Carnegie Mellon University
Dietrich College of Humanities and Social Sciences
School of Computer Science
Dissertation

Submitted in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy

Draft as of August 1, 2023

Title: Advances in Statistical Gene Networks
Presented by: Jinjin Tian
Accepted by:
Department of Statistics & Data Science
Machine Learning Department
Readers:

KATHRYN ROEDER , ADVISOR

JING LEI, ADVISOR

ALESSANDRO RINALDO

ANDREJ RISTESKI

WEI CHEN (UNIVERSITY OF PITTSBURGH)

Approved by the Committee on Graduate Degrees:

RICHARD SCHEINES, DEAN DATE

Carnegie Mellon University

Advances in Statistical Gene Networks

Draft as of August 1, 2023

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
in
Statistics and Machine Learning

by

Jinjin Tian

Department of Statistics & Data Science
Machine Learning Department
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213

July 7, 2023

To my family and friends.

Abstract

Gene networks hold immense importance in understanding the underlying mechanisms that govern cellular activities and organismal behavior. As the true gene interaction is not observable, people often resort to observable gene expression data to statistically infer the gene network. In this thesis, we address the immense challenges in the statistical gene network, including 1) benchmark tool for gene network estimation 2) nonlinear gene network estimation methods 3) the application of gene networks in Autism associated gene understanding.

In Chapter 2, we address benchmarking imputation methods on gene coexpression estimation. We develop a new simulation tool that allows realistic simulation of a homogeneous cell group, heterogeneous cell groups, as well as complex cell groups relationships such as tree and trajectory structure, together with gene co-expression structure. We show the usefulness of our tool by accessing the effect of gene expression denoising methods on downstream gene co-expression estimation. In Chapter 3, we address the limitation of current gene co-expression estimation methods in capturing nonlinear relationships. We show that averaging cell-specific gene coexpression over a population gives a novel dependence measure that can detect any non-linear, non-monotone, and non-global relationship. We formally establish the consistency and robustness and demonstrate its advantage over a large family of dependence measures. In Chapter 4, we explore the application of various types of gene networks in a case study of identifying active genes associated with autism spectrum disorders (ASD). To enable a systematic investigation, we also develop a novel gene group interaction measure, which extends an existing idea addressing the challenges when the true gene groups are unknown to nonlinear setups. Using a unified network-assisted gene risk modeling, we found that some types of gene networks are evidently more useful than others for our task: they help identify an assortment of unique "active" and "reactive" gene communities that are biologically interesting.

Acknowledgments

I would like to express my deepest gratitude and appreciation to all those who have supported and contributed to the completion of this PhD thesis.

First and foremost, I am immensely grateful to my advisors, Kathryn Roeder and Jing Lei, for their invaluable guidance, unwavering support, and boundless expertise throughout this research journey. Their mentorship has been instrumental in shaping my academic and personal growth. I am truly fortunate to have had the opportunity to work under their supervision.

I would like to extend my sincere thanks to the members of my thesis committee, Alessandro Rinaldo, Andrej Risteski and Wei Chen for their insightful feedback, constructive criticism, and valuable suggestions that have significantly enhanced the quality of this work.

I am grateful to the Department of Statistics and Data Science, and the Machine Learning department for providing the necessary resources, facilities, and funding that facilitated the smooth execution of this research. The vibrant academic environment and collaborative opportunities within the department have been instrumental in broadening my knowledge and fostering intellectual growth.

I am very fortunate to be surrounded by wonderful peers, these five years have been a great journey with them: special thanks to Addison, Beomjo, David, Mikaela, Nil-Jana, Tim, Tudor, and Sasha.

I am deeply appreciative of my research collaborators, both within the institution and beyond, for the engaging side projects I have had the opportunity to work on with them. Their contributions and shared experiences have enriched the research process and made it more rewarding. Special thanks to Aaditya Ramdas, Jelle Goeman, Eugene Katsevich, Xu Chen, Yuchen Li, Ashwini Pokle, for their support, stimulating discussions, and collaborative efforts. And a very special thanks to Aaditya Ramdas for his exceptional guidance during the early stages of my research career.

I also want to thank all the inspiring talks and discussions during Kathryn's Lab meeting, and the members that have gave me so much inspiration and guidance: special thanks to Bernie Devlin, Jiebiao Wang, Xuran Wang, Zhanrui Cai and Kevin Lin. I will miss those times when we explore science together.

I also would like to thank all my friends who provide both inspiring discussions as well as joy: all the members from the Statistical Queen Association, my wonderful office mates: Maria Jaha, Tiger Zeng, Weichen Wu, friends from MLD: Yusha Liu, Bingbin Liu and Yuchen Li.

I would like to specifically express my heartfelt gratitude to my parents, Yan Cao and Zhihua Tian, for their unwavering support, encouragement, and love throughout this journey. Their belief in me has been a constant source of motivation, and their understanding and sacrifices have been invaluable.

I would like to thank all the individuals who may not be mentioned explicitly but have played a role, no matter how small, in this research endeavor. Their contributions, assistance, and encouragement have been deeply appreciated.

Completing this PhD thesis would not have been possible without the collective support and encouragement of all these individuals. Their contributions have shaped this work and enriched my overall doctoral experience. I am forever grateful for their presence in my life.

Contents

One

Introduction

All figures in this chapter are used only to demonstrate or visualize different concepts needed for the remaining chapters of the thesis.

1.1 Preliminaries

The purpose of this section is to provide a reader with a concise, simplified, and targeted overview of genomics so readers have the necessary biological background to approach the upcoming chapters in this thesis.

It is now well-known that genes play a fundamental role in our existence. The decoding of genes occurs through the process of gene expression. A simplified understanding of the gene expression process is described by the "central dogma", which states that genes are transcribed into messenger RNA (mRNA), which is then translated into proteins. These proteins serve diverse functions within the body, including contributing to the structure, function, and regulation of tissues and organs. Figure 1.1 provides a visual representation of this abstract process. It is important to note that genes do not work in isolation during the gene expression process; rather, they interact during both transcription and translation. The interaction scheme is complex, and unlike other observable interaction systems like social networks, these interactions are not directly observable.

To comprehend the underlying biological factors that influence the gene expression process (as depicted in Figure 1.1), sophisticated data collection methods are necessary. Various laboratory instruments employ different biochemical approaches to investigate distinct aspects of this process, such as DNA sequencing data, RNA-sequencing data (which provides mRNA counts for each gene), and protein data (which provides protein counts for each gene). In this thesis, our focus is on a specific aspect of the process: transcription, which involves the conversion of genes into mRNA. It is widely acknowledged that genes dynamically produce mRNA in a stochastic manner Raj et al. (2006). Each gene exhibits a Poisson-like process with a certain production rate, which can be influenced by other genes (gene interactions), molecules, and experimental conditions. Biochemists often describe this process using Stochastic Differential Equations (SDE), while statisticians tend to model the observed mRNA counts directly using multivariate modeling techniques. It has

been demonstrated that these two approaches align Shahrezaei and Swain (2008): when the stochastic process described by SDE reaches a steady state, the mRNA count follows a negative binomial distribution (a generalized Poisson distribution) marginally. As SDE estimation and inference can be challenging, statistical models have played a crucial role in unraveling the gene expression process. The objective of this thesis is to develop statistical methods that, when applied to RNA-sequencing data, can uncover patterns within the data, particularly gene interaction patterns.

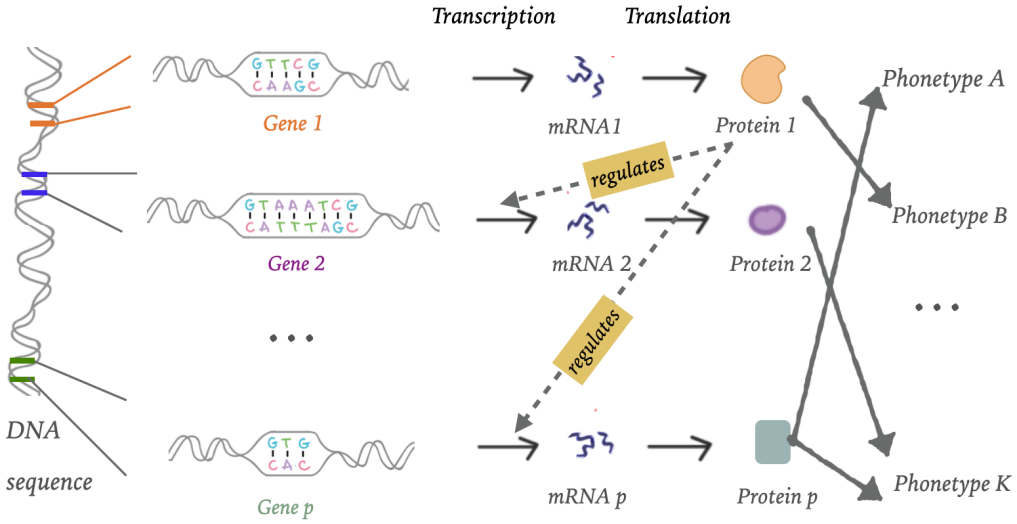


Figure 1.1: A simplified summary of the gene expression process.

1.2 Overview to Chapter 2

In Chapter 2, we extensively explore the current state-of-the-art in generating synthetic RNA-sequencing data. The main objective of this chapter is to develop data simulation software that is both realistic and user-friendly. While numerous software tools already exist for this purpose, we specifically focus on a crucial aspect that has been overlooked for a long time: capturing the interactions among genes during data synthesis. In the following, we briefly introduce the necessary scientific background and overview of our contribution. The work in this chapter resulted in the publication,

> Jinjin Tian, Jiebiao Wang, and Kathryn Roeder. "ESCO: single cell expression simulation incorporating gene co-expression." Bioinformatics 37.16 (2021): 2374-2381.

1.2.1 Scientific background

The two most widely used technologies for generating RNA-sequencing data are bulk RNA-seq and single-cell RNA-seq (scRNA-seq), as illustrated in Figure 1.2. Both methods measure gene expression levels by counting the mRNA produced by genes. However, there are significant differences between the two approaches. In bulk RNA-seq, a population of cells is collected and simultaneously processed, resulting in an averaged measurement of mRNA counts for the entire population. The resulting data matrix is structured in a gene-by-sample format, where each sample represents a bulk of cells. On the other hand, scRNA-seq measures gene expression on a cell-by-cell basis. Each cell is individually processed and analyzed, leading to a data matrix in a gene-by-cell format. scRNA-seq offers a much higher resolution compared to bulk RNA-seq, providing new opportunities for research. However, this increased resolution comes with some challenges. One challenge with scRNA-seq data is the presence of missing counts. A gene may exhibit zero expression in a particular cell due to biological reasons (i.e., the gene is not expressed in that cell) or technical noise (resulting in missing counts). Furthermore, the noise in the non-zero counts can be more pronounced compared to bulk RNA-seq data. To overcome these challenges, various denoising and imputation methods have been proposed to unlock the full potential of scRNA-seq data, particularly in addressing missing counts. However, a key question arises regarding how to evaluate and benchmark these methods since there is no ground truth available. While realistic simulation software has been developed to mimic the biological and data collection processes accurately, many of these tools overlook the importance of gene interactions during the gene expression process.

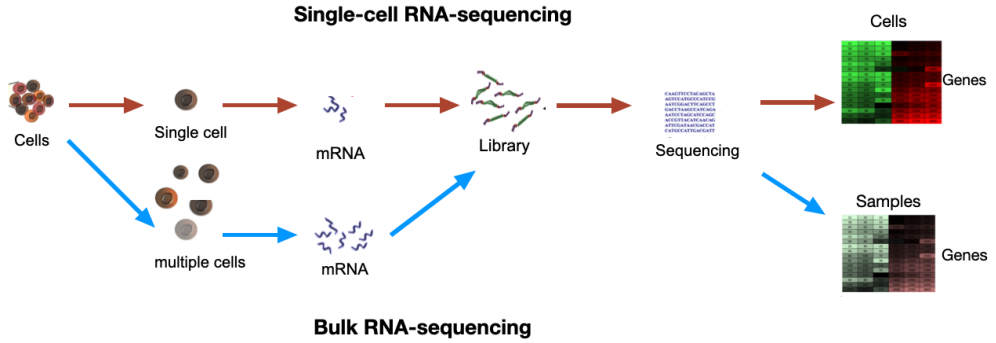


Figure 1.2: A simplified summary of the data collection process in RNA-sequencing.

1.2.2 Contribution overview

In our approach, instead of employing a complex stochastic differential equation framework to model gene interaction, we utilize a copula model to directly incor-

porate gene interaction while leveraging the existing understanding of marginal distributions in scRNA-seq data. This straightforward procedure offers users direct control over the gene interaction scheme and enables efficient generation of large-scale data. We extend the application of copula modeling beyond single cell group simulation to include multiple discrete cell groups, tree-structured cell groups, and continuous cell trajectories. This broadens the scope of our software and facilitates the generation of diverse cell group scenarios. Furthermore, we utilize our software to benchmark existing imputation methods that address the issue of missing counts in scRNA-seq data. Specifically, we evaluate the impact of these methods on downstream gene co-expression estimation, which measures the synchronization of gene expression levels. Surprisingly, we discovered that many methods, despite being effective at fitting the marginal distribution of the data, result in a spurious inflation of gene co-expression. This previously unknown limitation of these methods highlights the importance of careful evaluation. Among the evaluated methods, an ensemble approach demonstrates the most consistent top performance, albeit with increased computation time.

1.3 Overview to Chapter 3

In Chapter 3, our focus is on the estimation of gene co-expression, which captures the synchronization of gene expression levels, particularly for contemporary large heterogeneous scRNA-seq data where gene interactions often exhibit local characteristics. Specifically, the gene interactions can vary from one sample point to another, may only be present in a subset of samples, and can display non-linear or non-monotonic relationships. Existing dependence measures often do not specifically target local dependence relationships, and those that do tend to be computationally intensive. In this chapter, we introduce a novel gene co-expression measure that effectively captures local dependence patterns. This measure has the ability to identify and quantify non-linear or even non-monotonic relationships while providing statistical guarantees. By developing this innovative approach, we are able to address the limitations of previous methods and provide a more comprehensive understanding of gene co-expression in contemporary scRNA-seq datasets.

In the following, we briefly introduce the necessary scientific background and overview of our contribution. The work in this chapter results in the following preprint respectively,

> Jinjin Tian, Jing Lei, and Kathryn Roeder. "From local to global gene co-expression estimation using single-cell RNA-seq data." Biometrics, minor revision.

1.3.1 Scientific background

The true biological networks are of form of a directed network, which describes how a collection of molecular regulators interact with each other and with other

substances in the cell to govern the gene expression levels of mRNA and proteins which, in turn, determine the function of the cell. These networks, called genetic regulatory networks (GRNs), are central to all biological organisms, and their deciphering is crucial to understand the development, functioning, and pathology of these organisms. Once a remote theoretical possibility, this deciphering is now made possible by advances in genomics, most notably high-throughput profiling of gene expression patterns with DNA microarrays and RNA sequencing Karlebach and Shamir (2008); Delgado and Gómez-Vela (2019); Mercatelli et al. (2020); Nguyen et al. (2021). These advances have prompted the development of a plethora of models of GRNs and algorithms to reverse-engineer them from expression data. On one aspect, there are physical models mimicking the biological mechanisms at play, including promoter recognition, mRNA transcription, and protein translation. These models, typically based on systems of ordinary or stochastic differential equations Cao et al. (2012); Dibaeinia and Sinha (2020a), can generate realistic behavior but a large number of experimental data since they tend to have high-dimensional parameter spaces.

Alternatively, statistical models based on the analysis of dependencies between expression patterns (gene co-expressions) offer an intermediate level of complexity and have shown success in aiding the inference of large GRNs. Some methods utilize the bivariate dependency between expression patterns of gene pairs to infer "gene coexpression networks (GCN)" Langfelder and Horvath (2008); Reshef et al. (2011). Although these pairwise gene relationships lack directionality (and causal interpretation), they serve as reliable candidates for subsequent causal structure discovery, which is often computationally demanding Vowels et al. (2022). In this thesis, our focus is on statistical approaches for estimating undirected gene networks (i.e. GCNs), and the inference of directional gene networks (i.e. GRNs) is beyond the scope of our current work.

1.3.2 Contribution overview

In Chapter 3, we leverage a recently proposed ambitious concept: a gene relationship measure at the single-cell level, under the name of cell-specific gene networks. We demonstrate that by averaging the cell-specific gene relationships across a population, we obtain a novel univariate dependence measure called the averaged Local Density Gap (aLDG). This measure effectively accumulates local dependence information and has the capability to detect non-linear and non-monotonic relationships. We establish the robustness of aLDG through a consistent nonparametric estimator that performs well both at the population and empirical levels. Additionally, we explore the application of aLDG in various scenarios by averaging the cell-specific gene relationships over mini-batches defined by external structural information, such as spatial or temporal factors. This approach helps to highlight meaningful local structure change points. We examine the usefulness of aLDG and its minibatch variant in different contexts, including pairwise gene relationship estimation, detection of

bifurcating points in cell trajectories, and visualization of spatial transcriptomics structures. Through simulations and analysis of real data, we demonstrate that aLDG outperforms existing methods, particularly when applied to scRNA-seq data.

1.4 Overview to Chapter 4

In Chapter 4, our focus is on understanding the mechanism of differential expression (DE) in Autism Spectrum Disorder (ASD). While numerous genes that exhibit differential expression between ASD and neurotypical brains have been identified, their precise role in ASD development remains elusive. A gene can be differentially expressed as a causal factor contributing to the phenotype ("active") or as a result of the phenotype itself ("reactive"). Our work represents the first endeavor to comprehensively investigate the DE mechanism in ASD. By delving into this mechanism, we aim to provide fresh insights into the causal relationships and effects associated with ASD from a novel perspective.

1.4.1 Scientific background

Detecting which genes are highly associative of ASD when mutated can help future researchers better understand the genomic basis of ASD as well as design better treatment, but searching across the genome for so-called "autism risk genes" can be extremely timely and costly. A standard analysis to find autism risk genes involves sequencing the genome of trios (an individual with ASD as well as the two parents without ASD) and determining which genes have a mutation in the individual with ASD that leads to severe disruption in how it is expressed (if any). This type of mutation is called de novo loss-of-function (dnLoF) mutations, which provide a great signal-to-noise ratio, but unfortunately, are extremely rare to observe in sequencing data. Some (He et al., 2013; Fu et al., 2022) also pool other forms of genomic information aside from dnLoF mutations to help inferring likely autism risk genes. Still, among thousands of trios sequenced, only a few hundreds of genes were deemed as autism risk genes, and preliminary studies suggest there should be nearly a thousand ASD risk genes (Neale et al., 2012; He et al., 2013). Researchers have also started to investigate the problem from another angle: studying the heterogeneity in gene expression patterns when contrasting ASD and normal samples. Those endeavors represented by Gandal et al. (2022) found over four thousands of genes that are differentially expressed between ASD and normal samples, however, which of them are the cause of ASD and which are affected by ASD remains unknown.

1.4.2 Contribution overview

To unravel the differential expression (DE) mechanism in Autism Spectrum Disorder (ASD), we employ an integrative approach that incorporates information from other sources, specifically the Transmission And De novo Association (TADA) analysis. TADA analysis directly assesses the likelihood of a gene being the cause of ASD

based on DNA sequencing data. Integrating these disparate sources of information is a challenging task, as we observed minimal overlap between genes identified as TADA-significant (genes carrying mutations strongly associated with ASD) and DE-significant genes (genes exhibiting significant differential expression). To bridge this gap, we leverage their shared biological mechanism: gene interactions within a gene network. Firstly, we embark on a systematic investigation of various gene network concepts in the context of modeling ASD risk genes, with a focus on nonlinearity and group interactions. Secondly, we propose a novel model based on Hidden Markov Random Field (HMRF) to jointly model DE and TADA signals, while incorporating gene network information. Our approach enhances the overlap between DE and TADA signals in a meaningful manner by carefully regularizing the signals through "message passing" within the gene network. Through the network regularization of these two sources of information, we successfully identify distinct clusters of "active" and "reactive" DE genes. The active clusters are found to be associated with synaptic and neuronal functions, enriched in neuron-type cells, aligning with the prevailing belief that ASD arises from dysfunctions in neuronal activities. Conversely, the reactive clusters predominantly pertain to responsive functions and are enriched in non-neuronal cells, providing novel insights into the impact of ASD on the misfunctioning of non-neuronal activities. Our findings shed light on the underlying molecular mechanisms of ASD and contribute to a deeper understanding of its complexities.

Two

ESCO: scRNA-seq simulation incorporating gene co-expression

Gene-gene co-expression networks (GCN) are of biological interest for the useful information they provide for understanding gene-gene interactions. The advent of single cell RNA-sequencing allows us to examine more subtle gene co-expression occurring within a cell type. Many imputation and denoising methods have been developed to deal with the technical challenges observed in single cell data; meanwhile, several simulators have been developed for benchmarking and assessing these methods. Most of these simulators, however, either do not incorporate gene co-expression or generate co-expression in an inconvenient manner. Therefore, with the focus on gene co-expression, we propose a new simulator, ESCO, which adopts the idea of the copula to impose gene co-expression, while preserving the highlights of available simulators, which perform well for simulation of gene expression marginally. Using ESCO, we assess the performance of imputation methods on GCN recovery and find that imputation generally helps GCN recovery when the data are not too sparse, and the ensemble imputation method works best among leading methods. In contrast, imputation fails to help in the presence of an excessive fraction of zero counts, where simple data aggregating methods are a better choice. These findings are further verified with mouse and human brain cell data. The ESCO implementation is available as R package `ESCO`[1].

Publication. This work was done in collaboration with Jiebiao Wang and Kathryn Roeder, and contains content that appears in Tian et al. (2021):

> Jinjin Tian, Jiebiao Wang, and Kathryn Roeder. "ESCO: single cell expression simulation incorporating gene co-expression." Bioinformatics 37.16 (2021): 2374-2381.

[1] https://github.com/JINJINT/ESCO

2.1 Introduction

A synchronization between gene expression leads to gene co-expression. Cell heterogeneity, due to cell type or cell cycle, can generate correlations between genes that are highly expressed in similar cells. Alternatively, any form of gene cooperation within a cell type, such as gene co-regulation, also results in co-expression. To differentiate these two settings, we refer them as the gene co-expression across heterogeneous cell groups and gene co-expression within homogeneous cell groups respectively, throughout this article. Understanding gene co-expression in the former setting helps with cell-type identification, and in the latter setting, it helps detect gene regulation relationships and can further provide insights into genetic disorders (Pang et al., 2020; Polioudakis et al., 2019; Parikshak et al., 2013; Willsey et al., 2013b).

Single-cell RNA sequencing (scRNA-seq), a recent breakthrough technology that paves the way for measuring transcription at single cell resolution to study precise biological functions, allows us to target gene co-expression within homogeneous cell groups for the first time. Indeed, early statistical models argued that genes within homogeneous cell groups were independent (Quinn et al., 2018). However, they overlooked the investigations from the biological end, which reveal that correlation arises due to the stochastic nature of gene expression and gene regulation dynamics (Raj et al., 2006).

scRNA-seq data present many challenges for co-expression analysis, due to the sparsity of counts, which include many zeros, mainly arising from low capture and sequencing efficiency in the data collecting process. Sparsity occurs in both a gene- and a cell-specific manner and is observed to have the greatest impact on genes that have low expression. An ever-growing literature attempts to address these challenges using imputation and other denoising methods (Chen et al., 2020; Gong et al., 2018; Huang et al., 2018; Li and Li, 2018; Van Dijk et al., 2018; Eraslan et al., 2019; Linderman et al., 2018). To systemically benchmark these methods, we require realistic simulation tools to construct a ground truth for scRNA-seq data with realistic technical noise; however, currently there is a paucity of methods for this purpose.

Numerous scRNA-seq simulators using both non-parametric and parametric approaches have been proposed during recent years, e.g., Splat (Zappia et al., 2017), SymSim (Zhang et al., 2019a), PROSSTT (Papadopoulos et al., 2019), and SERGIO (Dibaeinia and Sinha, 2020b). Each of those methods focuses on producing realistic marginal behavior of gene expression, and successfully modeling these features, as well as capturing cell type heterogeneity. But, those simulators either ignore gene co-expression, or they generate it in a way that is hard to benchmark. Real data clearly display gene co-expression within homogeneous cell groups (Figure 2.4A) and gene co-expression across heterogeneous cell groups (Figure 2.4B). By contrast, almost all gene pairs show no correlation for simulated data generated using Splat,

even without the challenge of added technical noise (Figure 2.4C). While the data simulated by SymSim may show a modest level of gene co-expression (Figure 2.4D left panel), that correlation arises from the cell type confounding, rather than true gene-gene interaction (Figure 2.4D right panel). PROSSTT, shares a similar issue with SymSim, in that it also introduces co-expression via a random dot product model. SERGIO, on the other hand, directly approximates the biological gene expression process via a series of differential equations with gene regulation relationship as constrains, therefore it is able to introduce gene co-expression based on real gene-gene interactions. However, it is hard to anticipate the final level of co-expression from the imposed gene regulation relationship, hence it is difficult to systematically benchmark the outcome.

Here we propose a new simulation tool, **E**nsemble **S**ingle-cell expression simulator incorporating gene **CO**-expression, ESCO, which is constructed as an ensemble of the best features among current simulators to preserve the marginal performance, while allowing easily incorporating co-expression structure among genes using a copula. Particularly, ESCO allows realistic simulation of a homogeneous cell group, heterogeneous cell groups, as well as complex cell group relationships such as tree and trajectory structure, together with a flexible input of co-expression. As for technical noise, ESCO integrates the parametric and non-parametric approaches in current literature and gives the user flexibility to choose. In order to mimic a specific real data set, ESCO can estimate all the hyperparameters in a feasible way for both a homogeneous cell group or heterogeneous cell groups. ESCO is implemented in the R package `ESCO`, which is built upon the R package `Splatter` (Zappia et al., 2017), in order to provide a unified software framework.

