shigeru BH Ko, Norio Gouda, Tetsutaro Hayashi, and Itoshi Nikaido. Scode:

- an efficient regulatory network inference algorithm from single-cell rna-seq during differentiation. *Bioinformatics*, 33(15):2314–2321, 2017. doi: 10.1101/088856.
- Daniele Mercatelli, Laura Scalambra, Luca Triboli, Forest Ray, and Federico M Giorgi. Gene regulatory network inference resources: A practical overview. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, 1863(6):194430, 2020. doi: 10.1016/j.bbagr.2019.194430.
- Patrick E Meyer, Kevin Kontos, Frederic Lafitte, and Gianluca Bontempi. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP journal on bioinformatics and systems biology*, 2007:1–9, 2007. doi: 10.1155/2007/79879.
- Thomas Moerman, Sara Aibar Santos, Carmen Bravo González-Blas, Jaak Simm, Yves Moreau, Jan Aerts, and Stein Aerts. Grnboost2 and arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics*, 35(12):2159–2161, 2019. doi: 10.1093/bioinformatics/bty916.
- Jamil Momand, Gerard P Zambetti, David C Olson, Donna George, and Arnold J Levine. The mdm-2 oncogene product forms a complex with the p53 protein and inhibits p53-mediated transactivation. *cell*, 69(7):1237–1245, 1992.
- Jason Montojo, Khalid Zuberi, Harold Rodriguez, Farzana Kazi, George Wright, Sylva L Donaldson, Quaid Morris, and Gary D Bader. Genemania cytoscape plugin: fast gene function predictions on the desktop. *Bioinformatics*, 26(22):2927–2928, 2010.
- Achille Nazaret, Justin Hong, Elham Azizi, and David Blei. Stable differentiable causal discovery. *arXiv preprint arXiv:2311.10263*, 2023. doi: 10.2139/ssrn.4837242.
- Ignavier Ng, Shengyu Zhu, Zhitang Chen, and Zhuangyan Fang. A graph autoencoder approach to causal structure learning. *arXiv preprint arXiv:1911.07420*, 2019. doi: 10.1109/tkde.2023.3283425/mm1.

- Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and dag constraints for learning linear dags. *Advances in Neural Information Processing Systems*, 33:17943–17954, 2020. doi: 10.1109/mlsp.2012.6349796.
- Ignavier Ng, Shengyu Zhu, Zhuangyan Fang, Haoyang Li, Zhitang Chen, and Jun Wang. Masked gradient-based causal structure learning. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 424–432. SIAM, 2022. doi: 10.1137/1.9781611977172.48.
- Ignavier Ng, Biwei Huang, and Kun Zhang. Structure learning with continuous optimization: A sober look and beyond. In *Causal Learning and Reasoning*, pages 71–105. PMLR, 2024. doi: 10.55776/sta223.
- Hung Nguyen, Duc Tran, Bang Tran, Bahadir Pehlivan, and Tin Nguyen. A comprehensive survey of regulatory network inference methods using single cell rna sequencing data. *Briefings in bioinformatics*, 22(3):bbaa190, 2021. doi: 10.1093/bib/bbaa190.
- Shinya Oki, Tazro Ohta, Go Shioi, Hideki Hatanaka, Osamu Ogasawara, Yoshihiro Okuda, Hideya Kawaji, Ryo Nakaki, Jun Sese, and Chikara Meno. Chip-atlas: a data-mining suite powered by full integration of public ch ip-seq data. *EMBO reports*, 19(12):e46255, 2018.
- Meryem Ozcencil, Arlinda Dullovi, Romane Catherine Christiane Higos, Zuzana Hořejší, and Roberto Bellelli. The linker histone h1–brca1 axis is a crucial mediator of replication fork stability. *Life Science Alliance*, 6(9), 2023. doi: 10.26508/lsa.202301933.
- Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, pages 1595–1605. Pmlr, 2020. doi: 10.14428/esann/2024.es2024-120.