2.2 Methods

2.2.1 Models

Despite their differences, current simulation approaches arguably follow a general flowchart (Figure 2.1). For example, Splat (Zappia et al., 2017) simulates scRNA-seq data using a hierarchical model in which the gamma-Poisson distribution imposes a mean and variance trend; SymSim (Zhang et al., 2019a) is based on a similar hierarchical model with gene kinetics guiding the hyperparameter selection, a non-parametric approach to introduce more realistic noise, and a focus on tree-structured heterogeneity; PROSSTT (Papadopoulos et al., 2019) aims to simulate realistic cell trajectories using a model based on Brownian motion; SERGIO (Dibaeinia and Sinha, 2020b) starts from the gene regulation relationship and solves a series of stochastic differential equations given by gene kinetics to impose those regulations. The more complex non-parametric modeling tends to fit data better than parametric modeling, given that the aim is to mimic data for which the model has already been trained. However, this approach is not practical for producing simulated data similar to a new data set. For example, the non-parametric methods like

SymSim and SERGIO use grid search over a large number of tuning parameters. By contrast, the parametric Splat approach can be tuned to data by fitting a one-step statistical regression model. ESCO also follows the general flowchart in Figure 2.1, but it aims to incorporate the best features from the existing methods. Figure 2.2 illustrates the superiority of ESCO, as it allows simulation of scRNA-seq data with various cell heterogeneity patterns and customized gene co-expression patterns. The correlation pattern input is successfully replicated in the simulated data, both within and between homogeneous cell groups. In this section, we elaborate on the specific simulation models that ESCO adopts, following the framework outlined in Figure 2.1. More detailed descriptions of the simulation models and the time complexity for simulating large complex data are provided in Appendix 2.5.1 and 2.5.4 respectively.

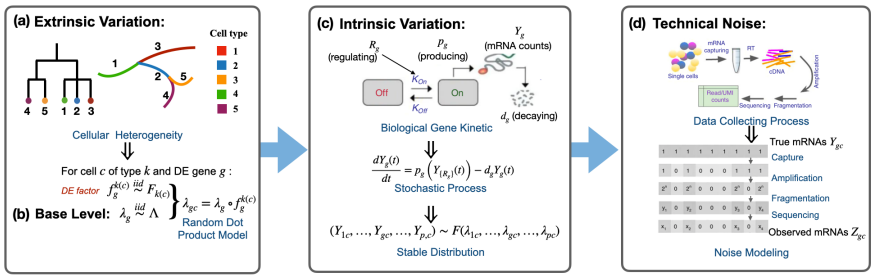
Base expression level. We simulate base expression level in an empirical way that allows inputting any density function, either non-parametric or parametric. Particularly, we denote the base expression level for gene g as λ_g, and we let $\lambda_g \overset{iid}{\sim} \Lambda$ for all g.

Extrinsic variation. The heterogeneity of cell groups is driven by the differential expressed (DE) behavior among certain gene sets across groups. Therefore we implement the cell group heterogeneity, i.e., the extrinsic variation, via modeling the behavior of DE genes (G^{DE}). We use the random dot product model to introduce this heterogeneity by imposing a DE factor generated separately on the otherwise homogeneous gene expression means. Particularly, we generate the different cell group structures we want, via modeling the DE factor f_g^k for gene g in cell group k in each of the following ways.

𝔸. *Discrete cell groups:.* In order to generate clear and distinguishable cell groups, we randomly split the set of DE genes into subsets, each is identified as marker genes for a cell group. Then we simulate the DE factor for each marker gene set as a LogNormal random variable with different mean and variance indexed by group identity.

𝔹. *Tree-structured cell groups:.* We utilize the idea in SymSim (Zhang et al., 2019a), which makes the DE factor of similar cell groups more related to each other. Particularly, we generate the DE factor from a multivariate normal distribution, where the covariance matrix is given by the tree structure of the data. Additionally, in order to assure the identifiability of different cell groups, we introduce extra heterogeneity via strengthening the DE factor for a small proportion of DE genes, which are identified as marker genes in this setting (different from those in the discrete cell group setting).

A.



B.

Statistical estimation Grid search

	Models				
	Biological Truth				Technical Noise (Dropouts)
	Base Level	Extrinsic Variation	Intrinsic Variation	GCN	
Splat	Gamma	fold change	Gamma-Poisson	None	Zero-Inflated
SymSim	Function of gene kinetics related hyper-parameters	Change of low-dim hyper-parameters	Beta-Poisson	None	Down sampling
PROSSTT	Weighted sum of low-dim expression program	Fold change of low-dim expression program	Gamma-Poisson	Yes	None
SERGIO	Expression program guided by GRN	Mean change of master regulators	Poission	Yes	Zero-Inflated
ESCO	Empirical	Mean fold change	Gamma-Poisson	Yes	Zero-Inflated / Down sampling

Figure 2.1: Summary of simulators for scRNA-seq data. **A.** The general modeling flowchart of commonly used simulators. Simulators often start with **(a)** extrinsic variation that arose from cell heterogeneity in the biological sense, and import this model to **(b)** the base expression mean generated for each gene, to formalize the heterogeneous expression means for a gene in a cell of a particular cell type. Then, those means are used to generate the expression level, i.e., mRNA counts, by modeling the **(c)** intrinsic variation, i.e., the stochasticity of gene expression in a cell with a defined base rate of expression. This process is often modeled by the gene kinetic model in biochemistry, which could be stated as a stochastic process in statistical terms. The stable distribution of this stochastic process can usually be approximated by distributions like negative binomial / Poisson / beta Poisson. Finally, some simulators allow the generation of technical noise **(d)** separately, by adding noise, step by step, to the true counts, to mimic the data collection process (the cartoon display is from Zhang et al. (2019a)). Usually, this stepwise process is approximated by the zero-inflation model, where the true counts are set to zero with probability related to expression level. **B.** Summary of the current state of simulators following the general modeling flowchart described above, with blue and orange text color indicating whether they use statistical estimation or grid search when fitting the simulator to a real data set. The objective of ESCO is to create an ensemble of the best features among current simulators in each step, while allowing easily imposing co-expression structure among genes via a copula.

ℂ. *Continuous cell trajectories:.* We utilize the idea in PROSSTT (Papadopoulos et al., 2019), which uses Brownian motion to generate the DE factors, so that the smooth cell heterogeneity can be generated.

Finally, we generate the base expression with an adjustment of library size for each gene g in cell c as

$$\lambda_{gc} = L_c \widetilde{\lambda}_{gc} / \sum_g \widetilde{\lambda}_{gc} \quad \text{for each cell } c, \tag{2.1}$$

where $\log L_c \overset{iid}{\sim} F_L$, and $\widetilde{\lambda}_{gc} \overset{iid}{\sim} \lambda_g f_g^{k(c)}$ if $g \in G^{\text{DE}}$, with $k(c)$ denotes the group identity of cell c; otherwise $\widetilde{\lambda}_{gc} \sim \lambda_g$.

Intrinsic variation.

Marginal distribution:. Gene expression in individual cells is an inherently stochastic process (Raj et al., 2006). If the gene regulation is ignored, this process is just a simple two state birth-death process. The steady-state distribution for this stochastic process in most cases turns out to be a Gamma-Poisson, Beta-Poisson, or Poisson, which is justified from the theoretical biochemistry aspect (Grün et al., 2014; Kim and Marioni, 2013), the experimental data sampling aspect (Quinn et al., 2018), and also the common observations from the data. Splat (Zappia et al., 2017) and PROSSTT (Papadopoulos et al., 2019) utilize the negative binomial model in the simulation of marginal gene expression; while SymSim (Zhang et al., 2019a) uses a Beta-Poisson instead; SERGIO (Dibaeinia and Sinha, 2020b) simulates the gene expression via solving the series of ordinary differential equation functions following the literature about gene kinetics with regulation (Schaffter et al., 2011).

ESCO adopts the negative binomial model, since it is widely accepted in the literature and enjoys support from biochemistry, experimental data sampling, and empirical observations. Particularly, following Splat (Zappia et al., 2017), we can naturally enforce a mean-variance trend by simulating the Biological Coefficient of Variation (BCV) for each gene. BCV is defined as the square root of the standard deviation divided by the mean, i.e., the square root of the coefficient of dispersion. It has been pointed out (McCarthy et al., 2012) that one should not assume a common dispersion for all the genes, as a gene-specific variation is often detected in RNA-seq case studies. Splat simulates BCV as a weighted sum of a common dispersion and a gene-specific dispersion, such that some information can be shared across genes to benefit the estimation, while preserving the gene-specific variation.

Co-expression:. The gene expression (either the truth or the observed) is not necessarily independent even within cells of the same type, resulting from gene regulation. Characterizing the joint distribution requires solving the steady distribution of

multiple correlated stochastic processes, which usually does not have a closed-form solution and requires large computational power (Pratapa et al., 2020; Dibaeinia and Sinha, 2020b). Since the marginal distribution of gene expression is understood fairly well, naturally, we think of using the copula to model the gene dependence. This idea is shown to be successful in Inouye et al. (2017) to model bulk RNA-seq data.

A copula is defined by a joint cumulative distribution function (CDF), $C(u) : [0,1]^p \to [0,1]$ with uniform marginal distributions. One of the most popular copula models is the Gaussian copula, which is defined simply as:

$$C_{\Sigma}^{\text{Gauss}} = N_{\Sigma}(\Phi^{-1}(u_1), \Phi^{-1}(u_2), \ldots, \Phi^{-1}(u_p)) \tag{2.2}$$

where Φ^{-1} denotes the inverse function of standard normal CDF, and N_{Σ} denotes the joint CDF of a multivariate normal random vector with zero means and correlation matrix Σ. Due to the well-known consistency between Σ and the empirical Pearson correlation matrix, the Gaussian copula allows for directly interpretable dependence simulation, and therefore is adopted by ESCO.

Technical noise:. Currently, there are mainly two single cell library preparation protocols: (1) full-length mRNAs profiling without the use of UMIs (e.g., with a standard Smart-Seq protocol); and (2) profiling only the end of the mRNA molecule with the addition of UMIs (e.g., 10x Chromium). The former protocol is usually applied for a small number of cells and with a large number of reads per cell, providing full information on transcript structure. The latter is normally applied for many cells with shallower sequencing, and it is impacted less by amplification and gene length biases. We focus on the UMI-based protocol in this paper because it is usually less biased with greater sparsity.

There currently exist two approaches to simulate the technical noise: one is based on data generating process, and the other is based on data visualization and fitting. As an example of the former, SymSim (Zhang et al., 2019a) uses the empirical approximation of the major steps in the experimental procedures such as mRNA capture, PCR amplification, RNA fragmentation, and sequencing, to directly imitate the technical noise. On the other hand, Splat (Zappia et al., 2017) simulates the technical noise by adopting a zero-inflation model, where the zero-inflation probability relates to the gene expression level in a way that comes from the observed trend in the real data.

There are both pros and cons with regard to these two approaches. The empirical approach facilitates the generation of more realistic noise, but suffers from finding appropriate configuration to match a particular data set (actually, SymSim uses a grid search to do the matching). In contrast, the parametric approach allows a one-step estimation of the parameters from the real data, but can suffer from poor goodness-of-fit due to the mismatch of models. Therefore, ESCO integrates both procedures and gives users the freedom to choose between the two.

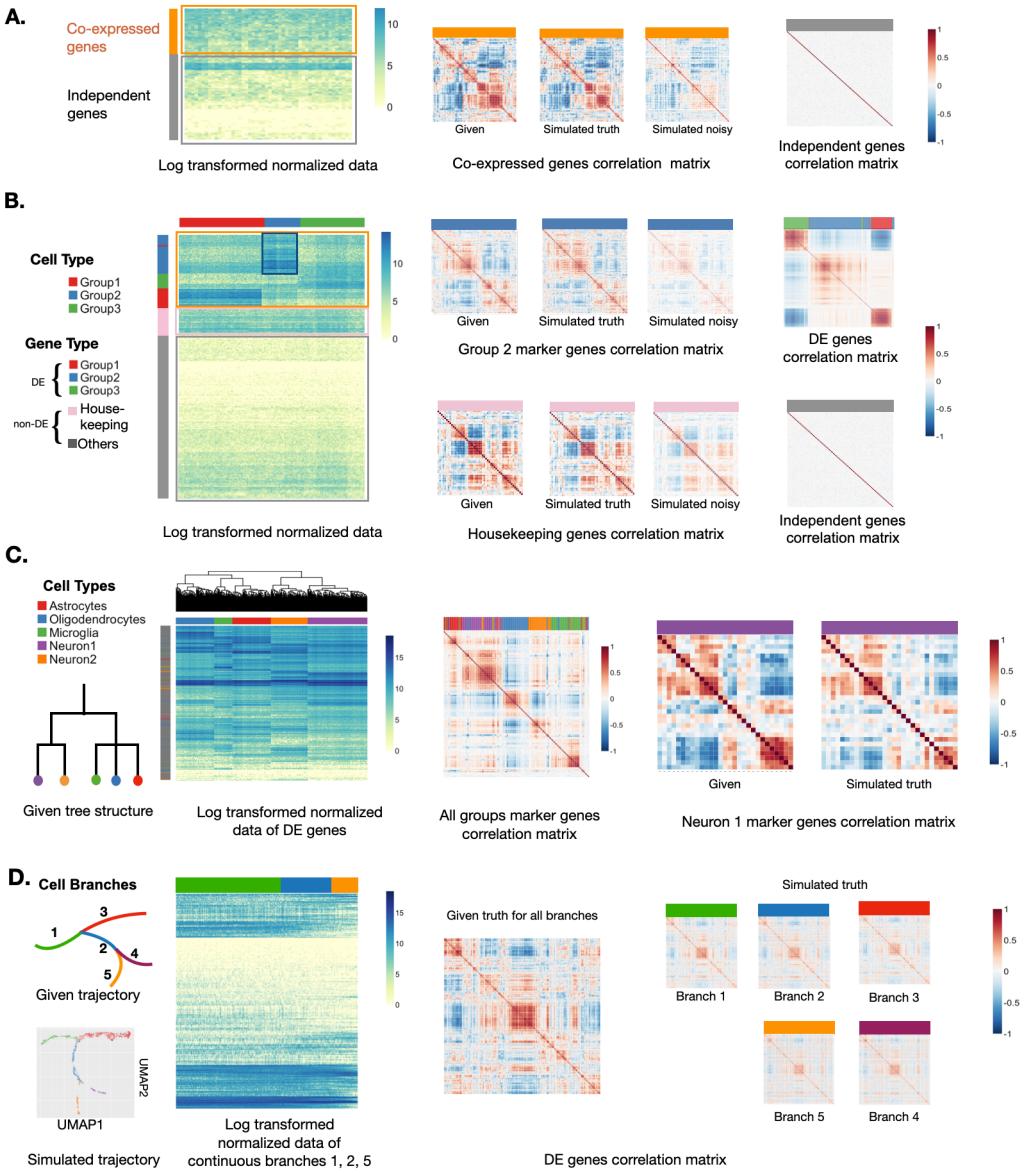


Figure 2.2: (Continued on the following page.)

Figure 2.2: ESCO can simulate scRNA-seq data of various cell heterogeneity and gene co-expression. **A.** The simulation results for one homogeneous cell group consisting of 200 cells and 500 genes. The first panel displays the heatmap of log2 transformed normalized simulated expression data, where rows represent genes and columns represent cells; 30% of genes are chosen to be co-expressed genes, and the rest are independent genes. The following displays depict, in order, the given correlation structure for co-expressed genes, the simulated correlation structure among those co-expressed genes without noise, and that with technical noise, and the simulated correlation structure for independent genes. **B.** The simulation results for three discrete heterogeneous cell groups consisting of 500 cells and 1000 genes. 30% of the genes are chosen to be cell-type DE genes and presumably co-expressed, among which each marks one cell type. Another 10% of genes are chosen to be housekeeping genes, and also presumably co-expressed. The rest are independent non-DE genes. The first display shows the heatmap of log2 transformed normalized simulated data, where different gene types (rows) and cell types (columns) are marked with color bars on the margin. The following displays depict, in order in each row, the given correlation structure for both marker genes of Group2 and co-expressed housekeeping genes, the simulated correlation structure among those co-expressed genes without noise, and that with technical noise; and, at the end of each row the simulated correlation structure among all DE genes across all cells, and that among all independent genes across all cells, with corresponding gene types marked with a color bar on top. **C.** The simulation results for five heterogeneous cell groups that follow a tree structure given in the first panel. We simulate 1000 cells and 2000 genes: 30% of genes are chosen to be DE genes and presumably co-expressed, among which 5% are markers; the rest are independent non-DE genes. The second panel shows the heatmap of log2 transformed normalized simulated data. Different cell types are marked with color bars on the column margin, together with the hierarchical clustering of cells. The following displays depict, in order, the resulting correlation structure among all marker genes across all cells, with corresponding gene types marked with a color bar on top; the given correlation structure for co-expressed marker genes of Neuron1 cells, and the resulting correlation structure among those co-expressed genes. **D.** The simulation results for five heterogeneous cell groups that follow a smooth cell trajectory structure given in the top left panel. There are 1000 cells and 2000 genes; 30% of genes are chosen to be DE genes and presumably co-expressed and share the same correlation structure within each branches, and the rest are independent non-DE genes. The following displays depict, in order, the UMAP for the first two dimensions of the simulated data, the heatmap of log2 transformed normalized simulated data for all DE genes in one continuous path (i.e., branches $1 \to 2 \to 5$), with branch ID marked with a color bar on top; the given shared correlation structure for the DE genes, and the resulting correlation structure simulated of those genes within each branch.

2.2.2 Estimation

ESCO facilitates mimicking any particular data set, consisting of either homogeneous or heterogeneous cell groups, by estimating the hyperparameters from the data. Through learning the parameters in the parametric model, this approach fits data as well as possible, given the limitations of the parametric choice, as illustrated by comparing mouse brain cells (Zeisel et al., 2015) with simulated outcomes. A good match is obtained for mean, variance of expression, UMAP of cells, percent zero outcomes, and co-expression patterns (Figure 2.5).

Next, we elaborate on our specific estimation strategies. Recall that ESCO takes a hierarchical modeling approach, paired with a copula. As such, an empirical Bayesian approach to parameter estimation would be appropriate. However, it is usually infeasible to compute the solution. Therefore, we follow Splat and estimate the parameters in each layer separately. Particularly, we assume the data are already normalized (i.e., no batch effect arises due to technical reason) and have disjoint marker gene sets across cell types, and consider the three estimation tasks in the following.

Estimating the heterogeneity. We have introduced three types of heterogeneity of gene expression (discrete, tree, and trajectory), but we only present an estimation procedure for the discrete one here, and leave a full elaboration of the more complex structure of the other two models to future work. Nevertheless, ESCO is usable for these two models provided the tree structure and trajectory information is available from side information. When the tree/trajectory information is not available, in contrast with SymSim and PROSSTT, we caution against using a grid search to chose model parameters due to the difficulty in determining a good "match" in these complex heterogeneity cases. SymSim and PROSSTT use summary statistics, such as global mean and variance, as standards for a good "match", but two datasets can have similar mean and variance and totally different cell heterogeneity structure.

Following our modeling of the discrete heterogeneous cell groups, we first split all the genes to DE and non-DE genes based on their Area Under the Curve (AUC) scores in cell group prediction using SC3 (Kiselev et al., 2017), provided that we already have the true cell group annotation. Particularly, we use 0.7 as our cutting threshold of the AUC score, i.e., classifying the genes with AUC score no less than 0.7 as DE genes and the others as non-DE genes.

We then use the DE genes to estimate the DE factors. Particularly, we divide those DE genes into marker genes for each cell group based on their classification result from SC3 (Kiselev et al., 2017). We assume that the mean distribution of marker genes in their marked cell group follows the same distribution in the other cell group and a DE factor that follows LogNormal distribution indexed by the cell type. Therefore, we estimate the DE factor for marker genes of cell group k via

fitting a LogNormal distribution on the ratio of their sample mean within cell group k and those outside cell group k.

Estimating the intrinsic variation. First, as for estimating the parameters related to marginal intrinsic variation, we follow the technique used in Splat (Zappia et al., 2017), with a few refinements. We allow non-parametric fitting of the library size distribution and base mean distribution, which can be done quickly by computing the empirical CDF and also later on sampled from using Metropolis-Hastings sampling due to the univariate nature. One may refer to Zappia et al. (2017) for further details about the estimation procedure for other marginal parameters included in the algorithm, such as BCV and outlier.

As for the estimation of the covariance in copula model, we cluster similar cells and form metacells (Baran et al., 2019) first to circumvent challenges due to technical noise and sparse counts. As an integrated version of the original real data, the size of metacells must be carefully selected so that the technical variation can be reduced, while some biological variation can be preserved. We refer the reader to the source paper of MetaCell (Baran et al., 2019) for further details. A more statistically convincing approach would be the non-parametric estimation procedure called SKEPTIC (Liu et al., 2012), which is built for a continuous marginal paired with a Gaussian copula. However, SKEPTIC is derived assuming a continuous marginal without additional noise. In our case, the data are discrete, and the underlying truth is severely masked by the additional zeros, so we find it challenging to recover signals from real data. Therefore, we did not consider this direction, though careful adjustment of the estimation procedure and corresponding consistency under the discrete marginals masked by false zeros is worth attention in future work.

Estimating the technical noise. ESCO also allows estimation of the technical noise when adopting the parametric zero-inflation model. Though Splat already includes the corresponding estimation via fitting a logistic regression between the log-transformed gene mean and their observed zeros proportions, it is biased towards inflating the probability of excess zeros as explained in the Appendix 2.5.2, where we provide a correction of the bias in the end.

2.3 Results

Recall that a particularly prominent aspect of noise that complicates scRNA-seq data analysis is sparsity due to low capture and sequencing efficiency in the data collecting process. Excess sparsity has been shown to corrupt the analysis of scRNA-seq data in many ways (e.g., cell clustering, trajectory inference, DE gene detection, etc.). Imputation methods can generally help according to several benchmarking efforts (Zhang and Zhang, 2018; Andrews and Hemberg, 2018). However, the

influence of sparsity on gene co-expression, particularly within the homogeneous cell group, has been overlooked by many. ESCO provides an easy way to fill in the gap, as it allows for the generation of flexible gene co-expression as a ground truth. In the following we present a systematic evaluation of the performance of imputation methods on the recovery of gene co-expression using ESCO.

2.3.1 Sparsity attenuates the gene co-expression.

First, we show that sparsity indeed impedes the recovery of gene co-expression in scRNA-seq data. Highly expressed genes are much less likely to suffer from technical noise, as they have sufficient replicates to be detected in the data collecting process, in contrast to relatively lowly expressed genes. To illustrate this point we contrast gene co-expression for marker genes in scRNA-seq data (Velmeshev et al., 2019) to bulk RNA-seq data (Parikshak et al., 2016). Genes are classified as high or mid, based on their expression values. In scRNA-seq data, the mid-genes demonstrate substantially less correlation when compared to the high-genes (Figure 2.3A top panel). But in the bulk RNA-seq data, mid and high-genes demonstrate equivalent levels of correlation (Figure 2.3A bottom panel). Because we expect little, if any, impact of technical noise in bulk data, and similar levels of correlation for marker genes in these two data sources, this investigation suggests that sparsity attenuates measured correlation of gene expression in scRNA-seq data. Thus we look to imputation for improved performance.

2.3.2 Imputation can help recover GCN with moderate sparsity.

Working with the Zeisel data (Zeisel et al., 2015), we consider a subset consisting of the 4000 most differentially expressed genes and 526 cells from three cell types (astrocytes_ependymal, endothelial-mural, microglia) that have distinct marker genes. We simulate data from 1000 genes and 200 cells with hyperparameters estimated from the real data, while manually changing the sparsity level such that the zero proportion ranges from 60% to 90% (the real data has $\approx$ 43% zeros). The objective is to recover GCN with greater accuracy by imputing zeros. Success is measured in two ways: improved estimates of gene clustering, based on co-expression networks, and improved identification of pairs of co-expressed genes, based on permutation test of correlation. Comparing the truth to imputation, the former is assessed by computing the Adjusted Rand Index (ARI) and the latter using Area Under the Curve (AUC). We evaluate ARI and AUC for each imputation method under a range of sparsity levels (i.e., zero proportion) for the marker genes within cell groups, the housekeeping genes across cell groups, and the DE genes across cell groups (Figure 2.3B).

To compute ARI we choose the number of clusters that maximizes the score, calculated over a range of clusters numbers (2-9). To calculate AUC we label gene pairs as connected or un-connected based on the co-expression significance in permutation testing of the simulated truth. We then assess the prediction accuracy

(AUC scores) of connections for each imputation method using their estimated co-expression. All the results are averaged over 10 replicates.

We observe the following results (Figure 2.3B). (1) Generally, imputation helps (beat the un-imputed raw data, depicted by the bold dashed black line) recovering both gene co-expression within homogeneous cell groups and gene co-expression across heterogeneous cell groups, but fails to help much with gene co-expression within homogeneous cell groups when facing excessive sparsity ($>$90% zeros), while tends to introduce specious gene co-expression across heterogeneous cell groupswhen facing moderate sparsity ($\sim$ 60-80% zeros). (2) As for a comparison among different methods, there is no universal winner for all settings, but the ensemble method, depicted by the bold black line, provides the best or close to the best performance across almost all settings we considered.

In the following section, we aim to verify our findings of imputation using real scRNA-seq data. It is conjectured that the co-expression of marker genes in the mouse brain will be similar to that of the human brain. Therefore, we expect the recovered gene correlation from a data set measuring mouse brain will follow a similar pattern to those from the data set measuring the human brain. Particularly, we use Zeisel data (Zeisel et al., 2015) for the mouse brain and Velmeshev data (Velmeshev et al., 2019) for the human brain. The Zeisel data have deeper sequencing for single cells and consequently are less noisy, with less sparsity, compared with the Velmeshev data, which have a much greater number of nuclei sampled, each with fewer reads. Therefore, we can see the influence of the sparsity level on gene co-expression by directly comparing these two data sets. We select five common cell types in both data sets and use the Zeisel data as the benchmark. We evaluate the correlation matrix of marker genes before and after imputation of Zeisel data, across cell types and within one cell type (i.e., interneurons). Figure 2.3C(a) plots both the gene co-expression across heterogeneous cell groups and gene co-expression within homogeneous cell groups before and after imputation with EnImpute method (Zhang et al., 2019b) using Zeisel data, while Figure 2.3C(b) plots the same results but using the Velmeshev data. We can see that for the Zeisel data (moderate level of sparsity), imputation enhances the gene co-expression pattern both within homogeneous and across heterogeneous cell groups. In contrast, for the Velmeshev data (excessive sparsity), imputation fails to help much to recover the gene co-expression across heterogeneous cell groups pattern, while failing utterly for gene co-expression within homogeneous cell groups, which is expected, as it is a harder task. This investigation supports some of our findings of imputation, i.e., imputation can generally help, but may fail as sparsity levels increase to a very high level.

2.3.3 Data aggregating is a better way to recover GCN with excessive sparsity.

Despite the excessive sparsity in the Velmeshev data, these data have the advantage of abundant numbers of cells, which inspired us to explore another approach for

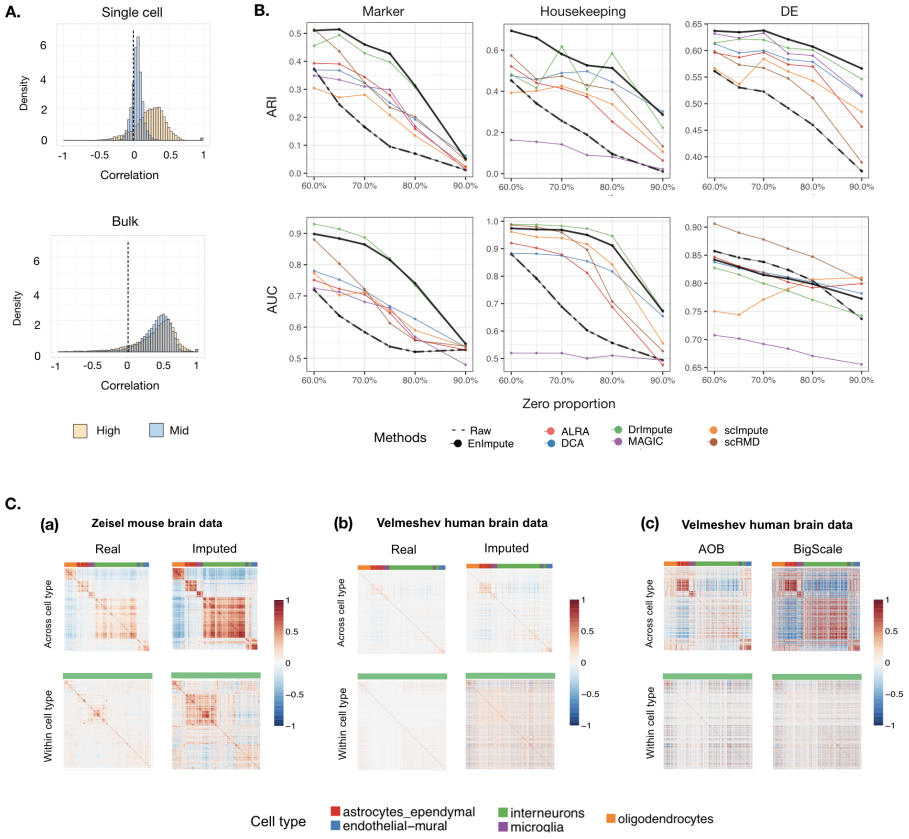


Figure 2.3: Application of ESCO in benchmarking imputation for gene co-expression recovery. **A.** Evidence that sparsity attenuates gene co-expression. The top panel depicts the histogram of Pearson's correlations for the 1000 highest expressed (≈0-10% quantile) genes and 1000 moderately expressed genes (≈60-70% quantile) in Velmeshev scRNA-seq data. The bottom figure depicts the histogram of Pearson's correlations for the same genes as in the top panel, but using the corresponding bulk data. **B.** The performance of different imputation methods on recovering the gene co-expression. We simulate 1000 genes and 200 cells for three cell groups, using the parameters estimated from the Zeisel data, and aggregate the results from 10 replicates. The corresponding ARI score and AUC score (represented by each row) of each imputation method versus different sparsity levels (represented by zero proportion) on different types of gene co-expression (represented by each column, respectively, as marker genes, housekeeping genes, DE genes) are plotted. **C.** Verification of the findings of imputation using real data. **(a)** The correlation matrix of marker genes before and after imputation of Zeisel data, across cell types (five in total) and within one cell type (interneurons). **(b)** The correlation matrix of marker genes before and after the imputation of the Velmeshev data. **(c)** The correlation matrix of marker genes of the Velmeshev data after AOB and BigScale aggregation.

recovering gene co-expression: data aggregation that utilizes the abundance of measured cells. We introduce two methods below, a simple heuristic (AOB) and a complex algorithm (BigSCale).

Averaging over cell bags.. If one has successfully assigned the cell type labels, one may be able to use the simple procedure of averaging gene expression over random splits within cell types, and then compute the gene co-expression based on those averaged values (Polioudakis et al., 2019). We will refer to this procedure as AOB (**A**veraging **O**ver **B**ags). The only tuning parameter here is the bag size, which should be chosen carefully so that we can mitigate the influence of sparsity and other noise, while still maintaining some variability among samples.

Pre-clustering and transforming the expression value.. More recently, a method called BigSCale (Iacono et al., 2019) was developed for the problem of recovering GCN in a similar, but more complex way. This algorithm first clusters cells sharing highly similar transcriptomes together, and then treats them as biological replicates to evaluate the noise and an indirect measure of correlation. This method works well when there is a sufficiently large number of cells for meaningful cell clusters to form, but it is computationally challenging.

We find both methods work well in recovering gene co-expression across heterogeneous cell groups (Figure 2.3C(c)), though neither successfully recover gene co-expression within homogeneous cell groups. Future efforts are needed to recover these subtle signals.

2.4 Discussion

In this paper, we propose a new scRNA-seq simulator, ESCO, which borrows the good features of the current state of art simulators in an ensemble, while for the first time, allowing both interpretable and controllable gene co-expression generation. Specifically, ESCO allows realistic simulation of various cell group structure, ranging from simple homogeneous cell groups to tree-structured discrete cell groups to continuously changing cell trajectories, together with gene co-expression. ESCO outperforms other methods as it preserves the highlights of all the other existing simulators in one R package, including the hierarchical semi-parametric modeling of homogeneous groups from Splat, the tree-structure generation from SymSim, and the trajectory generation from PROSSTT, all while interjecting gene-gene interactions. Specifically, ESCO allows the flexible generation of both gene co-expression across heterogeneous cell groups arising from a cell group structure and gene co-expression within homogeneous cell groups arising from gene-gene interaction in one functional cell group, which have been overlooked and underdeveloped in other methods.

There is still much room for future work in this area. The efficient estimation of the hyperparameters from the real data in the tree-structured cell group and

continuous cell trajectories scenario still needs improvement. Currently, most simulators rely on a grid search of parameters to find parameters that fit a particular data, but these parameter choices do not extend to new settings, and it is extremely challenging to simulate data similar to new data sets. The ability to simulate realistic batch effects in various settings is also not satisfactory in the current methods. ESCO, which mimics Splat in this regard, shares this shortcoming. A careful, deep-dive to produce realistic batch effects is needed.

2.5 Appendices

2.5.1 Model details

Modeling the extrinsic variation.

𝔸. *Discrete cell groups:.* Particularly, denote the set of DE genes as G^{DE}, and the marker gene set $\{G^i\}_{i=1}^K$ for k cell groups such that $G^1 \cup G^2 \cdots \cup G^k \cup \ldots G^K = G^{\mathrm{DE}}$, we let the DE factor for each DE gene g in cell group k be

$$f_g^k = \begin{cases} h_g^k & \text{if } g \in G^k; \\ 1 & \text{otherwise,} \end{cases} \tag{2.3}$$
$$\text{where } \log h_g^k \overset{iid}{\sim} N(\mu_k, \sigma_k).$$

𝔹. *Tree-structured cell groups:.* Specifically, given the similarity between cell groups by a $K \times K$ correlation matrix Σ generated from the tree structure, and a set of DE genes G^{DE}, we firstly select a small proportion of G^{DE} and split them into the marker genes for each group $G^1, G^2, \ldots, G^K$. We let the DE factor for each DE gene g in cell group k be

$$f_g^k = \begin{cases} h_g^k m^k; & \text{if } g \in G^k \\ h_k^g; & \text{otherwise} \end{cases} \tag{2.4}$$
$$\text{where } \ (\log h_g^1, \ldots, \log h_g^K) \overset{iid}{\sim} N(\boldsymbol{z}, \mathrm{diag}\{\sigma_1, \ldots, \sigma_K\}),$$
$$\text{with } \ \boldsymbol{z} := (z_g^1, \ldots, z_g^K) \overset{iid}{\sim} N(\boldsymbol{\mu}, \Sigma),$$

and $m^k > 1$ is a scalar parameter controlling the level of the additional heterogeneity for each group.

ℂ. *Continuous cell trajectories:.* Particularly, for each gene in the DE gene set G^{DE}, we simulate the DE factor at each step t in branch b with length T_b as

$$\begin{aligned}
&\text{for } t = 1, \ldots, T_b: \\
&\quad f_g^{(t,b)} = \exp\left(w_g^{(t,b)}\right), \\
&\qquad \text{where } \; w_g^{(t,b)} = w_g^{(t-1,b)} + v_g^{(t-1,b)} \\
&\qquad \text{with } \; v_g^{(t,b)} = v_g^{(t-1,b)} + N(0,\ 2/T_b). \\
&\text{In particular, we initialize} \\
&\quad v_g^{(0,b)} \sim N(0, \sigma_b); \\
&\quad w_g^{(0,b)} = \begin{cases} 0, & \text{if } p(b) = \emptyset; \\ w_g^{(T_{p(b)}, p(b))}, & \text{otherwise.} \end{cases}
\end{aligned} \tag{2.5}$$

Then, for each branch b, we randomly sample several time points to generate the final cell samples, and let the "group" identity of cell sample c be $k(c) = (t, b)$.

Finally, we generate the base expression with an adjustment of library size for each gene g in cell c as

$$\begin{aligned}
&\lambda_{gc} = L_c \frac{\widetilde{\lambda}_{gc}}{\sum_g \widetilde{\lambda}_{gc}} \quad \text{for each cell } c, \\
&\quad \text{where } \; \widetilde{\lambda}_{gc} \overset{iid}{\sim} \begin{cases} \lambda_g f_g^{k(c)}, & \text{if } g \in G^{\mathrm{DE}}, \\ \lambda_g, & \text{otherwise;} \end{cases} \\
&\quad \text{and } \; \log L_c \overset{iid}{\sim} F_L,
\end{aligned} \tag{2.6}$$

where $k(c)$ denotes the group identity of cell c.

Modeling the intrinsic variation.

𝔸. *Marginal:.* Particularly, we generate the marginal counts $\widetilde{Y}_{gc}$ as:

$$\begin{aligned}
&\widetilde{Y}_{gc} \sim NB\left(\frac{1}{B_{gc}}, \frac{1}{\lambda_{gc} B_{gc}^2 + 1}\right) \\
&\quad \text{where } \; B_{gc} \sim \left(\phi + \frac{1}{\lambda_{gc}}\right) \sqrt{df / \mathcal{X}^2(df)};
\end{aligned} \tag{2.7}$$

where ϕ is the common dispersion parameter, and df represents the degree of freedom of the $\mathcal{X}^2$, and NB represents the Negative Binomial distribution.

𝔹. *Co-expression:.* Recall a copula is defined by a joint cumulative distribution function (CDF), $C(u) : [0,1]^p \to [0,1]$ with uniform marginal distributions. One of the most popular copula models is the Gaussian copula, which is defined simply as:

$$C_{\Sigma}^{\text{Gauss}} = N_{\Sigma}(\Phi^{-1}(u_1), \Phi^{-1}(u_2), \ldots, \Phi^{-1}(u_p)) \tag{2.8}$$

where Φ^{-1} denotes the inverse function of standard normal CDF, and N_{Σ} denotes the joint CDF of a multivariate normal random vector with zero means and correlation matrix Σ.

Then we generate true counts Y_{gc} via the following model:

$$\begin{aligned} Y_{gc} &= NB_{gc}^{-1}\left(\Phi^{-1}(X_{gc})\right) \quad \text{for } g = 1, 2, \ldots, p, \\ &\text{where } \; (X_{1c}, X_{2c}, \ldots, X_{pc}) \sim N(\mathbf{0}, \Sigma); \end{aligned} \tag{2.9}$$

and NB_{gc}^{-1} is the quantile function of the Negative Binomial distribution with parameters indexed by cell c and gene g in equation (2.7), and Σ is the target correlation matrix.

Modeling the technical noise. Particularly, as for the empirical approach from SymSim, one may resort to Zhang et al. (2019a) for details. While as for the parametric approach from Splat, the observed counts Z_{gc} from the data is generated via the following

$$\begin{aligned} Z_{gc} &= Y_{gc}(1 - D_{gc}) \\ &\text{where } \; D_{gc} \sim Ber(\pi_{gc}) \\ &\text{with } \; \pi_{gc} = \frac{1}{1 + \exp\{-k(\log(\lambda_{gc}) - x_0)\}}, \end{aligned} \tag{2.10}$$

where π_{gc} denotes the probability of zero-inflation, given the expression mean λ_{gc}, Ber denotes the Bernoulli distribution, and Z_{gc} denotes the final observed counts.

2.5.2 Estimating the technical noise

ESCO also allows estimation of the median zero-inflation and shape parameters in equation (2.10). Though Splat already includes the corresponding estimation via fitting a logistic regression between the log-transformed gene mean and their observed zeros proportions, it is biased towards inflating the probability of excess zeros, as can be understood via the following reasoning:

Given a real scRNA-seq data set $\mathcal{Z} \in \mathbb{R}^{p \times n}$, where each element Z_{gc} is the observed count of the expression of gene g in cell c, let

$$\pi'_{gc} := \Pr\{Z_{gc} = 0\}. \tag{2.11}$$

Splat estimates π'_{gc} via fitting a logistic function to model the relationship between the log means of the normalized counts and the proportion of cell samples that are

zero for each gene. Then Splat plugs the estimation $\widehat{\pi}_{gc}$ in place of π_{gc} in equation (2.10) to simulate $\widehat{Z}_{gc}$,

$$\widehat{Z}_{gc} = \widehat{Y}_{gc}(1 - \widehat{D}_{gc}), \quad \text{where } \widehat{D}_{gc} \sim Ber(\widehat{\pi}_{gc}). \tag{2.12}$$

and $\widehat{Y}_{gc}$ is the imitation of the true counts Y_{gc} for gene g in cell c simulated in the previous steps.

Assuming the estimation of π'_{gc} is accurate and the simulated true counts $\widehat{Y}_{gc}$ well mimics the real truth Y_{gc}, then this approach would cause more sparsity than expected, since the proportion of zeros in the simulated observation will be

$$\begin{aligned}
\Pr\Big\{\widehat{Z}_{gc} = 0\Big\} &= \Pr\Big\{\widehat{Y}_{gc} = 0\Big\} + \Pr\Big\{\widehat{Y}_{gc} \neq 0, \widehat{D}_{gc} = 1\Big\} \\
&\overset{(*)}{=} \Pr\Big\{\widehat{Y}_{gc} = 0\Big\} + \Pr\Big\{\widehat{Y}_{gc} \neq 0\Big\} \Pr\Big\{\widehat{D}_{gc} = 1\Big\},
\end{aligned} \tag{2.13}$$

where (*) is true since $\widehat{Y}_{gc}$ and $\widehat{D}_{gc}$ are independent once condition on λ_{gc}. Therefore,

$$\begin{aligned}
\Pr\Big\{\widehat{Z}_{gc} = 0\Big\} &= \Pr\Big\{\widehat{Y}_{gc} = 0\Big\} + \Pr\Big\{\widehat{Y}_{gc} \neq 0\Big\} \widehat{\pi}_{gc} \\
&\geq \Pr\Big\{\widehat{Y}_{gc} = 0\Big\} \widehat{\pi}_{gc} + \Pr\Big\{\widehat{Y}_{gc} \neq 0\Big\} \widehat{\pi}_{gc} \\
&= \widehat{\pi}_{gc} = \pi'_{gc} = \Pr\{Z_{gc} = 0\},
\end{aligned} \tag{2.14}$$

From the above calculation, one simple correction for this bias uses:

$$\widetilde{\pi}_{gc} = \frac{\widehat{\pi}_{gc} - \Pr\Big\{\widehat{Y}_{gc} = 0\Big\}}{1 - \Pr\Big\{\widehat{Y}_{gc} = 0\Big\}} \tag{2.15}$$

as the plug-in for equation (2.10). Particularly, ESCO approximates $\Pr\Big\{\widehat{Y}_{gc} = 0\Big\}$ using the CDF of Poisson with mean λ_{gc} at zero.

2.5.3 Supplementary Figure

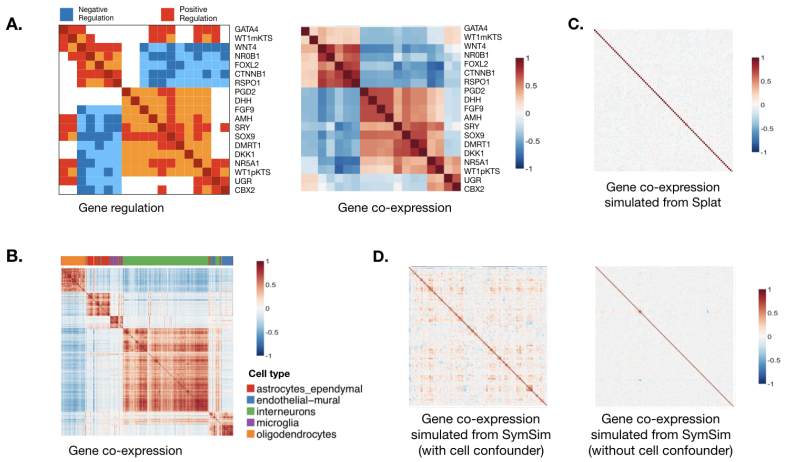


Figure 2.4: Gene co-expression is informative, but we lack satisfactory methods to simulate it for scRNA-seq data. **A.** Connection between gene regulation and gene co-expression. The left panel shows the regulation relationship between the 19 genes in Gonadal Sex Determination (Ríos et al., 2015), while the right panel shows Pearson's correlation matrix for these 19 genes with inferred expression (Pratapa et al., 2020). **B.** Connection between gene co-expression and cell group clusters. The correlation matrix of the 500 most significant marker genes of the five major cell types from the Zeisel data (Zeisel et al., 2015) with corresponding gene types marked with a color bar on top, clustered using hierarchical clustering. **C.** The correlation matrix for 200 simulated genes from Splat (Zappia et al., 2017), without zero-inflation. **D.** The correlation matrix for 200 simulated genes from SymSim (Zappia et al., 2017), without zero-inflation. The left and right panels show results with and without the cell confounding effect, respectively. Specifically, the confounding effect arise as SymSim generates the gene expression for gene g in cell c via a random product model, that is expression $Y_{gc} = \lambda_g \tau_c$, where $\lambda_g \overset{iid}{\sim} F$, and $\tau_c \overset{iid}{\sim} G$. Once conditioning on the cell confounder τ_c, the correlation between expression of genes g_1 and g_2 disappears.

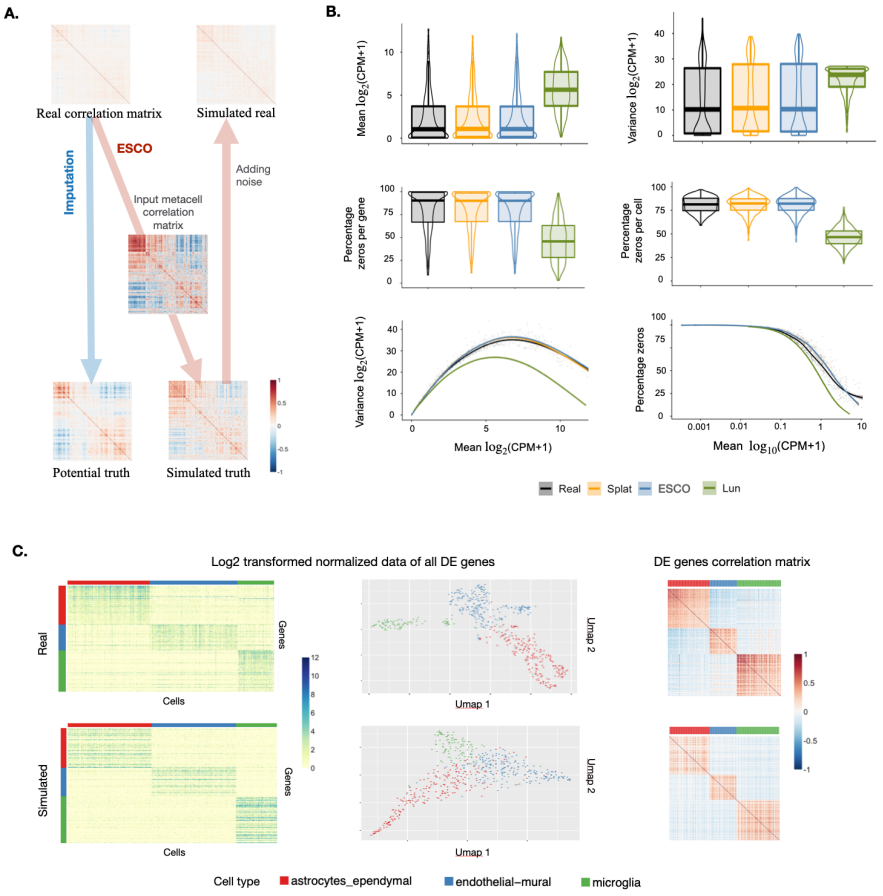


Figure 2.5: ESCO can learn both the cell heterogeneity and gene co-expression from the data. **A.** The generation process of gene co-expression for one homogeneous cell group from real data using ESCO. Particularly, the example is for 500 randomly selected genes in pyramidal CA1 cell type (911 cells) from Zeisel data. **B.** The comparison of marginal features of real data consist of 500 randomly selected genes in pyramidal CA1 cell type (911 cells) extracted from Zeisel data, and the corresponding simulated data using different simulators. Particularly, Lun (Lun et al., 2016) is one of the earliest scRNA-seq simulators, which has been found to be suboptimal (Zappia et al., 2017). We include it here as a clear contrast with the state-of-art methods. **C.** The comparison of real data consist of 4000 most differential expressed genes in three cell types (astrocytes_ependymal, endothelial_mural, microglia) of 526 cells in total extracted from Zeisel data, and the corresponding simulated data using ESCO. While the UMAP depiction differs somewhat, the expression and co-expression patterns match closely.

2.5.4 Supplementary Table

(a) With gene co-expression

(#genes, #cells)	(1000, 300)	(5000, 500)	(10000, 1000)	(15000, 3000)	(20000, 5000)
One group	10.6	17.2	49.8	343.5	1102.8
Discrete groups	15.8	27.5	89.7	458.9	1365.7
Tree structured	17.8	31.2	80.5	454.6	1328.2
Trajectories	16.3	29.5	99.1	452.6	1270.8

(b) Without gene co-expression

(#genes, #cells)	(1000, 300)	(5000, 500)	(10000, 1000)	(15000, 3000)	(20000, 5000)
One group	2.5	8.0	10.4	42.0	94.2
Discrete groups	6.6	12.5	28.0	91.7	184.0
Tree structured	7.4	16.0	34.1	112.7	212.6
Trajectories	7.3	12.9	30.2	101.7	196.5

Table 2.1: Time (seconds) spent of simulating large complex data.

Three

Averaged local density gap

In genomics studies, the investigation of gene relationships often brings important biological insights. Currently, the large heterogeneous datasets impose new challenges for statisticians because gene relationships are often local. They change from one sample point to another, may only exist in a subset of the sample, and can be non-linear or even non-monotone. Most previous dependence measures do not specifically target local dependence relationships, and the ones that do are computationally costly. In this paper, we explore a state-of-the-art network estimation technique that characterizes gene relationships at the single cell level, under the name of *cell-specific gene networks*. We first show that averaging the *cell-specific gene relationship* over a population gives a novel univariate dependence measure, the averaged Local Density Gap (aLDG), that accumulates local dependence and can detect any non-linear, non-monotone relationship. Together with a consistent nonparametric estimator, we establish its robustness on both the population and empirical levels. Then, we show that averaging the *cell-specific gene relationship* over mini-batches determined by some external structure information (e.g. spatial or temporal factor) better highlights meaningful local structure change points. We explore the application of aLDG and its minibatch variant in many scenarios, including pairwise gene relationship estimation, bifurcating point detection in cell trajectory, and spatial transcriptomics structure visualization. Both simulations and real data analysis show that aLDG outperforms existing ones.

Publication. This work was done in collaboration with Jing Lei and Kathryn Roeder, and contains content that appears in Tian et al. (2022):

> Jinjin Tian, Jing Lei, and Kathryn Roeder. "From local to global gene co-expression estimation using single-cell RNA-seq data." Biometrics, minor revision.

3.1 Introduction

Experimental biologists and clinicians seek a deeper understanding of biological processes and their link with disease phenotypes by characterizing cell behavior.