- Nan Papili Gao, SM Minhaz Ud-Dean, Olivier Gandrillon, and Rudiyanto Gunawan. Sincerities: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles. *Bioinformatics*, 34(2):258–266, 2018. doi: 10.1101/089110.
- Peter J Park. Chip-seq: advantages and challenges of a maturing technology. *Nature reviews genetics*, 10(10):669–680, 2009.
- Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X. doi: 10.1017/cbo9780511803161.007.
- Spyros Pettas, Korina Karagianni, Eirini Kanata, Athanasia Chatziefstathiou, Nikoletta Christoudia, Konstantinos Xanthopoulos, Theodoros Sklaviadis, and Dimitra Dafou. Profiling microglia through single-cell rna sequencing over the course of development, aging, and disease. *Cells*, 11(15):2383, 2022. doi: 10.3390/cells11152383.
- Steven M. Picksley and David P. Lane. What the papers say: The p53-mdm2 autoregulatory feedback loop: A paradigm for the regulation of growth control by p53? *BioEssays*, 15(10):689–690, 1993. doi: <https://doi.org/10.1002/bies.950151008>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/bies.950151008>.
- Jean-Baptiste Pingault, Paul F O’reilly, Tabea Schoeler, George B Ploubidis, Frühling Rijdsdijk, and Frank Dudbridge. Using genetic data to strengthen causal inference in observational research. *Nature Reviews Genetics*, 19(9):566–580, 2018. doi: 10.1038/s41576-018-0020-3.
- Aditya Pratapa, Amogh P Jalihal, Jeffrey N Law, Aditya Bharadwaj, and TM Murali. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods*, 17(2):147–154, 2020. doi: 10.1101/642926.

- Yueyang Qi, Shuangkai Han, Lin Tang, and Lin Liu. Imputation method for single-cell rna-seq data using neural topic model. *GigaScience*, 12:giad098, 2023. doi: 10.1093/gigascience/giad098.
- Savarimuthu Baskar Raj, S Ramaswamy, and Bryce V Plapp. Yeast alcohol dehydrogenase structure and catalysis. *Biochemistry*, 53(36):5791–5803, 2014. doi: 10.1021/acs.jafc.0c00776.s001.
- Jiahua Rao, Xiang Zhou, Yutong Lu, Huiying Zhao, and Yuedong Yang. Imputing single-cell rna-seq data by combining graph convolution and autoencoder neural networks. *Isience*, 24(5), 2021. doi: 10.1101/2020.02.05.935296.
- Andrea Rau, Florence Jaffrézic, Jean-Louis Foulley, and Rebecca W Doerge. An empirical bayesian method for estimating biological networks from temporal microarray data. *Statistical Applications in Genetics and Molecular Biology*, 9(1), 2010. doi: 10.2202/1544-6115.1513.
- Uku Raudvere, Liis Kolberg, Ivan Kuzmin, Tambet Arak, Priit Adler, Hedi Peterson, and Jaak Vilo. g: Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 update). *Nucleic acids research*, 47(W1):W191–W198, 2019. doi: 10.1093/nar/gkz369.
- Alexander Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated dag! causal discovery benchmarks may be easy to game. *Advances in Neural Information Processing Systems*, 34:27772–27784, 2021. doi: 10.1109/icassp49357.2023.10094624.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. doi: 10.1101/2022.11.18.517004.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and*

- Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. doi: 10.1007/978-3-662-54345-0_3.
- Antonella Russo, Eugenia Cordelli, Tullia Salvitti, Elisa Palumbo, and Francesca Pacchierotti. Rad54/rad54b deficiency is associated to increased chromosome breakage in mouse spermatocytes. *Mutagenesis*, 33(4):323–332, 2018. doi: 10.1093/mutage/gey027.
- Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015. doi: 10.1371/journal.pone.0118432.
- Andriy O Samokhin, Paul Ambrose Lythgo, Jacques Yves Gauthier, M David Percival, and Dieter Brömme. Pharmacological inhibition of cathepsin s decreases atherosclerotic lesions in apoe^{-/-}-mice. *Journal of cardiovascular pharmacology*, 56(1):98–105, 2010. doi: 10.1097/fjc.0b013e3181e23e10.
- Manuel Sanchez-Castillo, David Blanco, Isabel M Tienda-Luna, MC Carrion, and Yufei Huang. A bayesian framework for the inference of gene regulatory networks from time and pseudo-time series data. *Bioinformatics*, 34(6):964–970, 2018. doi: 10.1093/bioinformatics/btx605.
- Thomas Schaffter, Daniel Marbach, and Dario Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270, 2011. doi: 10.3410/f.12002958.13156058.
- Marius Schwabenland, Wolfgang Brück, Josef Priller, Christine Stadelmann, Hans Lassmann, and Marco Prinz. Analyzing microglial phenotypes across neuropathologies: a practical guide. *Acta neuropathologica*, pages 1–14, 2021. doi: 10.1007/s00401-021-02370-8.
- Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for

forecasting multiple time series. *arXiv preprint arXiv:2101.06861*, 2021. doi: 10.2139/ssrn.4603436.

Xinwei Shen, Furui Liu, Hanze Dong, Qing Lian, Zhitang Chen, and Tong Zhang. Weakly supervised disentangled generative causal representation learning. *Journal of Machine Learning Research*, 23(241):1–55, 2022. doi: 10.1109/jbhi.2023.3337661/mm1.

Shohei Shimizu, Aapo Hyvarinen, Yutaka Kano, and Patrik O Hoyer. Discovery of non-gaussian linear causal models using ica. *arXiv preprint arXiv:1207.1413*, 2012. doi: 10.1021/acsnano.1c09059.s001.

Bill Shipley. *Cause and correlation in biology: A user’s guide to path analysis, structural equations and causal inference with R*. Cambridge university press, 2016.

Hantao Shu, Jingtian Zhou, Qiuyu Lian, Han Li, Dan Zhao, Jianyang Zeng, and Jianzhu Ma. Modeling gene regulatory networks using neural network architectures. *Nature Computational Science*, 1(7):491–501, 2021. doi: 10.1038/s43588-021-00099-8.

A. Sierksma, A. Lu, R. Mancuso, N. Fattorelli, N. Thrupp, E. Salta, J. Zoco, D. Blum, L. e, B. De Strooper, and M. Fiers. but not to TAU pathology. *EMBO Mol Med*, 12(3):e10606, Mar 2020. doi: 10.1016/s0031-3025(16)36647-8.

Donna K Slonim. From patterns to pathways: gene expression data analysis comes of age. *Nature genetics*, 32(4):502–508, 2002. doi: 10.1038/ng1033.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. doi: 10.32657/10220/46011.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood

- training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428, 2021. doi: 10.21437/eurospeech.1995-21.
- Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72, 1991. doi: 10.1177/089443939100900106.
- Peter Spirtes and Kun Zhang. Causal discovery and inference: concepts and recent methodological advances. In *Applied informatics*, volume 3, pages 1–28. Springer, 2016. doi: 10.1186/s40535-016-0018-x.
- Xiangyu Sun, Oliver Schulte, Guiliang Liu, and Pascal Poupart. Nts-notears: Learning nonparametric dbns with prior knowledge. *arXiv preprint arXiv:2109.04286*, 2021. doi: 10.5220/0011716000003411.
- Damian Szklarczyk, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, et al. String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research*, 47(D1):D607–D613, 2019. doi: 10.1093/nar/gky1131.
- Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L Gable, Tao Fang, Nadezhda T Doncheva, Sampo Pyysalo, et al. The string database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic acids research*, 51(D1):D638–D646, 2023. doi: 10.1093/nar/gkac1000.
- Wenzhuo Tang, Renming Liu, Hongzhi Wen, Xinnan Dai, Jiayuan Ding, Hang Li, Wenqi Fan, Yuying Xie, and Jiliang Tang. A general single-cell analysis framework via conditional diffusion generative models. *bioRxiv*, pages 2023–10, 2023. doi: 10.1101/2023.10.13.562243.
- Biruhalem Taye, Candida Vaz, Vivek Tanavde, Vladimir A Kuznetsov, Frank Eisenhaber, Richard J Sugrue, and Sebastian Maurer-Stroh. Benchmarking selected