Gene expression offers a fruitful avenue for insights into cellular traits and changes in cellular state. Advances in technology that enable the measurement of RNA levels for individual cells via Single-cell RNA sequencing (scRNA-seq) significantly increase the potential to advance our understanding of the biology of disease by capturing the heterogeneity of expression at the cellular level (Haque et al., 2017). Gene differential expression analysis, which contrasts the marginal expression levels of genes between groups of cells, is the most commonly used mode of analysis to interrogate cellular heterogeneity. By contrast, the relational patterns of gene expression have received far less attention. The most intuitive relational effect is gene co-expression, a synchronization between gene expressions, which can vary dramatically among cells. Converging evidence has revealed the importance of co-expression among genes. When looking at a collection of highly heterogeneous cells, such as cells from multiple cell types, significant gene co-expression may indicate rich cell-level structure. Alternatively, when looking at a batch of highly homogeneous cells, gene co-expression could imply gene cooperation through gene co-regulation (Raj et al., 2006; Emmert-Streib et al., 2014). Biochemistry offers a complementary motivation for the advantages of studying co-expression in addition to marginal expression levels of genes. The biological system of a cell is generally described by a non-linear dynamical system in which gene expression is variable (Raj et al., 2006). Therefore, the observed gene expression level varies by time and condition, even within the same cell, while the cooperation between genes is more stable over time and condition. For this reason, it can be argued that co-expression may more reliably characterize the biological system or state of the cell (Dai et al., 2019). scRNA-seq, allows us to investigate gene co-expression at different resolutions, to understand not only how genes interact with each other within different cells, but also how the interactions relate to cell heterogeneity.

The recent work by Dai et al. (2019) attempts an ambitious task: characterizing the gene co-expression at a single cell level (termed "cell-specific network" CSN). Specifically, for a pair of genes and a target cell, Dai et al. (2019) construct a 2-way 2×2 contingency table test by binning all the cells based on whether they are in the marginal neighborhoods of the target cell and assigning the test results as a binary indicator of gene association in the target cell. Viewed over all gene pairs, the result is a cell-specific gene network. Forgoing interpretation of the detected associations, they utilize the CSN to obtain a data transformation. Specifically, they replace the transcript counts in the gene-by-cell matrix with the degree sequence of each cell-specific network. Although this data transformation shows encouraging success in various downstream tasks, such as cell clustering, it remains unclear what the detected "cell-specific" gene association network really represents. The implementation details and interpretation of the results are presented at a heuristic level, making it difficult for others to appreciate and generalize this line of work.

In a follow-up paper, Wang et al. (2021b) take the first steps to capitalize on the CSN approach by redirecting the concept to obtain an estimator of co-

expression. Specifically, they propose averaging the "cell specific" gene association indicators over cells in a class to recover a global measure of gene association (avgCSN). The resulting measure performs remarkably well in certain simulations and detailed empirical investigations of brain cell data. Compared to Pearson's correlation, the avgCSN gene co-expression appears less noisy and provides more accurate edge estimation in simulations. It is also more powerful in a test to uncover differential gene networks between diseased and control brain cells. Finally, it provides biologically meaningful gene networks in developing cells.

The empirical success of avgCSN likely lies in the nature of gene expression data: often noisy, sparse and heterogeneous, meaning not all cells exhibit co-expression at all times due to cellular state and conditions. For this reason, a successful method must be robust and sensitive to local patterns of dependencies. Being an average of a series of binary local contingency table tests, the error in each entry of avgCSN is limited, meanwhile the non-negative summands ensure that local patterns are not cancelled out. By contrast, measures like Pearson's correlation can have both negative and positive summands, and therefore the final value can be small even if the dependence structure is clear for a subset of the cells. To make the method more stable, Wang et al. (2021b) proposed some heuristic and practical techniques to compute avgCSN, for which we would like to have more principled insights. Examples are the choice of window size in defining neighborhoods in the local contingency table test, the choice of thresholding in constructing an edge, and the range of cells to aggregate over. Many natural questions emerge: how does avgCSN relate to other gene co-expression measures and the full range of general univariate dependence measures, and why does it perform well in practice? Through theoretical analysis and extensive experimental evaluations, we address these questions, revealing that avgCSN is an empirical estimator of a new dependency measure, which enjoys various advantages over the existing measures.

For comparison, we briefly review the related work in gene co-expression measures and general univariate dependence. Since the work by Eisen et al. (1998), Pearson's correlation has been the most popular gene co-expression measure for its simple interpretation and fast computation. However, Pearson's correlation fails to detect non-linear relationships and is sensitive to outliers. Another class of co-expression methods is based on mutual information (MI) (Bell, 1962; Steuer et al., 2002; Daub et al., 2004). The computation of MI involves discretizing the data and tuning parameters, and the dependence measure does not have an interpretable scale. Reshef et al. (2011) proposed the maximal information coefficient (MIC) as an extension of MI, but MIC was shown to be over-sensitive in practice. More comparisons of different co-expression measures and the constructed co-expression networks can be found in Song et al. (2012); Allen et al. (2012).

In the broader statistical literature, the problem of finding gene co-expression is closely related to that of detecting univariate dependence between two random variables. Specifically, for a pair of univariate random variables X, Y, how to

measure the dependence between them has been a long-standing problem. The problem is often described as finding a function $\delta(X, Y)$, which measures the discrepancy between the joint distribution F_{XY} and product of marginal distribution $F_X F_Y$. Numerous solutions to this problem have been provided: include the Renyi correlation (Rényi, 1959) measuring the correlation between two variables after suitable transformations; various regression-based techniques; Hoeffding's D (Hoeffding, 1948), distance correlation (dCor) (Székely et al., 2007), kernel-based measure like HSIC (Gretton et al., 2005) and rank based measure like Kendall's τ and the refinement later, $\tau^\star$ (Bergsma et al., 2014). Most of these methods have not yet been widely adopted in genetics applications.

Aside from avgCSN, the methods mentioned so far do not specifically target dependence relationships that are local and often assume the data are random samples from a common distribution (in contrast with a mixture distribution) in the theoretical analysis. However, real gene interactions may change as the intrinsic cellular state varies and may only exist under specific cellular conditions. Furthermore, with data integration now being a routine approach to combat the curse of dimensionality, samples from different experimental conditions or tissue types are likely to possess different gene relationships and thus create more complex situations for detecting gene interactions. In this setting, much like avgCSN, an ideal measure accumulates subtle local dependencies, possibly only observed in a subset of the cells. A co-expression measure that aims to detect local patterns, developed by Wang et al. (2014), counts the proportion of matching patterns of local expression ranks as the measure of gene co-expression. Specifically, they aggregate the gene interactions across all subsamples of size k. However, despite its promising motivation, it has low power to detect non-monotone relationships. MIC (Reshef et al., 2011) and HHG Heller et al. (2013) are also measures that attempt to account for local patterns of dependencies.

In this paper, we first give a detailed review of the related methods in Section 3.2. Then in Section 3.3.1, we show that avgCSN is indeed an empirical estimate of a valid dependence measure, which we define as averaged Local Density Gap (aLDG). In Section 3.3.2 and Section 3.3.3, we formally establish its statistical properties, including estimation consistency and robustness. We also investigate data-adaptive hyperparameter selection to justify and refine the heuristic choices in application in Section 3.3.4. Finally, we provide a systematic comparison of aLDG and its competitors via both simulation and real data examples in Section 3.5.

3.2 A brief review of dependence and association measures

Before starting on the description of the various dependence measures, let us remark that Rényi (1959) proposed that a measure of dependence between two stochastic variables X and Y, $\delta(X, Y)$, should ideally have the following properties:

(i) $\delta(X, Y)$ is defined for any X, Y neither of which is constant with probability 1.

(ii) $\delta(X, Y)$=$\delta(Y, X)$.

(iii) $0 \leq \delta(X, Y) \leq 1$.

(iv) $\delta(X, Y) = 0$ if and only if X and Y are independent.

(v) $\delta(X, Y) = 1$ if either $X = g(Y)$ or $Y = f(X)$, where f anf g are measurable functions.

(vi) If the Borel-measurable functions f and g map the real axis in a one-to-one way to itself, then $\delta(f(X), g(Y)) = \delta(X, Y)$.

Particularly, a measure satisfying (iv) is called a strong dependence measure.

Apart from the above properties, there are two more properties that are particularly useful in single-cell data analysis. Single-cell data often contain a significant amount of noise, among which outliers account for a non-negligible fraction. Therefore *robustness* is a desirable property in a dependence measure. Specifically, keeping with previous literature (Dhar et al., 2016), by robustness we mean that the value of the measure does not change much when a small contamination point mass, far away from the main population, is added. A formal description and corresponding evaluation metric will be described later. Another often overlooked property is *locality*, which is a relatively novel concept and has not been properly defined to the best of our knowledge. Nevertheless, this concept has been catching attention over the recent decade (Reshef et al., 2011; Heller et al., 2013, 2016; Wang et al., 2014), especially in work motivated by genetic data analysis. *Locality* targets a special kind of dependence relationship that is generally restricted to a particular neighborhood in the sample space. A natural example is dependence that occurs in some, but not necessarily all of the components in a finite mixture. Another is dependence within a moving time window in a time series. Generally speaking, the interactions change as the hidden condition varies, or only exist under a specific hidden condition. A dependence measure that is *local* should be able to accumulate dependence in the local regions.

No measure has all of the properties mentioned above, as far as we know. Our new measure possesses all but properties (v) and (vi). In the following, we review a selected list of univariate dependence measures in more details.

3.2.1 Moment based measures

The first class of methods is based on various moment calculations. The main advantage is fast computation and minimum tuning, while the main drawback is non-robustness to outliers from their moment-based nature.

Pearson's correlation. The simplest measure is the classical Pearson's correlation:

$$\text{Pearson's } \rho(X,Y) := \frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}. \tag{3.1}$$

Plugin the sample estimation of covariance and variance, consistency and asymptotic normality can be proven using law of large numbers and the central limit theorem, respectively. Pearson's ρ has been, and probably still is, the most extensively employed measure in statistics, machine learning, and real-world applications, due to its simplicity. However, it is known to detect only linear relationships. Also, as is the case for regression, it is well known that the product-moment estimator is sensitive to outliers: even just a single outlier may have substantial impact on the measure.

Maximal correlation. The maximal correlation (MC) is based on Pearson's ρ. It is constructed to avoid the problem that Pearson's ρ can easily be zero even if there is strong dependence. Gebelein (1941) first propose MC as

$$\text{MC}(X,Y) := \sup_{f,g} \rho(f(X), g(X)). \tag{3.2}$$

Here the supremum is taken over all Borel-measurable functions f, g with finite and positive variance for $f(X)$ and $g(Y)$. The measure MC can detect non-linear relationships, and in fact, it is a strong dependence measure. However, often MC cannot be evaluated explicitly except in special cases, because there does not always exist functions f_0 and g_0 such that $\text{MC} = \rho(f_0(X), g_0(Y))$. Also, it has been found to be overly "sensitive", i.e. it gives high value for distributions arbitrarily "close" to independence in practice.

Distance correlation. A recent surge of interests has been placed on using distance metrics to achieve consistent independence testing against all dependencies. A notable example is the distance correlation (dCor) proposed by Székely et al. (2007):

$$\begin{aligned} \text{dCor}(X,Y) := \frac{V(X,Y)}{\sqrt{V(X,X)V(Y,Y)}}, \qquad & (3.3) \\ \text{where } V(X,Y) = \mathbb{E}|X-X'||Y-Y'| + \mathbb{E}|X-X'|\mathbb{E}|Y-Y'| & \\ - 2\mathbb{E}_{X,Y}\Big[\mathbb{E}_{X'}|X-X'|\mathbb{E}_{Y'}|Y-Y'|\Big], & \end{aligned}$$

with (X', Y') an i.i.d copy of (X, Y). The distance correlation enjoys universal consistency against any joint distribution of finite second moments; however, in practice, it does not work well for non-monotone relationship (Shen et al., 2020). Also, it is not robust from its moment based nature, as proven by Dhar et al. (2016).

HSIC. Recall the definition and formula for the maximal correlation, about which we mentioned it is difficult to compute since it requires the supremum of the correlation $\rho(f(X), g(Y))$ taken over Borel-measurable f and g. In the framework of reproducing kernel Hilbert spaces (RKHS), it is possible to pose this problem and compute an analogue of MC quite easily. A state-of-the-art method in this direction is the so-called Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2005). Denote the support of X and Y as $\mathcal{X}$ and $\mathcal{Y}$ respectively, HSIC considers f, g to be in RKHS $\mathcal{F}$ and $\mathcal{G}$ of functionals on sets $\mathcal{X}$ and $\mathcal{Y}$ respectively. Then HSIC is defined to be the Hilbert-Schmidt (HS) norm of a Hilbert-Schmidt operator. We refer the reader to Gretton et al. (2005) for detailed description. What might be of interest is that, in many cases, HSIC is equivalent to dCor.

3.2.2 Rank based measures

Another line of work based on ordinal statistics is developed in parallel to the moment-based methods. A random variable X is called ordinal if its possible values have an ordering, but no distance is assigned to pairs of outcomes. Ordinal data methods are often applied to data in order to achieve robustness.

Spearman's ρ_S, Kendall's τ and $\tau^\star$. The two most popular measures of dependence for ordinal random variables X and Y are Kendall's τ and Spearman's ρ_S. Both Kendall's τ and Spearman's ρ_S are proportional to sign versions of the ordinary covariance, which can be seen from the following expressions for the covariance:

$$\begin{aligned}\mathrm{Cov}(X, Y) &= \frac{1}{2}\mathbb{E}\left[(X - X')(Y - Y')\right] \propto \text{Kendall} \\ &= \mathbb{E}\left[(X' - X'')(Y' - Y''')\right] \propto \text{Spearman},\end{aligned}$$

where $(X', Y'), (X'', Y''), (X''', Y''')$ are i.i.d replications of (X, Y). Note that Kendall's τ is simpler than Spearman's ρ_S in the sense that it can be defined using only two rather than three independent replications of (X, Y), so often Kendall's τ is preferred. A concern from certain applications is that Kendall's τ and Spearman's ρ_S are not *strong* dependence measures, so tests based on them are inconsistent for the alternative of a general dependence. In fact, it is often observed that they have difficulty detecting nonmonotone relationship. Later, an extension $\tau^\star$ (Bergsma et al., 2014) mitigates such deficiency by modifying Kendall's τ to a strong measure.

Hoeffding's D and BKR. Related to the ordinal statistics-based methods, another class of methods start from the cumulative distribution function (CDF), some of which are equivalent to ordinal forms due to the relationship between CDF and ranks. The oldest example is the Hoeffing's D proposed by Hoeffding (1948):

$$\text{Hoeffing's D} := \mathbb{E}_{X,Y}\Big[(F_{X,Y} - F_X F_Y)^2\Big],$$

where F_X, F_Y, $F_{X,Y}$ are the CDF of X, Y, (X,Y) respectively. Still, Hoeffing's D is not a strong measure, while its modified version BKR (Blum et al., 1961):

$$\text{BKR} := \mathbb{E}_X \mathbb{E}_Y \Big[(F_{X,Y} - F_X F_Y)^2 \Big]$$

is. It turns out Hoeffding's D belongs to a more general family of coefficients, which can be formulated as

$$\text{C}_{gh} := \int g(F_{X,Y} - F_X F_Y) dh(F_{XY})$$

for some g and h. We will abbreviate Hoeffding's D as HoeffD in the figures in the remainder of paper.

3.2.3 Dependence measures aware of local patterns

Most of the methods mentioned so far do not specifically target dependence relationships that can be local in nature. In the following, we describe a few measures that were designed to capture complex relationships, whether local or not.

Maximal Information Coefficient. The idea behind the Maximal Information Coefficient (MIC,Reshef et al. (2011) statistic consists in computing the mutual information locally over a grid in the data set and then take as statistic the maximum value of these local information measures over a suitable choice of grid. However, several examples were given in Simon and Tibshirani (2014) and Gorfine et al. (2012) where MIC is clearly inferior to dCor.

HHG. Heller et al. (2013) pointed out another way to account for local patterns: that is, looking at dependence locally and then aggregating the dependence over the local regions. The local regions is simply defined as bins via partitioning the sample space. Additionally, HHG takes a multi-scale approach: multiple sample space partitions are conducted, and results are aggregated over all of them. This results in a provably consistent permutation test. However, the cost of implementation is significantly longer computation time than its competitors: it takes $O(n^3)$ computation time while its competitors normally take at most $O(n^2)$.

Matching ranks. Another method that developed specifically for accounting local pattern is proposed by (Wang et al., 2014). Given n pair of observations of (X,Y), $\{(x_i, y_i)\}_{i=1}^n$, they propose to count the number of subsequences of size k: $(x_{i_1}, x_{i_2}, \dots x_{i_k})$ and $(y_{i_1}, y_{i_2}, \dots y_{i_k})$ such that their rank is matched. We refer to this measure as MR (Matching Ranks). Specifically, we write the scaled version of MR such that it is in range [0,1]:

$$\begin{aligned}\text{MR} := \frac{1}{2\binom{n}{k}} \sum_{1 \le i_1 < i_2 \cdots < i_k \le n} \Big(& \mathbf{I}\{rank(x_{i_1}, x_{i_2}, \dots x_{i_k}) = rank(y_{i_1}, y_{i_2}, \dots y_{i_k})\} \\ & + \mathbf{I}\{rank(x_{i_1}, x_{i_2}, \dots x_{i_k}) = rank(-y_{i_1}, -y_{i_2}, \cdots - y_{i_k})\} \Big),\end{aligned}$$

where $rank(a_1, \ldots, a_k) = (r(a_1), \ldots, r(a_k))$ where $r(a_i)$ is the rank of element a_i within the sequences $(a_1, \ldots, a_k)$, and the equality inside the indicator function applies element-wisely. Though claimed to be able to detect complex relationship, this measure is inferior to others in some non-monotone dependence case like quadratic relationship.

3.3 Our method: averaged Local Density Gap

First, we elaborate on the origin of our work, which was inspired by gene co-expression analysis using single-cell data. In the context of gene co-expression analysis, the pair of random variables X, Y represents the expression level of a pair of genes, and the goal is to find the relationship between them. Pearson's correlation is one commonly used metric for this task. In light of the many shortcomings of this global measure of dependence, Dai et al. (2019) proposed to characterize the gene relationships for every cell. Their method takes the following approach: for the gene pair (X, Y), and a target cell j, partition the n samples based on whether $|X_\cdot - X_j| < h_x$ and $|Y_\cdot - Y_j| < h_y$, where h_x and h_y are predefined window sizes. This partition can be summarized as a 2×2 contingency table (Table 3.1). Then evidence against independence in this 2×2 table can be quantified by a general contingency table test statistic. Dai et al. (2019) uses

$$S_{X,Y}^{(j)} := \frac{\sqrt{n}\left(n_{x,y}^{(j)} n - n_{x,\cdot}^{(j)} n_{\cdot,y}^{(j)}\right)}{\sqrt{n_{x,\cdot}^{(j)} n_{y}^{(j)} (n - n_{x,\cdot}^{(j)})(n - n_{\cdot,y}^{(j)})}}, \tag{3.4}$$

and conducts a one-sided α level test based on its asymptotic normality, that is

$$I_{XY}^{(j)} := \mathbb{I}\{S_{X,Y}^{(j)} > \Phi^{-1}(1-\alpha)\}. \tag{3.5}$$

	$\lvert Y_\cdot - Y_j\rvert \le h_y$	$\lvert Y_\cdot - Y_j\rvert > h_y$	
$\lvert X_\cdot - X_j\rvert \le h_x$	$n_{x,y}^{(j)}$		$n_{x,\cdot}^{(j)}$
$\lvert X_\cdot - X_j\rvert > h_x$			
	$n_{\cdot,y}^{(j)}$		n

Table 3.1: The 2×2 contingency table based on distance from j-th sample.

Dai et al. (2019) claim that $I_{XY}^{(j)}$ indicates whether or not gene pairs X and Y are dependent in cell j, and refer to the detected dependence as *local dependence*. Though interesting as a novel concept, it lacks rigor and interpretability. Alternatively we propose to define X and Y as being *locally independent* at position (x, y) as

$$f_{XY}(x, y) = f_X(x) f_Y(y), \tag{3.6}$$

then I_{XY} provides a way of assessing *local independence*. Specifically, as a one-sided test, $I_{XY}(j)$ assesses whether or not $f_{XY}(x,y) > f_X(x)f_Y(y)$, at position (x,y) marked by cell j. To assess global independence, aggregation, as proposed by Wang et al. (2021b), is needed. Their empirical measure can be formally written as:

$$\text{avgCSN} := \frac{1}{n}\sum_{i=1}^{n} I_{XY}^{(j)}. \tag{3.7}$$

Some simple approximations gives us a population correspondence of avgCSN. Assume the variables X, Y have joint density f_{XY}, and marginal densities, f_X and f_Y, that have common support. Let $\widehat{f}_{XY}, \widehat{f}_X, \widehat{f}_Y$ be the estimated densities given observations of (X,Y). Under the assumption that the bandwidth $h_x, h_y \to 0$ and $\sqrt{h_x h_y n} \to \infty$, with some simple algebra (see Appendix 3.7.1 for detailed derivation), we see that

$$\begin{aligned} \text{avgCSN} &\approx \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{ \frac{\widehat{f}_{X,Y}(x_i,y_i) - \widehat{f}_X(x_i)\widehat{f}_Y(y_i)}{\sqrt{\widehat{f}_X(x_i)\widehat{f}_Y(y_i)}} \geq t_n \right\}, \\ &\text{where } t_n = \frac{\Phi^{-1}(1-\alpha)}{\sqrt{nh_xh_y}}, \end{aligned} \tag{3.8}$$

and $\alpha \in [0,1]$ is some hyperparameter related to the test level of the local contingency test (usually α is set to 0.05 or 0.01). Because $t_n \downarrow 0$ as n goes to infinity, we naturally think of the following population dependence measure:

$$\Pr_{X,Y}\left\{ \frac{f_{X,Y}(X,Y) - f_X(X)f_Y(Y)}{\sqrt{f_X(X)f_Y(Y)}} > 0 \right\}.$$

In the remainder of this section, we formally define a generalized version of this measure in Section 3.3.1, along with its properties on the population level. Then we discuss consistent and robust estimation in Section 3.3.3 and provide guidance on hyper-parameter selection in Section 3.3.4. Finally, we comment on the relationship between our measure and some of the previous work in Section 3.3.5.

3.3.1 Definition and basic properties

Definition 3.1. (averaged Local Density Gap) Consider a pair of random variables X, Y whose joint and marginal densities both exist, and denote f_{XY}, f_X, f_Y as their joint and marginal densities. The averaged Local Density Gap (aLDG) measure is then defined as

$$\begin{aligned} \text{aLDG}_t &:= \Pr_{X,Y}\left\{T(X,Y) > t\right\}, \\ &\text{where } T(X,Y) := \frac{f_{X,Y}(X,Y) - f_X(X)f_Y(Y)}{\sqrt{f_X(X)f_Y(Y)}} \end{aligned} \tag{3.9}$$

and $t \geq 0$ is a tunable hyper-parameter.

From the definition, one can immediately realize the following lemma.

Lemma 3.2. For a pair of random variables X, Y whose joint and marginal densities both exist, we have

1. $X \perp Y \iff \text{aLDG}_0 = 0$;
2. if $t > 0$, then $X \perp Y \implies \text{aLDG}_t = 0$;
3. aLDG_t is non-increasing with regard t for all $t \geq 0$;
4. $\text{aLDG}_t \in [0, 1]$;
5. $\text{aLDG}_t(X, Y) = \text{aLDG}_t(Y, X)$;

As a concrete example of the aLDG measure, the left plot of Figure 3.2 displays aLDG, given different t for a bivariate Gaussian with different choices of correlation. We can see that (1) aLDG_t is non-increasing with regard t as our Lemma 3.2 suggests; (2) aLDG_t equals zero at independence for all $t \geq 0$, while aLDG_0 equals zero if and only if there is no dependence, as our Lemma 3.2 suggests; (3) aLDG_t increases with the dependency level, indicating that it is a sensible dependence measure.

Note that, from Lemma 3.2, aLDG_0 is a *strong*[1] measure of dependence. While being strong is a desirable feature of a dependence measure, for aLDG type of measure, we find that it comes with the sacrifice of robustness under independence (Proposition 1). On the other hand, setting $t > 0$ could result in insensitivity under weak dependence, but with a provable guarantee of robustness (Theorem 3.4). In summary, the hyper-parameter t serves as a trade-off between robustness and sensitivity. In Section 3.3.4 we will discuss the practical choice of t in more detail. For now, we treat it as a predefined non-negative constant.

3.3.2 Robustness analysis

In the following, we present a formal robustness analysis. An important tool to measure the robustness of a statistical measure is the influence function (IF). It measures the influence of an infinitesimal amount of contamination at a given value on the statistical measure. The Gross Error Sensitivity (GES) summarizes IF in a single index by measuring the maximal influence an observation could have.

Definition 3.3 (Influence function (IF) and Gross Error Sensitivity (GES)). Assume that the bivariate random variable (X, Y) follows a distribution F, the influence function of a statistical functional R at F is defined as

$$\text{IF}\big((x, y), R, F\big) := \lim_{\epsilon \to 0} \frac{R\big((1-\epsilon)F + \epsilon\delta_{(x,y)}\big) - R(F)}{\epsilon} \tag{3.10}$$

[1]Recall that a measure of dependence between a pair of random variable X, Y is *strong* if it equals zero if and only if X and Y are independent.

where $\delta_{(x,y)}$ is a Dirac measure putting all its mass at (x, y). The Gross Error Sensitivity (GES) summarizes IF in a single index by measuring the maximal influence over all possible contamination locations, which is defined as

$$\mathrm{GES}(R, F) := \sup_{(x,y)} \mid \mathrm{IF}\big((x, y), R, F\big) \mid . \tag{3.11}$$

An estimator is called B-robust if its GES is bounded.

Among the related work we have mentioned, only the robustness of τ, $\tau^\star$, and dCor have been theoretically investigated to the best of our knowledge. Dhar et al. (2016) proved that dCor is not robust while τ and $\tau^\star$ are. Their evaluation criteria is a bit different from ours. We investigate the limit of the ratio when the contamination mass goes to zero. They investigate the ratio limit when the contamination position goes far away, given fixed contamination mass. We argue that our analysis aligns better with the main statistical literature. In the following, we show that aLDG_t with $t > 0$ is B-robust, under some reasonable regularity conditions.

Theorem 3.4. Consider $t > 0$, and a bivariate distribution F of variable (X, Y) whose joint and marginal densities exist as f_{XY}, f_X, f_Y, and satisfy

$$f_{\max} := ||\sqrt{f_X f_Y}||_\infty < \infty; \qquad |\mathrm{aLDG}_{t-\epsilon} - \mathrm{aLDG}_t| \le L\epsilon, \ \forall\ \epsilon > 0; \tag{3.12}$$

then we have

$$\mathrm{GES}(\mathrm{aLDG}_t, F) \le L f_{\max} + 1 < \infty. \tag{3.13}$$

The proof of Theorem 3.4 is in Appendix 3.7.2. The first assumption about the boundness of density is common in density based statistical analysis. The second assumption about the aLDG_t smoothness may look less familiar, however after a transformation, it is no more than a CDF-smoothness assumption: recall that $T(X, Y) := \frac{f_{XY}(X) - f_X(X) f_Y(Y)}{\sqrt{f_X(X) f_Y(Y)}}$, then

$$|\mathrm{aLDG}_{t-\epsilon} - \mathrm{aLDG}_t| < L\epsilon \Longleftrightarrow \mathbb{P}\{|T(X, Y) - t| \le \epsilon\} \le L\epsilon, \tag{3.14}$$

that is, the CDF of random variable $T(X, Y)$ is L-lipschitz around t for $t > 0$. In Figure 3.1 we show the empirical density of $T(X, Y)$ for bivariate Gaussian of different correlation, which is generally bounded by some constant L at positive values.

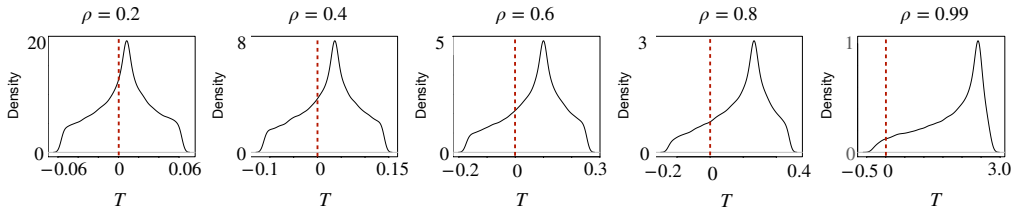


Figure 3.1: The empirical density of statistics T. The underlying bivariate distribution is Gaussian, and the value of T is calculated using the true Gaussian density. We can see that, as the correlation increases, the density of T near zero (annotated by the red dashed line) is smaller.

In the following, we show that aLDG_0 is not robust under independence.

Proposition 1. For any distribution F over a pair of independent random variables (X, Y) whose joint and marginal density exists and are smooth almost everywhere, we have

$$\text{GES}(\text{aLDG}_0, F) = \infty \tag{3.15}$$

if and only if X is independent of Y.

The proof of Proposition 1 is in Appendix 3.7.3. The right plot in Figure 3.2 provides some empirical evidence of the non-robustness of aLDG_0 under independence. Specifically, we plot the population value of the ratio inside limitation (3.10), under bivariate Gaussian with small enough contamination proportion ϵ, to approximately show that the IF value of aLDG_t at independence indeed goes to infinity as t goes to zero.

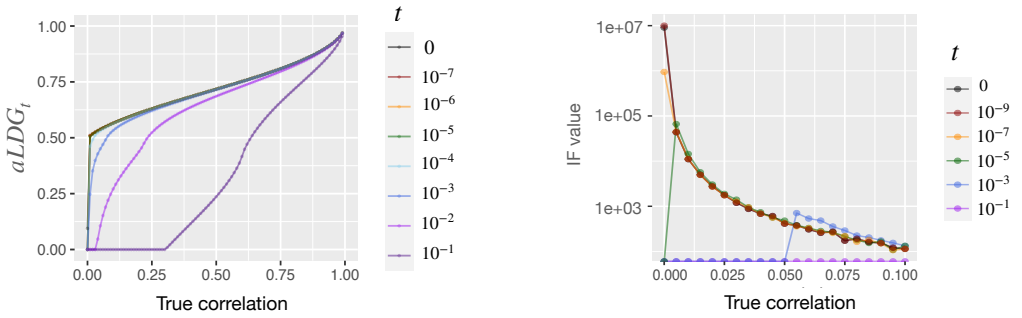


Figure 3.2: **(Left)** The true aLDG_t value for bivariate Gaussian with different levels of correlation under different choices of t. **(Right)** The influence function value approximated by setting the contamination proportion very small ($\epsilon = 10^{-6}$).

3.3.3 Consistent and robust estimation

In this section we investigate estimation of aLDG_t given finite samples. One natural way to estimate aLDG_t is using the following plug-in estimator: recall

that $\widehat{f}_{XY}, \widehat{f}_X, \widehat{f}_Y$ are the estimated joint and marginal densities, then given n observations $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ of (X, Y), aLDG_t can be estimated by

$$\widehat{\mathrm{aLDG}}_t := \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{\widehat{T}(x_i, y_i) \geq t\right\},$$
$$\text{where } \widehat{T}(x_i, y_i) := \frac{\widehat{f}_{X,Y}(x_i, y_i) - \widehat{f}_X(x_i)\widehat{f}_Y(y_i)}{\sqrt{\widehat{f}_X(x_i)\widehat{f}_Y(y_i)}} \tag{3.16}$$

In the following, we establish the non-asymptotic high probability bound of the estimation error using the above simple plug-in estimator $\widehat{\mathrm{aLDG}}_t$. The error rate is determined by the density estimation error for variable X, Y, as well as the probability estimation error for $T(X, Y)$.

Theorem 3.5. Consider $t > 0$, and a bivariate distribution F of variable (X, Y) whose joint and marginal densities exist as f_{XY}, f_X, f_Y, and satisfy

$$\inf_{x,y} f_{XY}(x, y),\ \inf_x f_X(x) \inf_y f_Y(y) \geq c_{\min},$$
$$\sup_{x,y} f_{XY}(x, y),\ \sup_x f_X(x) \sup_y f_Y(y) \leq c_{\max},$$

and for some η_n with $\lim_{n\to\infty} \eta_n \to 0$, with probability at least $1 - \frac{1}{n}$

$$||\widehat{f}_{XY} - f_{XY}||_\infty, ||\widehat{f}_X - f_X||_\infty, ||\widehat{f}_Y - f_Y||_\infty \leq \eta_n; \tag{3.17}$$

and for some constant $0 < L < \infty$,

$$|\mathrm{aLDG}_{t-\epsilon} - \mathrm{aLDG}_t| \leq L\epsilon \quad \text{for all } \epsilon > 0. \tag{3.18}$$

Then we have, with probability at least $1 - \frac{2}{n}$, we have

$$\left|\widehat{\mathrm{aLDG}}_t - \mathrm{aLDG}_t\right| \leq LC\eta_n + \sqrt{\frac{2\log n}{n}}, \tag{3.19}$$

where C depends only on $c_{\min}, c_{\max}$.

Theorem 3.5 is flexible in the sense that one can plug-in any kind of density estimator and its error rate to obtain the error rate of the corresponding $\widehat{\mathrm{aLDG}}$ estimator. The proof of Theorem 3.5 is in Appendix 3.7.4. Though Theorem 3.5 was for fixed t, we also provide similar result that holds true uniformly over all possible t in Appendix 3.6.

As for a concrete example, we provide explicit results for a special class of bivariate density and a simple density estimator. Specifically, we consider the true

marginal density f_X, f_Y that are L-Lipschitz, and the joint density f_{XY} that are simply the product of f_X, f_Y; we also consider the following density estimator[2]:

$$\widehat{f}_X(\cdot) = \frac{1}{n}\sum_{j=1}^{n} K_{h_n}(\cdot, x_j), \quad \widehat{f}_Y(\cdot) = \frac{1}{n}\sum_{j=1}^{n} K_{h_n}(\cdot, y_j),$$
$$\widehat{f}_{XY}(\cdot,\cdot) = \frac{1}{n}\sum_{j=1}^{n} K_{h_n}(\cdot, x_j)K_{h_n}(\cdot, y_j), \tag{3.20}$$

where $K_{h_n}(\cdot, u) := \mathbf{1}\{|\cdot - u| \le h_n\}/(2h_n)$ is one-dimensional boxcar kernel smoothing function with bandwidth h_n. From Proposition 2 in Appendix 3.7.6, the uniform estimation error rate η_n in this setting is $O(n^{-1/6}\sqrt{\log n})$, given the asymptotic near-optimal bandwidth $h = O(n^{-1/6})$. Therefore, applying Theorem 3.5 gives us estimation error rate of $O(n^{-1/6}\sqrt{\log n})$ for aLDG_t.

We also include robustness analysis of $\widehat{\mathrm{aLDG}}_t$ in Appendix 3.11. Specifically, we consider an empirical contamination model that is commonly encountered in single-cell data analysis: a small proportion of the sample points are replaced by "outliers" far away from the rest samples. We show that $\widehat{\mathrm{aLDG}}_t$ with and without outliers are close as long as the outlier proportion is small. This suggests that the estimator of aLDG_t preserves its robust nature.

3.3.4 Selection of hyper-parameter t

In this section, we propose two methods for selecting t, each of which has merit. We also provide guidance on which one is preferable in different practice settings.

Uniform error method. From the results in the previous section, we learn that aLDG_0 is not robust under independence. To prevent $\widehat{\mathrm{aLDG}}_t$ from approaching aLDG_0 under independence, it is sufficient to make sure that the estimation error of T under independence is uniformly dominated by t with high-probability. To compute the uniform estimation error of T under independence, we first manually construct the independence case via random shuffle. Given n samples $\{(x_i, y_i)\}_{i=1}^n$ of (X, Y), denote the corresponding empirical joint distribution as $\widehat{F}_{XY}$, and marginal joint distribution as $\widehat{F}_X$ and $\widehat{F}_Y$. Applying the random shuffle function π on indices of one dimension (i.e. Y), we have

$$\{(x_i, y_{\pi(i)})\}_{i=1}^n \sim \widehat{F}_X \widehat{F}_Y, \tag{3.21}$$

that is the shuffled samples $\{(x_i, y_{\pi(i)})\}$ now come from a different joint distribution where (X, Y) are independent.

[2]The density estimator used here is not chosen to be minimax optimal. We instead design it to align the best with the practical methods Dai et al. (2019) and Wang et al. (2021b), such that we can better justify and refine their heuristic choices of hyperparameter by theory.

We can then use the shuffled samples to compute the uniform estimation error of T under independence. Note that T under independence is exactly zero, therefore its uniform estimation error is just the uniform upper bound of its estimation. To stabilize the estimation of such upper bound, we use the median of estimated upper bound from $\max\{\lfloor 1000/n \rfloor, 5\}$ different random shuffles as the final estimation. We call this t selection method the *uniform error* method.

Asymptotic norm method. When using aLDG_t in large-scale data analysis, choosing t using the above data-dependent choice may be undesirable because it requires additional computations. In extensive simulations we observe that a simple alternative also performs fine in terms of maintaining consistency, power and robustness:

$$t = \Phi^{-1}\left(1 - \frac{1}{n}\right) \Big/ \left(\sqrt{\sigma_X \sigma_Y} n^{1/3}\right). \tag{3.22}$$

This choice is motivated by the following heuristic. Recall our derivation of aLDG statistics from avgCSN around (3.8): as the sample size n goes to infinity, and $h_x, h_y \to 0$, $h_x h_y n \to \infty$, the empirical estimation of aLDG_t using the boxcar kernel cioncide with avgCSN. Therefore, t_n in (3.8) could serve as a natural choice for t, but one need to be extra careful about α, which is the test level of local contingency test (3.5) in definition towards avgCSN. We specically modify α to decrease with n instead of a fixed value like 0.05 since we desire consistency: i.e. aLDG_t under independence should goes to zero as n goes to infinity. Finally, plugging in our choice of bandwidth $h_x = \sigma_X n^{-1/6}$, $h_y = \sigma_Y n^{-1/6}$ together with the new α_n in place of α into t_n (3.8), we get (3.22). We call this t selection method the *asymptotic norm* method.

Empirically we find that the *asymptotic norm* method is often too conservative given the small sample size (which is expected since it is based on the asymptotic normality of a contingency table test statistic). In practice, we recommend people use *uniform error* over *asymptotic norm* when the sample size is not too big (e.g., no bigger than 200). When the sample size is big enough (e.g., bigger than 200), and the computation budget is limited, we recommend the *asymptotic norm* method. In the rest of the paper, we use the *uniform error* method when the sample size is no bigger than 200 and the *asymptotic norm* method when the sample size is bigger than 200. We admit that there could be other promising ways of selecting t, for example, a geometry way we provided in Appendix 3.7.8. Here we only present the methods that we found working the best after a careful evaluation (see Appendix 3.7.8).

3.3.5 Relationships to HHG

The method that is most similar to aLDG is HHG (Heller et al. (2013)). Like aLDG, HHG (Heller et al., 2013) is based on aggregation of multiple contrasts between the local joint and marginal distributions

$$HHG := \sum_{i \neq j} M(i,j), \quad M(i,j) := (n-2) \frac{\Big(p_{XY}(B_{XY}^{i,j}) - p_X(B_X^{i,j}) p_Y(B_Y^{ij})\Big)^2}{p_X(B_X^{i,j})\Big(1 - p_X(B_X^{i,j})\Big) p_Y(B_Y^{ij})\Big(1 - p_Y(B_Y^{ij})\Big)},$$

with $B_X^{i,j} = \{x : |x - x_i| \leq |x_i - x_j|\}$, $B_Y^{i,j} = \{y : |y - y_i| \leq |y_i - y_j|\}$ and $B_{XY}^{i,j} = B_X^{i,j} \otimes B_Y^{i,j}$, p_{XY}, p_X, p_Y are joint probability function for (X, Y) and marginal probability function for X and Y respectively. While the two measures appear quite similar, they differ in two critical aspects.

The efficiency of single scale bandwidth. One notable difference between HHG and aLDG is that the former relies on a multi-scale choice of bandwidth for each sample point. Specifically, it utilizes multiple ($O(n)$) bandwidths for each data point. This results in a provably consistent permutation test; however, the cost of implementation is significantly longer computation time than its competitors. aLDG takes a single-scale approach, which considerably improves the computation efficiency. Moreover, the aLDG formulation provides a direct analogy to a density functional, which allows us to exploit existing work in density estimation to determine an appropriate bandwidth. This single-scale approach, though may not optimal, achieves comparable power to HHG, as shown in the upcoming simulation studies.

The merit of thresholding. Another difference is that empirically aLDG aggregates over thresholded summands, see (3.16). It turns out thresholding brings implicit robustness to noise. By contrast, consider the non-thresholded version of aLDG:

$$\text{aLDG}_{non} := \mathbb{E}\left[T(X,Y)\right]. \tag{3.23}$$

Even with slight departures from independence, aLDG_{non} can go to infinity. For example, consider the following joint and marginal distribution that admits a kernel product density mixture:

$$f_{XY}(x,y) = \alpha k_{0,r}(x) k_{0,r}(y) + (1-\alpha) k_{0,1}(x) k_{0,1}(y),$$
$$f_X(x) = \alpha k_{0,r}(x) + (1-\alpha) k_{0,1}(x), \quad f_Y(y) = \alpha k_{0,r}(y) + (1-\alpha) k_{0,1}(y)$$

where $\alpha \in (0,1)$, $0 < r \ll 1$ and $k_{\mu,r}(\cdot) := \frac{1}{r} k(\frac{\cdot - \mu}{r})$, with k as the density of 1-dim uniform distribution supported on $[-1,1]$.

Note that as $\alpha \to 0$ and $r \to 0$, the model is essentially an independence case contaminated with a small point mass. Additionally with $\alpha/r \to \infty$, we can show that (see Appendix 3.7.9 for details)

$$\mathbb{E}\left[T(X,Y)\right] \approx \frac{\alpha}{r} \to \infty, \tag{3.24}$$

that is the non-thresholded version of aLDG is very large under such simple case of small departure from independence, therefore is problematic. With thresholding, however, aLDG is guaranteed to be approximately α, which goes to zero for small perturbations, as one would desire.

3.4 Minibatched LDG: local relational structure

In many cases, a special structure emerges between cellular states. For example, a smooth transition where individual cells represent points along a continuum or lineage; or a spatial graph where cell states represent nodes in a graph. Cells in these cases change states by undergoing gradual transcriptional changes that are controlled by an underlying temporal or spatial factor.

The majority of the work in structured genetic data analysis focuses on marginal characterization, while higher-order perspectives like gene-gene relationships are underexplored. scHOT (Ghazanfar et al., 2020) makes the first attempt towards this direction: they infer gene pairs with relational differences along a trajectory or across spatial locations. Despite the novel perspective, their approach is rather heuristic: assuming the trajectories and corresponding pseudotime (or the spatial location) are given, they compute gene coexpression at each time point (or location) using weighted univariate correlation (weights are determined by a triangular kernel). To test whether a gene pair is differentially associated along a curve or across spatial location, they use the standard deviation of the series of time-specific gene coexpressions along the curve as the summary statistics and perform a permutation test. Wang et al. (2021b) explore a similar task, but they split the cells into multiple bins along the trajectory first and then compute one covariance matrix (avgCSN) for each bin using only cells from that bin. Finally, they test the differences between the covariance matrices as a whole and report the leverage genes as the differentially associated genes along the trajectory.

3.4.1 Minibatched LDG

Formally put, assume there are p genes and n cells, and each cell is associated with a structure covariate S taking values on a set $\mathcal{S}$. Assume the following data-generating mechanism:

(1) For each $i = 1, \ldots, n$, independently generate S_i from a distribution $\mathcal{Q}$. These are the structure covariates for each cell.

(2) For each $i = 1, \ldots, n$, generate $Z_i \in \mathbb{R}^p$ independently from $\mathcal{P}_{S_i}$, where $\{\mathcal{P}_s : s \in \mathcal{S}\}$ is a class of probability distribution on $\mathbb{R}^p$ indexed by s.

Then both scHOT and avgCSN estimate the dependence of gene pairs under $\mathcal{P}_{S_i}$, which is a $p \times p$ dependence matrix for the joint distribution $\mathcal{P}_{S_i}$. The local aggregation in scHOT or binning in avgCSN further reduces the estimation error from similar time points. The underlying assumption is that $\mathcal{P}_s$, and hence the corresponding dependence matrix indexed by s, varies smoothly as s changes.

The approach we are going to propose instead works on the mixture distribution $\mathcal{P}_S$ where S is treated as randomly generated from $\mathcal{Q}$. For gene pairs (i, j), we use

$Z_1, \ldots, Z_n$ to estimate at cell k the LDG matrix:

$$G_k(i,j) := \mathbf{I}\{T_k(i,j) > t\}, \quad T_k(i,j) := \frac{f_{ij}(Z_{ki}, Z_{kj}) - f_i(Z_{ki}) f_j(Z_{kj})}{f_i(Z_{ki}) f_j(Z_{kj})}. \quad (3.25)$$

Then for each cell k, we aggregate the LDG matrix of its neighboring cells according to their structure covariate value closeness (e.g. pseudotime or spatial location). This local aggregation pools G to get the final estimate of time/location-specific gene coexpression, and was designed to reduce estimation error. We call these estimations the *minibatched LDG*. This local aggregating approach was designed to reduce estimation error. The underlying assumption is that the G matrix, which is a random matrix, moves smoothly in its sample space as the structure covariate S changes.

In Section 3.5.2, we provide two real data examples to demonstrate how minibatched LDG can be used to highlight local structural change.

3.5 Empirical evaluation

3.5.1 Simulation results

In this section, we consider simulations that resembling single-cell data to gain insights underlying the behavior of aLDG relative to the other methods. Specifically, we investigate scenarios where the bivariate relationship is (1) finite mixture; (2) linear or nonlinear; (3) monotone or non-monotone. See Figure 3.3 for all the synthetic data distributions we considered. We evaluate each dependence measure from the following perspective: (1) ability to capture complex relationship; (2) ability to accumulate subtle local dependence; (3) interpretation of strength of dependence in common sense; (4) power as an independence test; and (5) computation time. In the following, we focus on one perspective in each subsection, showing selective examples that inform our conclusions, relegating other examples to supplementary materials.

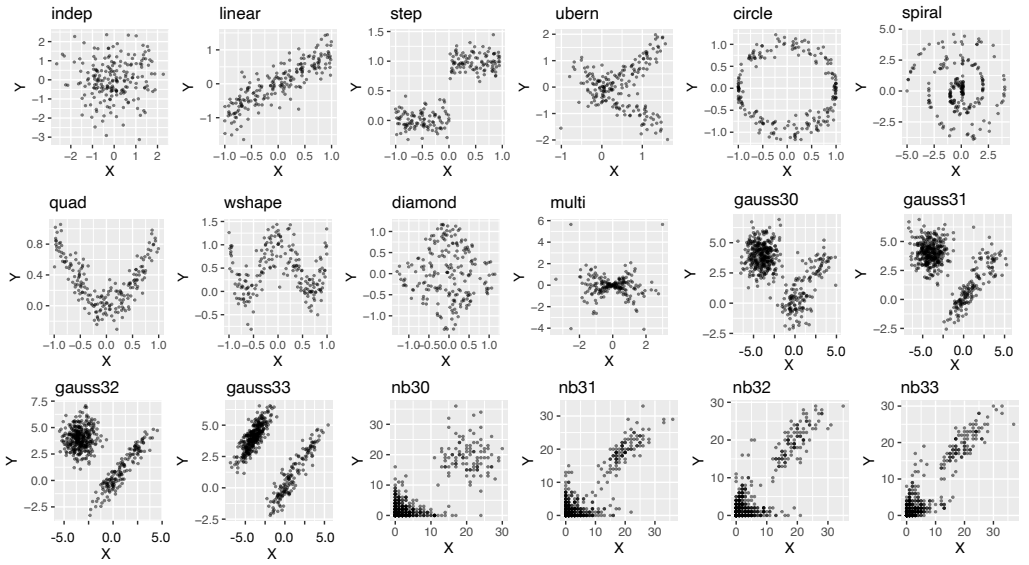


Figure 3.3: A summary of all the synthetic bivariate data distribution we considered in this paper. For each data distribution we plot the corresponding scatter plot using 1000 samples. We believe this series of distributions are representative enough as it covers cases from linear to nonlinear, monotone to nonmonotone, and also probabilistic mixture.

Detecting nonlinear, non-monotone relationships. By construction, aLDG is expected to detect any non-negligible deviation from independence. Though many existing measures, such as HSIC, Hoeffding's D, dCor, $\tau^\star$, claim to be sensitive to nonlinear, non-monotone relationships, some approaches are known to perform poorly under certain circumstances. By contrast, aLDG outperforms most of its competitors in the following standard evaluation experiment. Figure 3.4 illustrates three points: (1) at independence, except for dCor, HHG, and MIC, most measures produce negligible values, as desired; (2) for linear and monotone relationship, all measures produce high values as expected; and (3) for nonlinear non-monotone relationships only aLDG, dCor, HHG and MIC produce high values consistently. In conclusion, only aLDG can effectively detect various types of dependency relationships while maintaining near-zero value at independence. dCor, HHG, and MIC are known to be sensitive to small, artificial deviations from independence, and these simulations reveal that they are indeed too sensitive as they often produce high values at independence. A big portion of scRNA-seq data are collected over time; therefore, nonlinear, non-monotone and specifically oscillatory relationships are expected to happen. Therefore it is desirable to have a measure that is sensitive to dependence while remaining near zero of true independence, even under small perturbations.

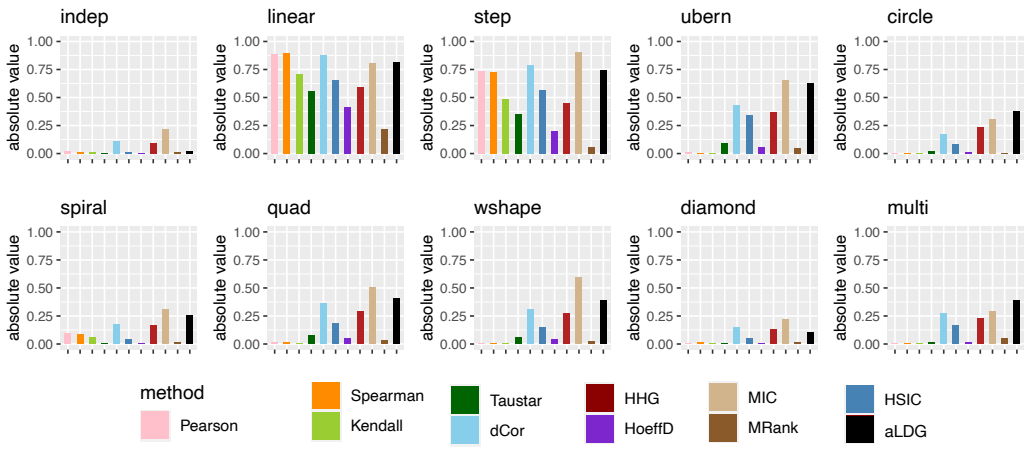


Figure 3.4: Empirical dependency estimates obtained for different data distributions for a variety of relationships between a pair of variables. For the visualization of different data distributions, see Figure 3.3. Here we show the corresponding dependence level given by different measures using 200 samples (averaged over 50 trials).

Accumulating subtle local dependencies. aLDG detects the subset of the sample space that shows a pattern of dependence. In Figure 3.5, we simulated data as a bivariate Gaussian mixture consisting of three components with a varying proportion of highly dependent components and estimated the corresponding dependence level. We find that aLDG, together with other dependence measures designed to capture local dependence (HHG and MIC) increase with the proportion of highly correlated components, indicates that these global dependence measures can also detect subtle local dependence structure. Similar results are obtained for Negative Binomial mixtures (Figure 3.15 in Appendix 3.7.10). As the finite mixture relationship is a common choice of model for scRNA-seq data, this suggests that measures able to accumulate dependencies across individual components could considerably benefit scRNA-seq data analysis.