- computational gene network growing tools in context of virus-host interactions. *Scientific reports*, 7(1):5805, 2017.
- Roberto Tiribuzi, Antonio Orlacchio, Lucia Crispoltoni, Mariangela Maiotti, Mauro Zampolini, Massimiliano De Angeliz, Patrizia Mecocci, Roberta Cecchetti, Giorgio Bernardi, Alessandro Datti, et al. Lysosomal β -galactosidase and β -hexosaminidase activities correlate with clinical stages of dementia associated with alzheimer’s disease and type 2 diabetes mellitus. *Journal of Alzheimer’s Disease*, 24(4):785–797, 2011. doi: 10.3233/jad-2011-100525.
- Andreas Tjärnberg, Maggie Beheler-Amass, Christopher A Jackson, Lionel A Christiaen, David Gresham, and Richard Bonneau. Structure-primed embedding on the transcription factor manifold enables transparent model architectures for gene regulatory network and latent activity inference. *Genome biology*, 25(1):24, 2024. doi: 10.1101/2023.02.02.526909.
- Caroline Uhler, Garvesh Raskutti, Peter Bühlmann, and Bin Yu. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, pages 436–463, 2013. doi: 10.1214/12-aos1080.
- Sipko Van Dam, Urmo Vosa, Adriaan van der Graaf, Lude Franke, and Joao Pedro de Magalhaes. Gene co-expression analysis for functional classification and gene–disease predictions. *Briefings in bioinformatics*, 19(4):575–592, 2018. doi: 10.3410/f.727201931.793589848.
- Sara van de Geer and Peter Bühlmann. ℓ_0 -penalized maximum likelihood for sparse directed acyclic graphs. *The Annals of Statistics*, 41(2):536 – 567, 2013. doi: 10.1214/13-AOS1085. URL <https://doi.org/10.1214/13-AOS1085>.
- B Vogelstein, S Sur, and C Prives. p53: the most frequently altered gene in human cancers. *Nature Education*, 3(9):6, 2010.
- Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. D’ya like dags? a

- survey on structure learning and causal discovery. *ACM Computing Surveys*, 55(4):1–36, 2022. doi: 10.1145/3527154.
- Dongfang Wang and Jin Gu. Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. *Genomics, proteomics & bioinformatics*, 16(5):320–331, 2018. doi: 10.1101/199315.
- Haolan Wang, Ming Guo, Hudie Wei, and Yongheng Chen. Targeting p53 pathways: mechanisms, structures, and advances in therapy. *Signal transduction and targeted therapy*, 8(1):92, 2023.
- David Warde-Farley, Sylva L Donaldson, Ovi Comes, Khalid Zuberi, Rashad Badrawi, Pauline Chao, Max Franz, Chris Grouios, Farzana Kazi, Christian Tannus Lopes, et al. The genemania prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic acids research*, 38(suppl_2):W214–W220, 2010.
- Dennis Wei, Tian Gao, and Yue Yu. Dags with no fears: A closer look at continuous optimization for learning bayesian networks. *Advances in Neural Information Processing Systems*, 33:3895–3906, 2020. doi: 10.1007/978-1-4842-6150-7_2.
- Cody M Wolfe, Nicholas F Fitz, Kyong Nyon Nam, Iliya Lefterov, and Radosveta Koldamova. The role of apoe and trem2 in alzheimer’s disease—current understanding and perspectives. *International journal of molecular sciences*, 20(1):81, 2018. doi: 10.3390/ijms20010081.
- Huilei Xu, Caroline Baroukh, Ruth Dannenfelser, Edward Y Chen, Christopher M Tan, Yan Kou, Yujin E Kim, Ihor R Lemischka, and Avi Ma’ayan. Escape: database for integrating high-content published data collected from human and mouse embryonic stem cells. *Database*, 2013, 2013. doi: 10.1093/database/bat045.
- Tulasi Yadati, Tom Houben, Albert Bitorina, and Ronit Shiri-Sverdlov. The ins and outs of cathepsins: physiological function and role in disease management. *Cells*, 9(7):1679, 2020. doi: 10.3390/cells9071679.

- Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9593–9602, 2021. doi: 10.1109/cvpr46437.2021.00947.
- E Yonish-Rouach, C Choisy, V Deguin, C Breugnot, and E May. The role of p53 as a transcription factor in the induction of apoptosis. *Behring Institute Mitteilungen*, (97):60–71, 1996.
- Elisheva Yonish-Rouach, Dalia Resnftzky, Joseph Lotem, Leo Sachs, Adi Kimchi, and Moshe Oren. Wild-type p53 induces apoptosis of myeloid leukaemic cells that is inhibited by interleukin-6. *Nature*, 352(6333):345–347, 1991.
- Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *International Conference on Machine Learning*, pages 7154–7163. PMLR, 2019. doi: 10.5121/csit.2022.120611.
- Yue Yu, Tian Gao, Naiyu Yin, and Qiang Ji. Dags with no curl: An efficient dag structure learning approach. In *International Conference on Machine Learning*, pages 12156–12166. Pmlr, 2021. doi: 10.2139/ssrn.4331205.
- Kelly A Zalocusky, Ramsey Najm, Alice L Taubes, Yanxia Hao, Seo Yeon Yoon, Nicole Koutsodendris, Maxine R Nelson, Antara Rao, David A Bennett, Jason Bant, et al. Neuronal apoe upregulates mhc-i expression to drive selective neurodegeneration in alzheimer’s disease. *Nature neuroscience*, 24(6):786–798, 2021. doi: 10.3410/f.740066834.793585890.
- Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):14049, 2017. doi: 10.3410/f.727212325.793533466.
- Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with

- no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31, 2018. doi: 10.1609/aaai.v36i6.20626.
- Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425. Pmlr, 2020. doi: 10.15368/theses.2015.112.
- Heng-Jun Zhou, Li-Qing Wang, Duan-Bu Wang, Jian-Bo Yu, Yu Zhu, Qing-Sheng Xu, Xiu-Jue Zheng, and Ren-Ya Zhan. Long noncoding rna malat1 contributes to inflammatory response of microglia following spinal cord injury via the modulation of a mir-199b/ik κ b/nf- κ b signaling pathway. *American Journal of Physiology-Cell Physiology*, 315(1):C52–C61, 2018. doi: 10.1152/ajpcell.00278.2017.
- W Zhou, SA Scott, SB Shelton, and KA Crutcher. Cathepsin d-mediated proteolysis of apolipoprotein e: possible role in alzheimer’s disease. *Neuroscience*, 143(3): 689–701, 2006. doi: 10.1016/j.neuroscience.2006.08.019.
- Hao Zhu and Donna Slonim. Improving gene regulatory network inference using dropout augmentation. *bioRxiv*, pages 2023–01, 2023. doi: 10.1101/2023.01.26.525733.
- Hao Zhu and Donna K Slonim. From noise to knowledge: probabilistic diffusion-based neural inference of gene regulatory networks. *J Comput Biol*, to appear, 2024. doi: 10.1101/2023.11.05.565675. URL <https://doi.org/10.1101/2023.11.05.565675>.
- Hao Zhu, Di Zhou, and Donna Slonim. Denoising diffusion causal discovery. In *Submitted to The Thirteenth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=Z756zcjNcC>. under review.
- Zhaonan Zou, Tazro Ohta, Fumihito Miura, and Shinya Oki. Chip-atlas 2021 update: a data-mining suite for exploring epigenomic landscapes by fully integrating chip-seq, atac-seq and bisulfite-seq data. *Nucleic acids research*, 50(W1):W175–W182, 2022. doi: 10.1093/nar/gkac199.

Zhaonan Zou, Tazro Ohta, and Shinya Oki. Chip-atlas 3.0: a data-mining suite to explore chromosome architecture together with large-scale regulome data. *Nucleic Acids Research*, page gkae358, 2024.