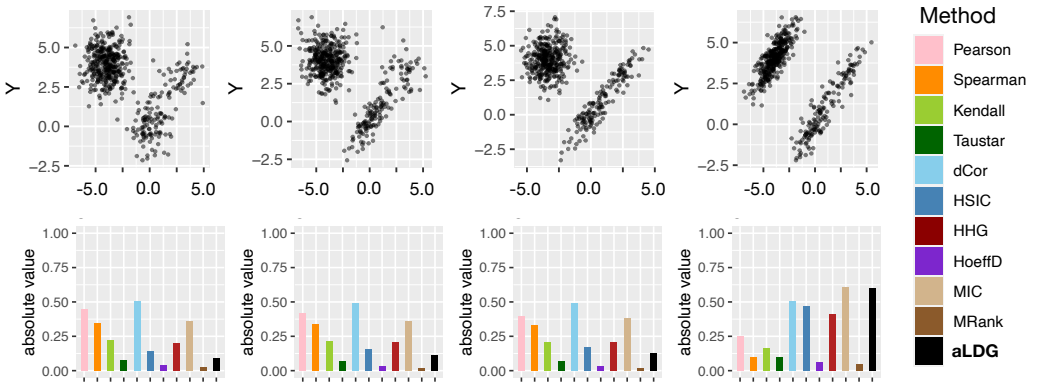


Figure 3.5: Empirical aLDG value for Gaussian mixtures. In each plot we show the dependence level given by different measures for 200 samples (averaged over 50 trials). The data are generated as a three-component Gaussian mixture. From left to right, there are 0, 1, 2 and 3 out of 3 components with correlation of 0.8, while the remaining components have correlation 0, i.e., the dependence level increases from left to right. For the visualization of these different data distributions, see Figure 3.3.

aLDG interprets degree of dependencies. Degree of dependencies While it is hard to define the relative dependence level in general, we argue that when one random variable is a function of the other, $Y = h(X)$, then the pair should be regarded as having the perfect dependence (and be assigned of dependence level 1). Moreover, the dependence level should decrease as independent noise is added. That is, for $Y_\epsilon = h(X) + \epsilon$, where $\epsilon \perp X$, one should expect the dependence measure δ to satisfy $\delta(Y_\epsilon, X) < \delta(Y, X)$. We checked this monotonicity property by simulating data with several bivariate relationships and varying levels of noise (Figure 3.6). Specifically, we simulate the noise ϵ to be standard normal, and $Y = h(X) + c\epsilon$ where $c \in [0, 1]$ indicates the noise level. We find that aLDG, HSIC, MIC, dCor, and HHG all show a clear decreasing pattern as the noise level increases; however, aLDG shows the most consistent monotonic drop from perfect dependence as the noise level increased.

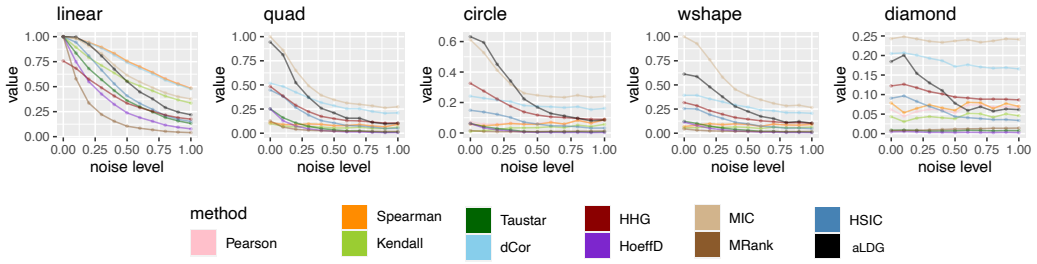


Figure 3.6: Empirical dependence measure versus noise levels for different bivariate relationships. For the visualization of different data distributions, see Figure 3.3. The results are shown for 100 samples (averaged over 50 trials). We claim that the higher the noise level is, the lower the estimated degree of dependence should be. Compared with other measures, aLDG decreases significantly as the noise level increases, and hence correctly infers the relative degree of dependence.

Power as an independence test. Dependence measures are natural candidates for tests of independence. In this context, most existing dependence measures rely on bootstrapping or permutation to determine significance; hence we adopt this practice for all the dependence measures under comparison. Figure 3.7 shows the empirical power under test level 0.05 for various types of data distribution and sample size, where we do 200 repetitions of permutations to estimate the null distribution. We observe the following outcomes: (1) almost all tests have controlled type-I error under independence; (2) Pearson's ρ, Spearman's ρ_S and Kendall's τ are powerless for testing nonlinear and non-monotone relationships; (3) aLDG, HHG, and HSIC are consistently among the top three most powerful approaches for testing both linear and nonlinear, monotone and non-monotone relationships. Similar observations can be made for tests based on Gaussian mixtures (Figure 3.16 in Appendix 3.7.10) and Negative Binomial mixtures (Figure 3.17 in Appendix 3.7.10).

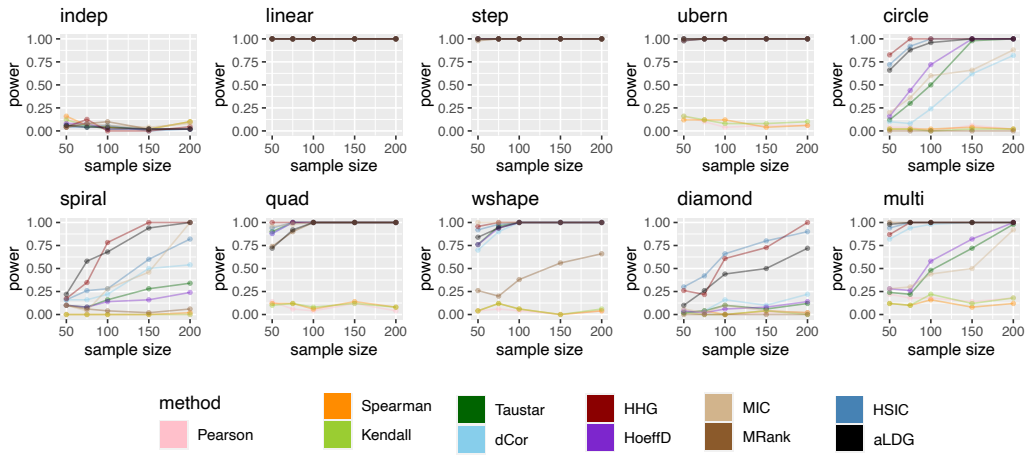


Figure 3.7: The empirical power of permutation test at level 0.05, based on different dependency measures under different data distributions and sample sizes. For the visualization of different data distributions, see Figure 3.3. The power is estimated using 50 independent trials.

Computational comparisons. Theoretically speaking, aLDG requires $O(n^2)$ in time of computation (where n is the number of samples), which is comparable to reported requirements for most dependence measures that can detect complex relationships. This empirically confirmed in a comparison of the computation time of aLDG with all its competitors. In (Figure 3.18 in Appendix 3.7.10) we plot the time of computation versus sample size n for different dependence measures[3]. In previous evaluations, we saw that HHG as a method motivated from capturing local dependence structure, was indeed a strong competitor to aLDG: it has high power as an independence test across almost all the data distribution we considered; however, it requires $O(n^3)$ time of computation, and (Figure 3.18 in Appendix 3.7.10) shows this large discrepancy from all the other methods, which normally takes $O(n^2)$ time.

Highlighting bifurcating point along trajectory. Realistic gene expression dynamics that include gene cooperation are hard to be captured using a probabilistic model. Instead, success has been achieved by modeling the gene cooperation dynamics using ordinary differential equations (ODE). The state-of-art method is a simulation software called BoolODE Pratapa et al. (2020). For each gene, BoolODE requires a Boolean function that specifies how that gene's cooperators combine to control its state. Each Boolean function is then converted into a nonlinear ordinary differential equation, together with Gaussian noise terms to make the equation stochastic. Simulating this system of stochastic differential equations generates the requisite

[3]The time include some constant wrapper function loading time, therefore, might be longer than a direct function call; however, the relative scale is still correct.

gene expression data. Under this model, we would like to estimate time-specific gene coexpression and detect potential branching points in the developmental trajectory.

The data is simulated using the simulation tool based on gene regulation specifically: BoolODE (Pratapa et al., 2020). The data are generated via simulating a series of ordinary differential equations given by the kinetics function defined by the gene regulation relationship.

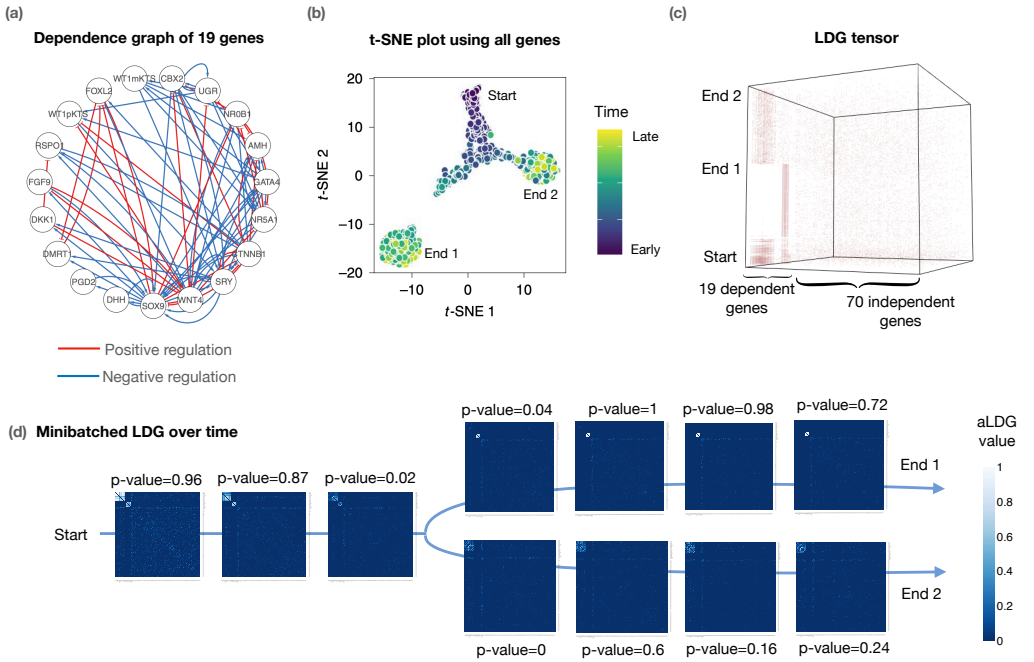


Figure 3.8: Trajectory analysis on simulated data. **(a)** The boolean regulating relationship among the 19 dependent genes (left top panel). **(b)** The tSNE plots of the simulated data. **(c)** The aLDG tensor computed using all the data at once. **(d)** The minibatched aLDG with the sLED based permutation tests p-values annotated on the top. We show just a few selected pseudotime points along the continuous trajectory.

We simulate 500 cells and 89 genes in total, in which 19 genes form a Boolean regulating relationship (shown in Figure 3.8 (a)), and the remaining 70 genes are independent. We design the simulation to generate a bifurcating trajectory, which starts from a common origin, and develops into two stable states (shown in Figure 3.8 (b)). We compute the LGD tensor and order the cells based on their pseudotime (we use slingshot (Street et al., 2018) to estimate the pseudotime). Figure 3.8 (c) visualizes the LDG tensor with different time stages and gene sets annotated. We can see that interesting patterns emerge: in the shared branch, the dependent genes have a two-block dependence structure, while after the splitting point, each branch preserves only one of the blocks; while for independent genes, no structure emerged,

as we expected. These patterns become more evident in the minibatched LDG over time shown in Figure 3.8 (d), where we take a window size $w = 20$ and do aggregating as we described earlier. At each time point, we conduct sLED test for its gene coexpression matrix and that of the time w units after it., and output the corresponding p-value. We observe that, the resulting p-values are only smaller than our testing level 0.05 near the branching point, meaning that our minibatched LDG method can reveal statistically significant changes around the branching point.

3.5.2 Real data applications and realistic simulations

In this section, we evaluate the performance of aLDG among the other measures using scRNA-seq data from two studies.

Case study: Chu dataset:. This dataset (Chu et al., 2016) contains 1018 cells of human embryonic stem cell-derived lineage-specific progenitors. The seven cell types, including H1 embryonic stem cells (H1), H9 embryonic stem cells (H9), human foreskin fibroblasts (HFF), neuronal progenitor cells (NPC), definitive endoderm cells (DEC), endothelial cells (EC), and trophoblast-like cells (TB), were identified by fluorescence-activated cell sorting (FACS) with their respective markers. On average, 9600 genes are measured per cell. In the following, we show some special gene pairs that exhibit strong, weak, or no relational patterns and the corresponding dependence values produced by different measures. We find that only aLDG gives a high value for strong relational patterns no matter how complex the pattern composition is; maintains near-zero values for known independent cases; and avoids a spurious relationship skewed by technical noise and sparsity (Figure 3.9).

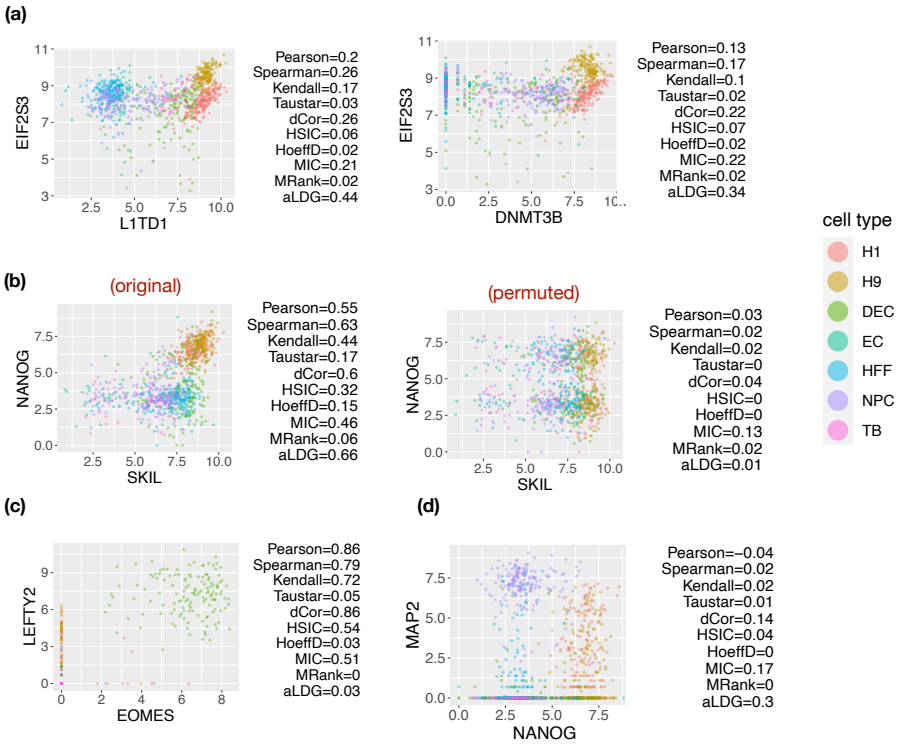


Figure 3.9: Example of gene pair scatter plots from the Chu dataset, which has 1018 cells from 7 cell types. Gene expression is recorded as counts per million (CPM) and $\log_2$ transformed. In each plot, we show the scatter plot of $\log_2(\text{CPM}+1)$ for a pair of genes and provide the corresponding estimated dependence values using different methods to the right of the plots. **(a)** aLDG gives a much higher value than the others in these scenarios which appear to illustrate a strong mixture dependence pattern, even when the signal is predominantly in one cell type. **(b)** aLDG produces a high value for the obvious three mixture relationship in the first subplot. By contrast, in the second subplot, the cell identity are randomly shuffled for each gene pair, resulting in a constructed case of independence. Most measures, including aLDG, give near-zero values in this setting. The exception is MIC, which gives a misleadingly high value. **(c)** This example illustrate performance when there is a high level of sparsity: MIC and the moment-based methods like Pearson, dCor, and HSIC provide estimates that are greatly overestimated, while aLDG, TauStar, and Hoeffding's D are not influenced by this phenomenon. **(d)** This gene pair combines the challenge of sparsity with considerable noise: aLDG is still able to capture the less noisy, local cluster pattern in the upper left corner.

Detecting change point along trajectory: Mouse liver datasets. The data set we use is a merged data set from four different sources using scMerge (Lin et al., 2019), as scHOT (Ghazanfar et al., 2020) did. The dataset contains cells captured from 8

real-time stages, different time stages may contain different cell types. The scHOT (Ghazanfar et al., 2020) paper conducted downstream analysis on this dataset for the three most interesting cell types: Cholangiocyte, Hepatoblast, and Hepatocyte (540 cells in total). Particularly, Hepatoblast cells are a predecessor of both Cholangiocyte and Hepatoblast cells, that is, at some time point the Hepatoblast lineage splits into two different developmental branches: one becomes Cholangiocyte cells, and the other becomes Hepatoblast cells. We focus on these three cell types in this section.

In Figure 3.10, we plot the first two principal components and indicate cell types and real-time stages for each point (cell), with the curves estimated by slingshot (Street et al., 2018) using only a randomly selected half of the data. We can see that the curves fit the data well, and the real-time stages generally agree with the pseudotime. Then we use the remaining half of the data to estimate the minibatched LDG. In Figure 3.11, we show results for the curve starting from Hepatoblast and ending at Cholangiocyte (conclusions are similar for the other branch). We visually spot a consistent emergence of strong gene coexpression patterns around the branching time (framed by the red rectangle).

Now that we have a gene coexpression matrix (i.e. the minibatched LDG) that changes over pseudotime, we consider the task of change point detection. Our estimated time-specific gene coexpression appears to be very sparse in many stages, making the dynamic community estimation based on the stochastic block model inappropriate. Other methods that impose fewer structure constraints require lots of tuning and computation time (Wang et al., 2021a), in order to get high-confidence results. In the following, we present a simple heuristic method instead, which works well in simulation and real data examples. Specifically, we use sLED (Zhu et al., 2017) to test whether time (i.e. pseudotime rank) i_1 and time i_2 are different: we input LDG tensor, and during permutation, we permute the entire time indices; the differences matrix is computed as the absolute differences between minibatched LDGs at time i_1 and i_2 using window size w (i.e. averaged LDG within the $[i_1 - w, i_1 + w]$th and the $[i_2 - w, i_2 + w]$th samples).

At each time point, we conduct sLED test for its gene coexpression matrix and that of the time w units after it., and output the corresponding p-value. We observe that the resulting p-values are only smaller than our testing level 0.05 near the branching point, meaning that our minibatched LDG method can reveal statistically significant changes around the branching point.

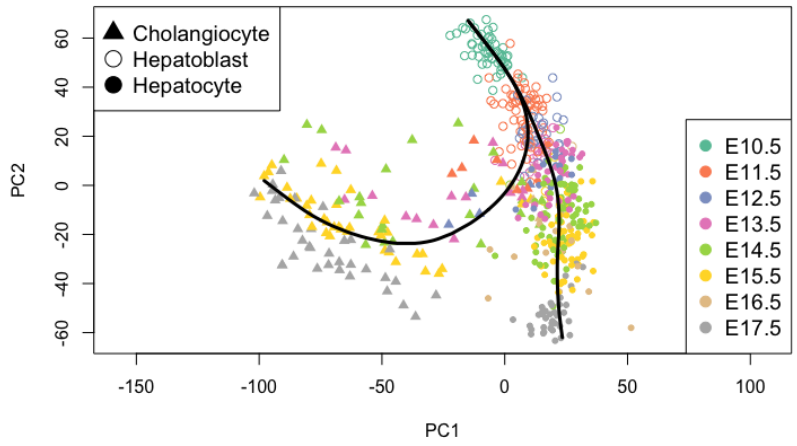


Figure 3.10: The PCA plot of liver development data. These data contain three cell types (annotated by different shapes) and and 8 stages (annotated by different colors). The black curve is the developmental trajectory fitted using slingshot with only half of the data.

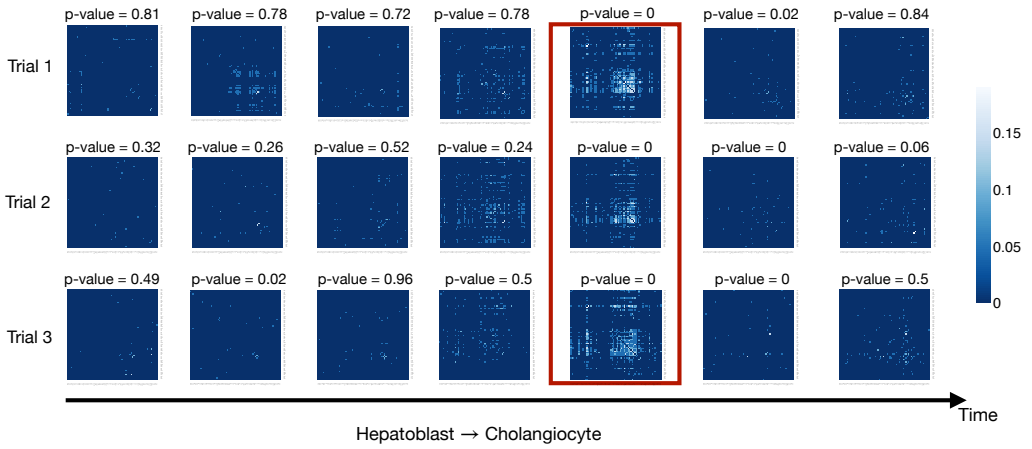
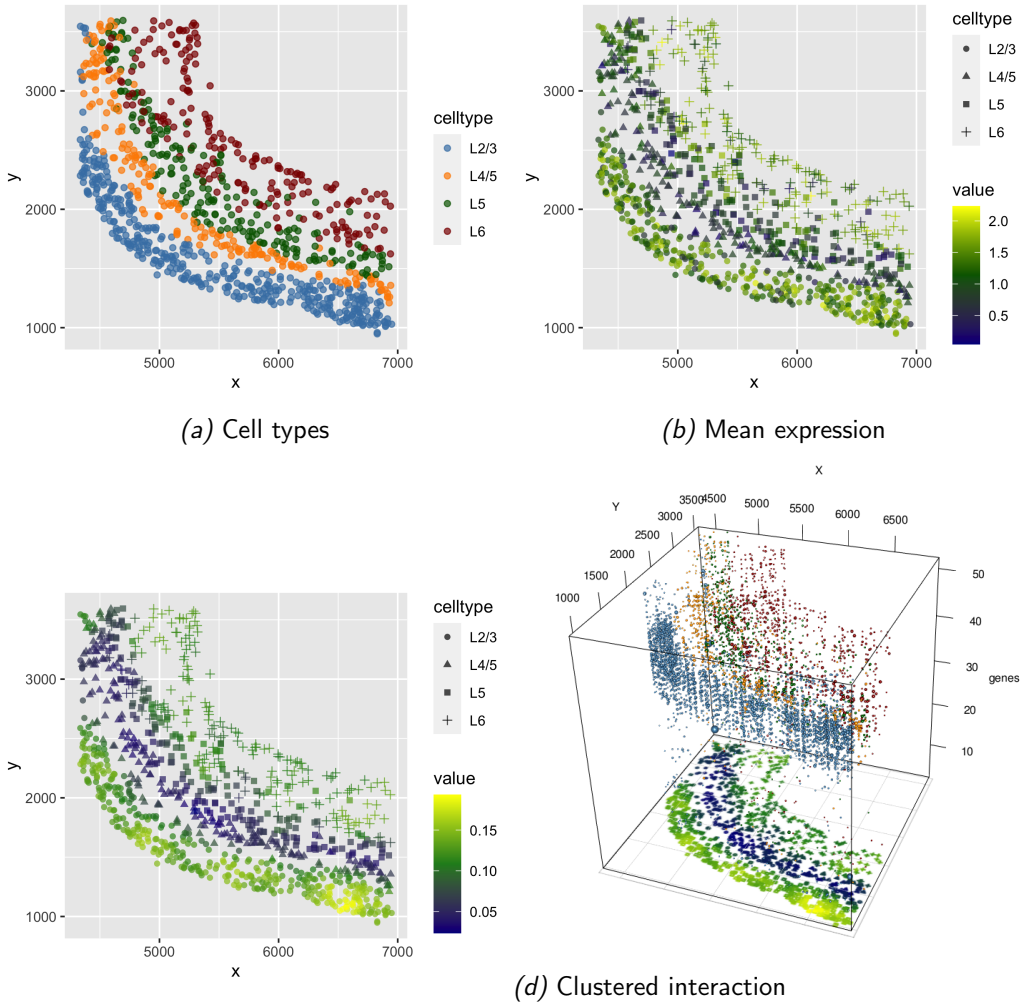


Figure 3.11: The minibatched LDG at some interesting pseudotime points for the curve starting with Hepatoblast cells and ending with Cholangiocyte cells, estimated using the other half of the data (the first half was used to estimate pseudotime). We show the estimated minibatched LDG for three independent trials (i.e., different data-splitting). We annotate on top of each coexpression matrices the sLED p-value (testing whether the current time point is significantly different from the latter one right after it).

Highlight brain structure: MERFISH brain datasets. This dataset was used by Fischer et al. (2021) for cell communication estimation. The dataset was first assimilated by Zhang et al. (2021), who measured mouse primary motor cortex with multiplexed error-robust fluorescence in situ hybridization (MERFISH) in 634 images across two mice with 254 genes observed in 284,098 cells. The cell-types were originally annotated by Zhang et al. (2021). We focus on L2/3, L4/5, L5, L6 cells, which were shown to form an interesting transition in the original paper. We further constrain the other experimental conditions to rule out any other confounding effects. The final dataset has around 2600 cells, and we can see that in Figure 3.12, the gene interaction shows more spatial patterns: the top and the bottom (especially the top) layer seem to have more interaction than the middle layer, and especially in the top layer.



(a) Cell types

(b) Mean expression

(c) Mean interaction

(d) Clustered interaction (sum of edges each gene)

Figure 3.12: The spatial plot annotated by cell-type, total gene expression level and total gene interaction, using 52 genes (the union of 26 differentially co-expressed and 26 non-differentially expressed genes), and L2/3, L4/5, L5, L6 IT cells. (a) The cell type annotation for each spatial sample; **(b)** The average of all gene expression levels for each spatial sample; **(c)** the average of all edges in minibatched LDG for each spatial sample; **(d)** The degree of each gene in minibatched LDG for each spatial sample, larger 3d point size represents larger degrees. We can see that gene 30-50 contribute to most of the gene interactions.

3.6 Conclusion and Discussion

In this paper, we formalize the idea of averaging the *cell-specific gene association* (Dai et al., 2019; Wang et al., 2021b) under a general statistical framework. We show

that this approach produces a novel univariate dependence measure, called aLDG, that can detect nonlinear, non-monotone relationships between a pair of variables. We then develop the corresponding theoretical properties of this estimator, including robustness and consistency. We also provide several hyper-parameter choices that are more justifiable and effective. Extensive simulations, motivated by expected scRNA-seq gene co-expression relationships and real data applications, show that this measure outperforms existing independence measures in various aspects: (1) it accumulates subtle local dependence over sub-populations; (2) it successfully interprets the relative strength of a monotonic function of dependence in the presence of noise better than many other measures that arose from independence test; (3) it is sensitive to complex relationships while robustly maintaining near-zero value at true independence, while several other measures are often overly sensitive to slight perturbations from independence and noise; (4) it computes comparatively rapidly compared to other dependence measures designed to capture complex relationships. Other measures perform well in some settings but fail in others that are highly relevant to the single-cell setting. For instance, MIC performed well as part of the sLED test for differences in co-expression matrices, but this measure tends to produce a high estimate of dependence even when the variables are independent, or nearly so (Figure 3.4 and Figure 3.6). The moment-based methods like Pearson, dCor, and HSIC perform poorly when the expression values are sparse, producing false indications of correlation (Figure 3.9), and yet sparsity is the norm in most single cell data. Our method is implemented in the R package aLDG[4], where we also include all the other methods that we have compared with, as well as functions for replication of experiments.

The aLDG method does have some practical challenges: as a measure based on density estimation, the hyperparameter choices such as bandwidth can affect the performance of the measure. Though we provide some asymptotically optimal choices of those hyperparameters, in practice, they can fail due to the small sample size. For any given setting, the hyperparameters can be adjusted based on realistic simulations of the actual data and a solid understanding of the scRNA-seq data distribution. Similarly, due to the reliance on density estimation, it is hard to extend this measure to a multivariate setting. The sample size required for accurate estimation grows exponentially with the dimension. In practice, this limitation has little practical importance because gene co-expression studies focus on bivariate relationships.

[4] `https://github.com/JINJINT/aLDG`

3.7 Appendices

3.7.1 From avgCSN to aLDG

Recall that we consider only a pair of random variables X, Y whose joint and marginal densities exist and have the same support, and denote f_{XY}, f_X, f_Y as their joint and marginal densities. Also, let $\widehat{f}_{XY}, \widehat{f}_X, \widehat{f}_Y$ be the estimated densities given observations of (X, Y), and $\widehat{p}_{X,Y}(x, y)$ be the proportion of samples points in a square of side length h centering at (x, y), and $\widehat{p}_X$ and $\widehat{p}_Y$ be defined similarly for the marginal distribution.

First we point out that a reformulation of avgCSN statistics reveals its link to the population dependence measure we are going to introduce. Under our notation, the original avgCSN Wang et al. (2021b) can be written as

$$\text{avgCSN} := \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{ \frac{\widehat{p}_{X,Y}(x_i, y_i) - \widehat{p}_X(x_i)\widehat{p}_Y(y_i)}{\sqrt{\widehat{p}_X(x_i)(1-\widehat{p}_X(x_i))\widehat{p}_Y(y_i)(1-\widehat{p}_Y(y_i))}} \geq \frac{\Phi^{-1}(1-\alpha)}{\sqrt{n}} \right\},$$

where Φ^{-1} is the quantile function of standard normal. When using a particular choice $\widehat{f}_{XY} = \widehat{p}_{X,Y}/h^2, \widehat{f}_X = \widehat{p}_X/h, \widehat{f}_Y = \widehat{p}_Y/h$, we have

$$\text{avgCSN} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{ \frac{\widehat{f}_{X,Y}(x_i, y_i)h^2 - \widehat{f}_X(x_i)h\widehat{f}_Y(y_i)h}{\sqrt{\widehat{f}_X(x_i)h(1-\widehat{f}_X(x_i)h)\widehat{f}_Y(y_i)h(1-\widehat{f}_Y(y_i)h)}} \geq \frac{\Phi^{-1}(1-\alpha)}{\sqrt{n}} \right\}.$$

Assuming the bandwidth $h \to 0$ and $h\sqrt{n} \to \infty$, the expression can be approximated by the following

$$\text{avgCSN} \approx \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{ \frac{\widehat{f}_{X,Y}(x_i, y_i) - \widehat{f}_X(x_i)\widehat{f}_Y(y_i)}{\sqrt{\widehat{f}_X(x_i)\widehat{f}_Y(y_i)}} \geq t_n \right\}, \quad \text{where } t_n = \frac{\Phi^{-1}(1-\alpha)}{h\sqrt{n}}.$$

3.7.2 Proof for Theorem 3.4

Proof. Denote the joint and marginal density of F as $f_{X,Y}, f_X, f_Y$. Consider a fixed contamination position (x', y'), then we have the corresponding contaminated joint and marginal density as

$$f_X^{(x')}(x) := \begin{cases} (1-\epsilon)f_X(x), & if\ x \neq x', \\ \infty, & if\ x = x'; \end{cases},$$

$$f_Y^{(y')}(y) := \begin{cases} (1-\epsilon)f_Y(y), & if\ y \neq y', \\ \infty, & if\ y = y'; \end{cases}$$

$$f_{X,Y}^{(x',y')}(x, y) := \begin{cases} (1-\epsilon)f_{X,Y}(x, y), & if\ (x, y) \neq (x', y'), \\ \infty, & if\ (x, y) = (x', y'). \end{cases}.$$

Denote the density gap under original distribution F as $\Delta^{\text{gap}} := f_{X,Y} - f_X f_Y$, and the corresponding density gap under contaminated distribution as $\Delta_{\text{gap}}^{(x',y')} :=$

$f_{X,Y}^{(x',y')} - f_X^{(x')} f_Y^{(y')}$, then

$$\Delta_{\text{gap}}^{(x',y')}(x,y) = (1-\epsilon)\Big(\Delta_{\text{gap}}(x,y) + \epsilon f_X(x) f_Y(y)\Big) \quad \text{if } x \neq x' \text{ and } y \neq y',$$

and the contaminated aLDG_t statistics

$$\begin{aligned}
&\text{aLDG}_t^{(x',y')} = \text{Pr}_{F'}\left\{\Delta_{\text{gap}}^{(x',y')} > t\sqrt{f_X^{(x')}(x) f_Y^{(y')}(y)}\right\} \\
&\leq \text{Pr}_{F'}\left\{(1-\epsilon)\Big(\Delta_{\text{gap}}(x,y) + \epsilon f_X(x) f_Y(y)\Big) > t(1-\epsilon)\sqrt{f_X(x) f_Y(y)},\ (x,y) \neq (x',y')\right\} \\
&\quad + \text{Pr}_{F'}\left\{(x,y) \neq (x',y')\right\} \\
&= (1-\epsilon)\text{Pr}_F\left\{\frac{\Delta_{\text{gap}}(x,y)}{\sqrt{f_X(x) f_Y(y)}} + \epsilon\sqrt{f_X(x) f_Y(y)} > t\right\} + \epsilon \\
&\overset{(a)}{\leq} (1-\epsilon)\text{Pr}_F\left\{\frac{\Delta_{\text{gap}}(x,y)}{\sqrt{f_X(x) f_Y(y)}} + \epsilon f_{\max} > t\right\} + \epsilon = (1-\epsilon)\text{aLDG}_{t-\epsilon f_{\max}} + \epsilon \\
&\leq (1-\epsilon)\big(\text{aLDG}_t + |\text{aLDG}_{t-\epsilon f_{\max}} - \text{aLDG}_t|\big) + \epsilon \\
&\overset{(b)}{\leq} (1-\epsilon)\big(\text{aLDG}_t + L f_{\max}\epsilon\big) + \epsilon,
\end{aligned}$$

where (a) comes from the assumption that $f_{\max} := ||\sqrt{f_X f_Y}||_\infty < \infty$, and (b) comes from the assumption that $|\text{aLDG}_{t-\epsilon} - \text{aLDG}_t| \leq L\epsilon$ for all $\epsilon > 0$.

Therefore,

$$\begin{aligned}
\text{IF}\left((x',y'), R_{\text{aLDG}_t}, F\right) &:= \lim_{\epsilon\to 0} \frac{\text{aLDG}_t^{(x',y')} - \text{aLDG}_t}{\epsilon} \\
&\leq -\text{aLDG}_t + (1-\epsilon) L f_{max} + 1 \\
&\leq L f_{max} + 1
\end{aligned}$$

Since the upper bound of IF does not depend on location of (x', y'), therefore,

$$\text{GES}(R_{\text{aLDG}_t}, F) \leq L f_{\max} + 1 < \infty.$$

□

3.7.3 Proof for Proposition 1

Proof. Denote the joint and marginal density of F as $f_{X,Y}, f_X, f_Y$. Consider a fixed contamination point (x', y') with mass ϵ, then we have the corresponding contaminated joint and marginal density as

$$f_X^{(x')}(x) := \begin{cases} (1-\epsilon) f_X(x), & if\ x \neq x', \\ \infty, & if\ x = x'; \end{cases},$$

$$f_Y^{(y')}(y) := \begin{cases} (1-\epsilon) f_Y(y), & if\ y \neq y', \\ \infty, & if\ y = y'; \end{cases}$$

$$f_{X,Y}^{(x',y')}(x,y) := \begin{cases} (1-\epsilon) f_{X,Y}(x,y), & if\ (x,y) \neq (x',y'), \\ \infty, & if\ (x,y) = (x',y'). \end{cases}.$$

Recall that the density gap $\Delta^{\text{gap}} := f_{X,Y} - f_X f_Y$, and hence the contaminated gap,

$$\Delta_{\text{gap}}^{(x',y')}(x,y) = (1-\epsilon)\Big(\Delta_{\text{gap}}(x,y) + \epsilon f_X(x) f_Y(y)\Big), \quad \text{if } (x,y) \neq (x',y')$$

and the contaminated aLDG statistics

$$\begin{aligned}
&\text{aLDG}_0^{(x',y')} = \Pr_{F'}\{\Delta_{\text{gap}}^{(x',y')} > 0\} \\
&\leq \Pr_{F'}\left\{(1-\epsilon)\Big(\Delta_{\text{gap}}(x,y) + \epsilon f_X(x) f_Y(y)\Big) > 0, (x,y) \neq (x',y')\right\} \\
&\qquad + \Pr_{F'}\left\{(x,y) \neq (x',y')\right\} \\
&= (1-\epsilon)\Pr_F\left\{\Delta_{\text{gap}}(x,y) + \epsilon f_X(x) f_Y(y) > 0\right\} + \epsilon.
\end{aligned}$$

Note that

$$\begin{aligned}
&\Pr_F\left\{\Delta_{\text{gap}}(x,y) + \epsilon f_X(x) f_Y(y) > 0\right\} \\
=&\Pr_F\left\{\Delta_{\text{gap}}(x,y) > 0\right\} + \Pr\left\{-\epsilon f_X(x) f_Y(y) < \Delta^{\text{gap}}(x,y) \leq 0\right\} \\
=&\text{aLDG}_0 + \Pr_F\left\{1-\epsilon < \frac{f_{X,Y}(x,y)}{f_X(x) f_Y(y)} \leq 1\right\} \\
=&\text{aLDG}_0 + \Pr_F\{1-\epsilon < c_F(u,v) \leq 1\},
\end{aligned}$$

where $c_F(u,v)$ is the joint density of $u := F_X^{-1}(x), v := F_Y^{-1}(x)$, i.e. the corresponding copula representation of distribution F. Then, denoting the volume of set $\Gamma_t : \{(u,v,t) : c_F(u,v) \leq t\}$ as $\text{Vol}(t)$, and the area of sublevel set $\gamma_t : \{(u,v) : c_F(u,v) \leq t\}$ as $\text{A}(t)$, and the contour line $\mathcal{C}(t) := \{(u,v) : c_F(u,v) = t\}$, we have

$$\begin{aligned}
&\lim_{\epsilon \to 0} \frac{1}{\epsilon} \Pr\{1-\epsilon < c_F(u,v) \leq 1\} = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{1-\epsilon < c_F(u,v) \leq 1} c_F(u,v) du dv \\
=&\lim_{\epsilon \to 0} \frac{\text{Vol}(1) - \text{Vol}(1-\epsilon)}{\epsilon} = \frac{d\text{Vol}}{dt} \mid_{t=1} \\
\overset{(a)}{=}& \frac{A(1)}{||\nabla c_F(u_0,v_0)||_2} \overset{(b)}{\leq} \frac{1}{||\nabla c_F(u_0,v_0)||_2}
\end{aligned}$$

where (u_0, v_0) is some point on $\mathcal{C}_t$ and $\nabla c_F(u_0,v_0)$ is the gradient of c_F at (u_0,v_0), and (a) comes from Theorem 1 in Trinh (2019) using the a.e. smoothness of the joint and marginal densities f_{XY}, f_X, f_Y; (b) uses the trivial bound $A(1) \leq 1$ since we are working on $[0,1]^2$ space.

Plug the above calculation back to IF function, we get

$$\begin{aligned}
\text{IF}\Big((x',y'), R_{\text{aLDG}_0}, F\Big) &= \frac{(1-\epsilon)\Big(\text{aLDG}_0 + \text{Vol}(1) - \text{Vol}(1-\epsilon)\Big) + \epsilon}{\epsilon} \\
&= 1 - \text{aLDG}_0 - \text{Vol}(1) \\
&\qquad + \lim_{\epsilon \to 0} \text{Vol}(1-\epsilon) + \lim_{\epsilon \to 0} \frac{1}{\epsilon}\left(\text{Vol}(1) - \text{Vol}(1-\epsilon)\right) \\
&\leq 1 - \text{aLDG}_0 + \frac{1}{||\nabla c_F(u_0,v_0)||_2},
\end{aligned}$$

where (u_0, v_0) is some point on the contour line $\mathcal{C}_t := \{(u,v) : c_F(u,v) = t\}$, and $\nabla c_F(u_0, v_0)$ is the gradient of c_F at (u_0, v_0).

Note that this upper bound is irrelevant with (x', y'), therefore we have

$$\mathrm{GES}(R_{\mathrm{aLDG}}, F) \leq 1 - \mathrm{aLDG}_0(F) + \frac{1}{||\nabla c_F(u_0, v_0)||_2} < \infty,$$

as long as X, Y is not independent.

However, when X, Y are independent, we have $c_F(u,v) \equiv 1$ for all $(u,v) \in [0,1]^2$, and $\mathrm{aLDG}_0 = 0$, then we have

$$\begin{aligned}
\mathrm{aLDG}_0^{(x',y')} &\geq \Pr_{F'}\{\Delta_{\mathrm{gap}}(x,y) + \epsilon f_X(x) f_Y(y) > 0, (x,y) \neq (x',y')\} \\
&= (1-\epsilon)\Big(\mathrm{aLDG}_0 + \Pr\{1-\epsilon < c_F(u,v) \leq 1\}\Big) \\
&= (1-\epsilon)(0+1) = 1-\epsilon,
\end{aligned}$$

and hence

$$\mathrm{IF}\Big((x',y'), R_{\mathrm{aLDG}_0}, F\Big) \geq \lim_{\epsilon \to 0} \frac{1-\epsilon}{\epsilon} = \infty.$$

Again this lower bound is irrelevant with (x', y'), therefore we have

$$\mathrm{GES}(R_{\mathrm{aLDG}_0}, F) = \infty.$$

□

3.7.4 Proof for Theorem 3.5

Proof. Denote the set

$$\begin{aligned}
S_t &:= \left\{(x,y) : \frac{f_{XY}(x,y) - f_X(x) f_Y(y)}{\sqrt{f_X(x) f_Y(y)}} > t\right\}, \\
\widehat{S}_t &:= \left\{(x,y) : \frac{\widehat{f}_{XY}(x,y) - \widehat{f}_X(x) \widehat{f}_Y(y)}{\sqrt{\widehat{f}_X(x) \widehat{f}_Y(y)}} > t\right\}.
\end{aligned}$$

From the assumption that $||\widehat{f}_{XY} - f_{XY}||_\infty, ||\widehat{f}_X - f_X||_\infty, ||\widehat{f}_Y - f_Y||_\infty \leq \eta_n$ with probability at least $1 - \frac{1}{n}$, we have the following holds for some constant $c > 0$ with probability at least $1 - \frac{1}{n}$:

$$\begin{aligned}
&\sup_{x,y} \left| \frac{f_{XY} - f_X f_Y}{\sqrt{f_X f_Y}} - \frac{\widehat{f}_{XY} - \widehat{f}_X \widehat{f}_Y}{\sqrt{\widehat{f}_X \widehat{f}_Y}} \right| \\
&\leq \frac{(3c_{\max} + 1)\eta_n}{c_{\min}^{\frac{1}{2}}} + \frac{(3c_{\max} + 1)\eta_n^2 + 2c_{\max}\eta_n}{c_{\min}^{\frac{3}{2}}} < C\eta_n,
\end{aligned}$$

where $C := \frac{(3c_{\max}+1)}{c_{\min}^{\frac{1}{2}}} + \frac{(3c_{\max}+1)+2c_{\max}}{c_{\min}^{\frac{3}{2}}}$ and correspondingly

$$S_{t+C\eta_n} \subseteq \widehat{S}_t \subseteq S_{t-C\eta_n}.$$

As a result, applying the empirical measure $\widehat{P}(S) := \frac{1}{n}\sum_i \mathbf{1}\{(x_i, y_i) \in S\}$ on these three sets, we get

$$\widehat{P}(S_{t+C\eta_n}) \leq \widehat{P}(\widehat{S}_t) = \widehat{\mathrm{aLDG}}(t) \leq \widehat{P}(S_{t-C\eta_n}). \tag{3.26}$$

Using the Hoeffding's inequality on binomials, we get

$$|\widehat{P}(S) - P(S)| < \sqrt{\frac{2\log n}{n}}$$

with probability at least $1 - \frac{1}{2n}$ for any deterministic set S. Applying this inequality to $\widehat{P}(S_{t+C\eta_n})$ and $\widehat{P}(S_{t-c\eta_n})$ in (3.26), we get

$$P(S_{t+C\eta_n}) - \sqrt{\frac{2\log n}{n}} \leq \widehat{P}(\widehat{S}_t) \leq P(S_{t-C\eta_n}) + \sqrt{\frac{2\log n}{n}}$$

with probability at least $1 - \frac{2}{n}$. This further implies that

$$\mathrm{aLDG}_{t+C\eta_n} - \sqrt{\frac{2\log n}{n}} \leq \widehat{\mathrm{aLDG}}(t) \leq \mathrm{aLDG}_{t-C\eta_n} + \sqrt{\frac{2\log n}{n}}$$

with probability at least $1 - \frac{2}{n}$. With the condition that $|\mathrm{aLDG}_{t-\epsilon} - \mathrm{aLDG}_t| \leq L\epsilon$ for all $\epsilon > 0$, we have

$$\mathrm{aLDG}_t - LC\eta_n - \sqrt{\frac{2\log n}{n}} \leq \widehat{\mathrm{aLDG}}_t \leq \mathrm{aLDG}_t + LC\eta_n + \sqrt{\frac{2\log n}{n}},$$

that is

$$\left|\widehat{\mathrm{aLDG}}_t - \mathrm{aLDG}_t\right| \leq LC\eta_n + \sqrt{\frac{2\log n}{n}}$$

with probability at least $1 - \frac{2}{n}$.

□

3.7.5 A uniform variant of consistency

Theorem 3.6. Consider a bivariate distribution F of variable (X, Y) whose joint and marginal densities exist as f_{XY}, f_X, f_Y, and satisfy

$$\inf_{x,y} f_{XY}(x,y),\ \inf_x f_X(x) \inf_y f_Y(y) \geq c_{\min},$$
$$\sup_{x,y} f_{XY}(x,y),\ \sup_x f_X(x) \sup_y f_Y(y) \leq c_{\max},$$

and for some η_n with $\lim_{n\to\infty}\eta_n \to 0$, with probability at least $1-\frac{1}{n}$

$$||\widehat{f}_{XY} - f_{XY}||_\infty, ||\widehat{f}_X - f_X||_\infty, ||\widehat{f}_Y - f_Y||_\infty \le \eta_n;$$

and for some constant $0 < L < \infty$,

$$|\text{aLDG}_{t-\epsilon} - \text{aLDG}_t| \le L\epsilon \quad \text{for all } \epsilon > 0, \quad \text{for all } t \ge 0.$$

Then we have, with probability at least $1-\frac{2}{n}$, we have

$$\sup_{t\ge 0}\left|\widehat{\text{aLDG}}_t - \text{aLDG}_t\right| \le LC\eta_n + 10\sqrt{\frac{\log n}{n}},$$

where C depends only on $c_{\min}, c_{\max}$.

Proof. Recall the bivariate functional

$$T : (x,y) \mapsto \frac{f_{XY}(x,y) - f_X(x)f_Y(y)}{\sqrt{f_X(x)f_Y(y)}}, \quad \widehat{T} : (x,y) \mapsto \frac{\widehat{f}_{XY}(x,y) - \widehat{f}_X(x)\widehat{f}_Y(y)}{\sqrt{\widehat{f}_X(x)\widehat{f}_Y(y)}}.$$

Correspondingly, for a $t \ge 0$, denote the set

$$S_t := \{(x,y) : T(x,y) > t\}, \quad \widehat{S}_t := \left\{(x,y) : \widehat{T}(x,y) > t\right\}.$$

We also denote the collection of such set over all $t \ge 0$ as $\mathcal{S} = \{S_t : t \ge 0\}$.

From proposition 4.20 (Wainwright, 2019), it is easy to see that the class $\mathcal{S}$ has VC dimension at most 1, since it can be written as the subgraph class of the function class $\{g_t : (x,y) \mapsto t - T(x,y); t \ge 0\}$ is a vector space of $dim(1)$ (as function T is deterministic and only t is changing). Using VC theorem, we get

$$\sup_{S\in\mathcal{S}}|\widehat{P}_n(S) - P(S)| \le \sqrt{\frac{32}{n}\left(\log(n+1) + \log(16n)\right)} \le 10\sqrt{\frac{\log n}{n}}$$

with probability at least $1-\frac{1}{2n}$, where $\widehat{P}(S) := \frac{1}{n}\sum_i \mathbf{1}\{(x_i,y_i) \in S\}$ is the empirical measure.

From the assumption that $||\widehat{f}_{XY} - f_{XY}||_\infty, ||\widehat{f}_X - f_X||_\infty, ||\widehat{f}_Y - f_Y||_\infty \le \eta_n$ with probability at least $1-\frac{1}{n}$, we have the following holds for some constant $c > 0$ with probability at least $1-\frac{1}{n}$:

$$\begin{aligned}&\sup_{x,y}\left|\frac{f_{XY} - f_X f_Y}{\sqrt{f_X f_Y}} - \frac{\widehat{f}_{XY} - \widehat{f}_X\widehat{f}_Y}{\sqrt{\widehat{f}_X\widehat{f}_Y}}\right| \\ \le &\frac{(3c_{\max}+1)\eta_n}{c_{\min}^{\frac{1}{2}}} + \frac{(3c_{\max}+1)\eta_n^2 + 2c_{\max}\eta_n}{c_{\min}^{\frac{3}{2}}} < C\eta_n, \end{aligned} \tag{3.27}$$

where $C := \frac{(3c_{\max}+1)}{c_{\min}^{\frac{1}{2}}} + \frac{(3c_{\max}+1)+2c_{\max}}{c_{\min}^{\frac{3}{2}}}$ and correspondingly

$$S_{t+C\eta_n} \subseteq \widehat{S}_t \subseteq S_{t-C\eta_n} \quad \text{for all } t \geq 0.$$

As a result, applying the empirical measure $\widehat{P}(S)$ on these three sets, we get

$$\widehat{P}(S_{t+C\eta_n}) \leq \widehat{P}(\widehat{S}_t) = \widehat{\text{aLDG}}(t) \leq \widehat{P}(S_{t-C\eta_n}) \quad \text{for all } t \geq 0. \tag{3.28}$$

Applying (3.27) to $\widehat{P}(S_{t+C\eta_n})$ and $\widehat{P}(S_{t-c\eta_n})$ in (3.28), we get

$$P(S_{t+C\eta_n}) - 10\sqrt{\frac{\log n}{n}} \leq \widehat{P}(\widehat{S}_t) \leq P(S_{t-C\eta_n}) + 10\sqrt{\frac{\log n}{n}} \quad \text{for all } t \geq 0$$

with probability at least $1 - \frac{2}{n}$. This further implies that

$$\text{aLDG}_{t+C\eta_n} - 10\sqrt{\frac{\log n}{n}} \leq \widehat{\text{aLDG}}(t) \leq \text{aLDG}_{t-C\eta_n} + 10\sqrt{\frac{\log n}{n}} \quad \text{for all } t \geq 0$$

with probability at least $1 - \frac{2}{n}$. With the condition that $|\text{aLDG}_{t-\epsilon} - \text{aLDG}_t| \leq L\epsilon$ for all $\epsilon > 0$ and $t \geq 0$, we have

$$\text{aLDG}_t - LC\eta_n - 10\sqrt{\frac{\log n}{n}} \leq \widehat{\text{aLDG}}_t \leq \text{aLDG}_t + LC\eta_n + 10\sqrt{\frac{\log n}{n}}, \quad \text{for all } t \geq 0$$

that is

$$\sup_{t\geq 0} \left| \widehat{\text{aLDG}}_t - \text{aLDG}_t \right| \leq LC\eta_n + 10\sqrt{\frac{\log n}{n}}$$

with probability at least $1 - \frac{2}{n}$.

□

3.7.6 Uniform estimation error of product kernel density estimator

Definition 3.7. Let β be a positive integer, we define $G(\beta)$ as the class of one-dimensional kernel function K, in which K has support $[-1, 1]$, and $\int K = 1$, $\int |K|^p < \infty$ for any $p \geq 1$, $\int |t|^\beta K(t)dt < \infty$ and $\int t^s K(t)dt = 0$ for any $1 \leq s \leq \beta$.

Definition 3.8. Let β be a positive integer, L be a positive constant, we define $H(\beta, L)$ as the class of one-dimensional density k, such that

$$\left| \frac{d^{\beta-1}k(x)}{x^{\beta-1}} - \frac{d^{\beta-1}k(y)}{x^{\beta-1}} \right| \leq L|x - y|, \quad \text{for all } x, y$$

In the following we analyse a special class of multivariate density function together with a special class of density estimator. Specifically, for positive integer

β, consider density function $k \in H(\beta, L)$, and kernel function $K \in G(\beta)$. For dimension $d \geq 1$, we consider the following multivariate density function in $\mathbb{R}^d$:

$$\mathbf{k}_{\alpha,\mu,r}(\boldsymbol{x}) := \prod_{i=1}^{d} k_{\alpha,\mu,r}(\cdot)(x_i), \quad \text{where } k_{\alpha,\mu,r}(\cdot) = (1-\alpha)k(\cdot) + \alpha \frac{1}{r} k\left(\frac{\cdot - \mu}{r}\right), \tag{3.29}$$

with $\alpha \in [0,1]$ as the mixture proportion, $\mu \geq 0$ the relative location, and $r > 0$ as the relative scale; we also consider the following multivariate kernel function

$$\mathbf{K}_h(\boldsymbol{x}) := \prod_{i=1}^{d} K_h(x_i), \quad \text{where } K_h(\cdot) := \frac{1}{h} K(\frac{\cdot}{h}),$$

with $h > 0 \in \mathbb{R}$; and the corresponding empirical kernel density estimator

$$\widehat{\boldsymbol{K}}_h(\cdot) := \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{K}_h(\boldsymbol{X}_i - \cdot), \tag{3.30}$$

given n observations $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ in $\mathbb{R}^d$.

Proposition 2. Consider $k_{\alpha,\mu,r}$ in (3.29) and $\widehat{K}_h$ in (3.30). Then for any $\delta > 0$, we have

$$\Pr\left\{ \sup_{\boldsymbol{x} \in \mathbb{R}^d} |\widehat{\boldsymbol{K}}_h(\boldsymbol{x}) - \boldsymbol{k}_{\alpha,\mu,r}(\boldsymbol{x})| > \sqrt{\frac{C \log(1/\delta)(1-\alpha+\frac{\alpha}{r})^d}{nh^d}} + c\left(1 - \alpha + \frac{\alpha}{r^{\beta+1}}\right)^d h^{d\beta} \right\} < \delta,$$

where C and c are positive constants which do not depend on h, α, μ, r. Particularly, choosing adaptively

$$h = \left(\frac{C \log \frac{1}{\delta} (1 - \alpha + \frac{\alpha}{r})^d}{c^2 n (1 - \alpha + \frac{\alpha}{r^{\beta+1}})^{2d}} \right)^{\frac{1}{(2\beta+1)d}},$$

we have

$$\Pr\left\{ \sup_{\boldsymbol{x} \in \mathbb{R}^d} |\widehat{\mathbf{K}}_h(\boldsymbol{x}) - \mathbf{k}_{\mu,r}(\boldsymbol{x})| > 2c \left(\frac{C \log \frac{1}{\delta}}{c^2 n} \right)^{\frac{\beta}{2\beta+1}} \left(1 - \alpha + \frac{\alpha}{r^{\beta+1}}\right)^{\frac{\beta+1}{2\beta+1} d} \right\} < \delta.$$

Remark 1. Back to the example in the main paper, the joint density for X, Y we considered is in fact $f_{X,Y}(x,y) = k(x)k(y)$ with $k \in H(1, L)$. And the density estimator we considered is in fact $\widehat{K}_h$ in (3.30) with the one-dimensional kernel function K as boxcar kernel smoothing function (which obviously belongs to $G(1)$). Then use Proposition 4 with $\beta = 1, \alpha = 0, d = 2$, we have with probability at least $1 - 1/n$,

$$||f_{XY} - \widehat{f}_{XY}||_\infty \leq O(n^{-\frac{1}{3}} \sqrt{\log n}).$$

Similarly, for the marginal densities, we have that, with bandwidth $h_n = O(n^{-1/6})$,

$$||f_X - \widehat{f}_X||_\infty \leq O(n^{-\frac{1}{6}}\sqrt{\log n}), \quad ||f_Y - \widehat{f}_Y||_\infty \leq O(n^{-\frac{1}{6}}\sqrt{\log n}).$$

Finally, recall the definition of error rate η_n, we have

$$\eta_n := \sup\{||f_{XY} - \widehat{f}_{XY}||_\infty, ||f_X - \widehat{f}_X||_\infty, ||f_Y - \widehat{f}_Y||_\infty\} \leq O(n^{-\frac{1}{6}}\sqrt{\log n})$$

with probability at least $1 - 1/n$.

Proof. We can decompose the deviation as the following:

$$\left\|\widehat{\boldsymbol{K}}_h - \boldsymbol{k}_{\alpha,\mu,r}\right\|_\infty \leq \left\|\widehat{\boldsymbol{K}}_h - \mathbb{E}\left[\widehat{\boldsymbol{K}}_h\right]\right\|_\infty + \left\|\mathbb{E}\left[\widehat{\boldsymbol{K}}_h\right] - \boldsymbol{k}_{\alpha,\mu,r}\right\|_\infty, \tag{3.31}$$

where the expectation in $\mathbb{E}\left[\widehat{\boldsymbol{K}}_h\right]$ is taken over given samples $X_1, \ldots, X_n$. In the following, we bound each term separately, throughout which we denote expressions that do not depend on h, α, r, μ as constants terms.

Step 1. To bound the first term in (3.31), we use Corollary 2.2 in Giné and Guillou (2002). Firstly we introduce the required condition.

Definition 3.9. (VC class) Let $\mathcal{F}$ be a uniformly bounded collection of measurable functions on $\mathbb{R}^d$. We say that $\mathcal{F}$ is a bounded measurable VC class of functions if the class $\mathcal{F}$ is separable and if there exist positive numbers A and v such that, for every probability measure P on $\mathbb{R}^d$ and every $0 < \epsilon < 1$,

$$\sup_P N(\mathcal{F}, L_2(P), \epsilon \|F\|_{L_2(P)}) \leq \left(\frac{A}{\epsilon}\right)^v, \tag{3.32}$$

where $N(T, d, \epsilon)$ denote the ϵ-covering number of the metric space (T, d), F is the envelope function of $\mathcal{F}$ and the supremum is taken over the set of all probability measure on $\mathbb{R}^d$. The quantities A and v are called the VC characteristics of $\mathcal{F}$.

Lemma 3.10. (Giné and Guillou (2002) Corollary 2.2) Consider $\mathcal{F}$ be a measurable uniformly bounded VC class of functions on $\mathbb{R}^d$ whose VC characters are A, v, and

$$\sup_{f\in\mathcal{F}} \mathrm{Var}_P[f] \leq \sigma^2; \quad \sup_{f\in\mathcal{F}} ||f||_\infty \leq U, \tag{3.33}$$

with $0 < \sigma^2 < \frac{U}{2}$, and $\sqrt{n}\sigma \geq U\sqrt{\log(\frac{U}{\sigma})}$. Then there exist positive constants C and C_0 depending only on A and v such that for all $\lambda \geq C_0$ and t satisfying

$$C_0\sqrt{n}\sigma\sqrt{\log\frac{U}{\sigma}} \leq t \leq \lambda\frac{n\sigma^2}{U},$$

we have

$$\Pr\left\{\sup_{f\in\mathcal{F}}|\sum_{i=1}^{n}f(X_i)-f(X_1)|\geq t\right\}\leq C\exp\left\{-\frac{\log\left(1+\frac{\lambda}{4C}\right)}{\lambda C}\frac{t^2}{n\sigma^2}\right\},$$

where $X_1,\ldots,X_n\overset{iid}{\sim}P$.

Denote the class of functions

$$\mathcal{F}_h:=\left\{\boldsymbol{K}_h\left(\cdot-\boldsymbol{x}\right),\ \boldsymbol{x}\in\mathbb{R}^d\right\}.$$

Then we can write

$$\left\|\widehat{\boldsymbol{K}}_h-\mathbb{E}\left[\widehat{\boldsymbol{K}}_h\right]\right\|_\infty=\sup_{\boldsymbol{x}\in\mathbb{R}^d}\left|\widehat{\boldsymbol{K}}_h(\boldsymbol{x})-\mathbb{E}\left[\widehat{\boldsymbol{K}}_h(\boldsymbol{x})\right]\right|=\frac{1}{n}\sup_{f\in\mathcal{F}_h}\left|\sum_{i=1}^{n}\left(f(\boldsymbol{X}_i)-f(\boldsymbol{X}_1)\right)\right|,$$

where $\boldsymbol{X}_1,\ldots,\boldsymbol{X}_n\overset{iid}{\sim}\boldsymbol{k}_{\alpha,\mu,r}$.

First we examine that $\mathcal{F}_h$ is VC class for $K\in G(\beta)$. Since K is compact supported and polynomial, therefore $\mathcal{F}_h$ is a VC class with $v=\binom{d+\beta}{d}$, and some constant A.

Then we examine the variance and infinity norm condition in (3.33): note

$$\begin{aligned}
\sup_{f\in\mathcal{F}}\mathrm{Var}_P[f]&=\sup_{\boldsymbol{x}\in\mathbb{R}^d}\mathrm{Var}_{\boldsymbol{u}\sim P}[\boldsymbol{K}_h(\boldsymbol{u}-\boldsymbol{x})]\\
&\leq\sup_{\boldsymbol{x}\in\mathbb{R}^d}\int_{\boldsymbol{u}\in\mathbb{R}^d}\boldsymbol{K}_h^2(\boldsymbol{u}-\boldsymbol{x})\boldsymbol{k}_{\alpha,\mu,r}(\boldsymbol{u})d\boldsymbol{u}\\
&=\sup_{\boldsymbol{x}\in\mathbb{R}^d}\frac{1}{h^{2d}}\prod_{i=1}^{d}\int_{\mathbb{R}}K^2(\frac{u_i-x_i}{h})k_{\alpha,\mu,r}(u_i)du_i\\
&\overset{\boldsymbol{u}=\boldsymbol{x}+h\boldsymbol{v}}{=}\sup_{\boldsymbol{x}\in\mathbb{R}^d}\frac{1}{h^d}\prod_{i=1}^{d}\int_{\mathbb{R}}K^2(v_i)k_{\alpha,\mu,r}(x_i+hv_i)dv_i\\
&\leq\sup_{\boldsymbol{x}\in\mathbb{R}^d}\frac{1}{h^d}\prod_{i=1}^{d}\left(||k_{\alpha,\mu,r}||_\infty\int_{\mathbb{R}}K^2(v_i)dv_i\right)\\
&=\left(\frac{(1-\alpha+\frac{\alpha}{r})}{h}\right)^d\left(||k||_\infty\int_{\mathbb{R}}K^2(x)dx\right)^d:=C_1\sigma^2,
\end{aligned}$$

where $C_1=\left(||k||_\infty\int_{\mathbb{R}}K^2(x)dx\right)^d$ is constant only depends on k and K. Also note

$$\sup_{f\in\mathcal{F}}||f||_\infty=\sup_{\boldsymbol{x},\boldsymbol{u}\in\mathbb{R}^d}||\boldsymbol{K}_h(\boldsymbol{u}-\boldsymbol{x})||_\infty=||\boldsymbol{K}_h||_\infty=||K_h||_\infty^d=\frac{||K||_\infty^d}{h^d}.$$

Let $U = 2C_2(1-\alpha+\frac{\alpha}{r})^d\frac{1}{h^d}$, with $C_2 = \|k\|_\infty \|K\|_\infty$, then it is easy to verify that

$$\sup_{f\in\mathcal{F}} ||f||_\infty < U, \quad 0 < \sigma^2 < U/2,$$

since $\int K^2 \le \|K\|_\infty \int K = \|K\|_\infty$, and $\frac{1}{2}\|k\|_\infty < 1 < 1-\alpha+\frac{\alpha}{r}$.
Since both σ^2 and U do not depend on n, therefore condition $\sqrt{n}\sigma \ge U\sqrt{\log(\frac{U}{\sigma})}$ is satisfied for all n bigger than finite $n_0 := \frac{U^2}{\sigma^2}\log\frac{U}{\sigma}$. Consider $0 < \epsilon < C_0\frac{\sigma^2}{U}$, $\lambda = C_0$, and $n > (C_0^2 \vee 1)n_0$, we can finally apply Lemma 3.10 and get

$$\begin{aligned}\Pr\left\{\sup_{\boldsymbol{x}\in\mathbb{R}^d} |\widehat{\boldsymbol{K}}_h - \mathbb{E}\left[\widehat{\boldsymbol{K}}_h\right]| > \epsilon\right\} &= \Pr\left\{\sup_{f\in\mathcal{F}} |\sum_{i=1}^n (f(X_i) - f(X_1))| > \epsilon n\right\} \\ &\le C\exp\left\{-\frac{C_1\log(1+\frac{C_0}{4C})}{C_0 C}\frac{\epsilon^2 n h^d}{(1-\alpha+\frac{\alpha}{r})^d}\right\}.\end{aligned}$$

Let the right hand side equals δ, in turn we have, for δ small enough (solve the upper bound on ϵ to get the lower bound on δ),

$$\Pr\left\{\sup_{\boldsymbol{x}\in\mathbb{R}^d} |\widehat{\boldsymbol{K}}_h - \mathbb{E}\left[\widehat{\boldsymbol{K}}_h\right]| > \sqrt{\frac{C_3\log(C/\delta)(1-\alpha+\frac{\alpha}{r})^d}{nh^d}}\right\} \le \delta,$$

where $C_3 := \sqrt{\frac{C_0 C}{C_1\log(1+\frac{C_0}{4C})}}$.

Step 2. For the second term in (3.31), first we prove that if $k \in H(\beta, L)$, then $k_{\alpha,\mu,r} \in H(\beta, (1-\alpha+\frac{\alpha}{r^{\beta+1}})L)$. Note that for this argument, we are only considering the one-dimensional case, therefore

$$k \in H(\beta, L) \Longleftrightarrow \sup_x \left|\frac{d^\beta k(x)}{dx^\beta}\right| \le L. \tag{3.34}$$

Using the chain rule, we have

$$\begin{aligned}\frac{d^\beta k_{\alpha,\mu,r}(x)}{dx^\beta} &= (1-\alpha)\frac{d^\beta k(x)}{dx^\beta} + \frac{\alpha}{r}\frac{d^\beta k\left(\frac{x-\mu}{r}\right)}{dx^\beta} \\ &= (1-\alpha)\frac{d^\beta k(u)}{du^\beta}\,|_{u=x} + \frac{\alpha}{r^{1+\beta}}\frac{d^\beta k\,(u)}{du^\beta}\,|_{u=\frac{x-\mu}{r}}\,.\end{aligned}$$

Therefore using (3.34), we have

$$\sup_x \left|\frac{d^\beta k_{\alpha,\mu,r}(x)}{dx^\beta}\right| \le \left((1-\alpha) + \frac{\alpha}{r^{1+\beta}}\right)L,$$

that is $k_{\alpha,\mu,r} \in H\left(\beta, (1-\alpha+\frac{\alpha}{r^{\beta+1}})L\right)$.

Then we have

$$
\begin{aligned}
&\left\|\mathbb{E}\left[\widehat{\boldsymbol{K}}_h\right] - \boldsymbol{k}_{\alpha,\mu,r}\right\|_\infty \\
=&\sup_{\boldsymbol{x}} \left| \int \boldsymbol{K}_h\left(\|\boldsymbol{u}-\boldsymbol{x}\|\right) \boldsymbol{k}_{\alpha,\mu,r}(\boldsymbol{u}) d\boldsymbol{u} - \boldsymbol{k}_{\alpha,\mu,r}(\boldsymbol{x}) \right| \\
=&\sup_{\boldsymbol{x}} \prod_{i=1}^{d} \int K_h\left(\|u_i - x_i\|\right)\left(k_{\alpha,\mu,r}(u_i) - k_{\alpha,\mu,r}(x_i)\right) du_i \\
=&\sup_{\boldsymbol{x}} \left| \prod_{i=1}^{d} \int K\left(|v_i|\right)\left(k_{\alpha,\mu,r}(x_i+hv_i) - k_{\alpha,\mu,r}(x_i)\right) \right| \\
\leq&\sup_{\boldsymbol{x}} \prod_{i=1}^{d} \Bigg\{ \left| \int K\left(|v_i|\right)\left(k_{\alpha,\mu,r}(x_i+hv_i) - k^{x_i,\beta}_{\alpha,\mu,r}(x_i+hv_i)\right) \right| \\
&\qquad\qquad + \left| \int K\left(|v_i|\right)\left(k^{x_i,\beta}_{\alpha,\mu,r}(x_i+hv_i) - k_{\alpha,\mu,r}(x_i)\right) \right| \Bigg\} \\
\overset{(i)}{=}&\sup_{\boldsymbol{x}} \prod_{i=1}^{d} \left| \int K\left(|v_i|\right)\left(k_{\alpha,\mu,r}(x_i+hv_i) - k^{x_i,\beta}_{\alpha,\mu,r}(x_i+hv_i)\right) \right| \\
\overset{(ii)}{\leq}&\sup_{\boldsymbol{x}} \prod_{i=1}^{d} \left| \int K\left(|v_i|\right)\left((1-\alpha+\frac{\alpha}{r^\beta})Lh^\beta |v_i|^\beta\right) \right| \\
=&\left((1-\alpha+\frac{\alpha}{r^{\beta+1}})Lh^\beta \left| \int K\left(|v|\right)|v|^\beta \right|\right)^d \\
:=&(1-\alpha+\frac{\alpha}{r^{\beta+1}})^d h^{d\beta} C_4,
\end{aligned}
$$

where $\cdot^{x,\beta}$ is the taylor expansion of $\cdot$ at x to order $\beta-1$, and $C_4 := L^d \left|\int K\left(|v|\right)|v|^\beta\right|^d$. Specifically, (i) is true since $k_{\alpha,\mu,r} \in H(\beta, (1-\alpha+\frac{\alpha}{r^\beta})L)$, and therefore $\left(k^{x_i,\beta}_{\alpha,\mu,r}(x_i+hv_i) - k_{\alpha,\mu,r}(x_i)\right)$ is a polynomial of degree $\beta-1$, then use the fact that $K \in G(\beta)$, we have the second term is zero; and (ii) is true from the fact that $k_{\alpha,\mu,r} \in H(\beta, (1-\alpha+\frac{\alpha}{r^{\beta+1}})L)$.

Combining the above analysis, we have

$$
\Pr\left\{ \sup_{\boldsymbol{x}\in\mathbb{R}^d} |\widehat{\boldsymbol{K}}_h - \boldsymbol{k}_{\alpha,\mu,r}| > \sqrt{\frac{C_3 \log(1/\delta)(1-\alpha+\frac{\alpha}{r})^d}{nh^d}} + C_4\left(1-\alpha+\frac{\alpha}{r^{\beta+1}}\right)^d h^{d\beta} \right\} \leq \delta,
$$

where C_3, C_4 are constants that do not depend on h, α, μ, r, but depend on k, K, d, n, β, L. □

3.7.7 Robustness on the empirical level

Definition 3.11. (Empirical contamination model) Given n samples $\{(x_i, y_i)\}_{i=1}^n$, we consider the corresponding contaminated samples $\{(x_i', y_i')\}_{i=1}^n$ that satisfying

$$(x_i', y_i') = (x_i, y_i) \text{ for } 1 \le i \le d_n; \qquad (x_i', y_i') = (x', y') \text{ for } d_n + 1 \le i \le n,$$

where $1 \le d_n \ll n$ is the number of outliers.

Denote the empirical aLDG_t under the contamination model Definition 3.11 as $\widehat{\text{aLDG}}_t'$. We consider characterizing the following modified influence function (defined to adapt empirical setting)

$$\text{MIF}((x', y'), \widehat{\text{aLDG}}, F_n) := |\widehat{\text{aLDG}}_t' - \widehat{\text{aLDG}}_t|.$$

In Theorem 3.12 we give an upper bound on MIF, which depends on the number of outliers d_n and sample size n.

Theorem 3.12. Consider the contamination model in Definition 3.11 with d_n outliers, and the empirical $\widehat{\text{aLDG}}_t$[5] using boxcar kernel density estimator[6] with bandwidth h_n. Assume the point mass (x', y') is far away from all the n uncontaminated samples:

$$(x', y'): \quad \min_{j\in[n]} |x_j - x'| > h_n, \ \min_{j\in[n]} |y_j - y'| > h_n.$$

Under the same conditions on the true data distribution as in Theorem 3.5, then with high probability, we have

$$\begin{aligned}\text{MIF}((x', y'), \widehat{\text{aLDG}}, F_n) &:= \left|\widehat{\text{aLDG}}_t' - \widehat{\text{aLDG}}_t\right| \\ &< 2\epsilon_n + \eta_{n-d_n} + 2\sqrt{\epsilon_n + \eta_{n-d_n}}\sqrt{\frac{\log n}{n}},\end{aligned}$$

where $\epsilon_n := \frac{d_n}{n}$ is the contamination mass, and F_n denote the empirical distribution of the uncontaminated data.

Proof. Given n bivariate samples $(x_1, y_1), \ldots, (x_n, y_n)$, denote

$$\begin{aligned} T(x, y) &:= \frac{f_{X,Y}(x, y) - f_X(x) f_Y(y)}{\sqrt{f_X(x) f_Y(y)}}, \quad T_i := T(x_i, y_i); \\ \widehat{T}(x, y) &:= \frac{\widehat{f}_{X,Y}(x, y) - \widehat{f}_X(x) \widehat{f}_Y(y)}{\sqrt{\widehat{f}_X(x) \widehat{f}_Y(y)}}, \quad \widehat{T}_i := \widehat{T}(x_i, y_i)\end{aligned}$$

[5] Equation (17) of the main paper
[6] Equation (21) of the main paper

where $\widehat{f}_{X,Y}, \widehat{f}_X, \widehat{f}_Y$ are some density estimator for $f_{X,Y}, f_X, f_Y$. Then the empirical aLDG can be written as

$$\widehat{\mathrm{aLDG}}_t = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{\widehat{T}_i \geq t\right\},$$

Denote the density estimator under the contaminated model as $\widehat{f}'_X, \widehat{f}'_Y, \widehat{f}'_{XY}$, and the corresponding statistics as $\widehat{T}'_i$, and $\widehat{\mathrm{aLDG}}'_t$. First we have

$$\begin{aligned}
\widehat{f}'_X(\cdot) &= \frac{1}{n}\sum_{j=d_n+1}^{n} K_{h_n}(\cdot, x_j) + \epsilon_n K_{h_n}(\cdot, x') \\
\widehat{f}'_Y(\cdot) &= \frac{1}{n}\sum_{j=d_n+1}^{n} K_{h_n}(\cdot, Y_j) + \epsilon_n K_{h_n}(\cdot, y'), \\
\widehat{f}'_{XY}(\cdot,\cdot) &= \frac{1}{n}\sum_{j=d_n+1}^{n} K_{h_n}(\cdot, x_j)K_{h_n}(\cdot, y_j) + \epsilon_n K_{h_n}(\cdot, x')K_{h_n}(\cdot, y').
\end{aligned}$$

And consequently, for $d_n + 1 \leq i \leq n$,

$$\begin{aligned}
\widehat{f}'_X(x'_i) &= \widehat{f}_X(x_i) - \epsilon_n \frac{1}{d_n}\sum_{j=1}^{d_n} K_h(x_i, x_j) \\
\widehat{f}'_Y(y_i) &= \widehat{f}_Y(y_i) - \epsilon_n \frac{1}{d_n}\sum_{j=1}^{d_n} K_h(y_i, y_j) \\
\widehat{f}'_{XY}(x_i, y_i) &= \widehat{f}_{X,Y}(x_i, y_i) - \epsilon_n \frac{1}{d_n}\sum_{j=1}^{d_n} K_h(x_i, x_j)K_h(y_i, y_j).
\end{aligned}$$

We assume that the true marginal densities f_X and f_Y are bounded by some constant $c_{\max}$ and the corresponding density estimation error is uniformly bounded by η_n with high probability. Denote

$$\widehat{c}_{\max} := \max\Big\{\sup_x \frac{1}{d_n}\sum_{j=1}^{d_n} K_h(x, x_j), \quad \sup_y \frac{1}{d_n}\sum_{j=1}^{d_n} K_h(y, y_j), \quad \sup_{x,y} \frac{1}{d_n}\sum_{j=1}^{d_n} K_h(x, x_j)K_h(y, y_j)\Big\},$$

then we have

$$c_{\max} - \eta_{d_n} \leq \widehat{c}_{\max} \leq c_{\max} + \eta_{d_n},$$

with high probability. Consequently we have

$$\max_{d_n+1 \leq i \leq n} |\widehat{T}'_i - \widehat{T}_i| \leq \epsilon_n \widehat{c}_{\max} \leq \epsilon_n(c_{\max} + \eta_{d_n})$$

with high probability.

Therefore, for all i, with high probability, we can conclude

$$\begin{aligned}\widehat{T}_i \geq t + \epsilon_n(c_{\max} + \eta_{d_n}) \quad \text{or} \quad \widehat{T}_i < t - \epsilon_n(c_{\max} + \eta_{d_n}) \\ \implies \mathbf{1}\{\widehat{T}_i > t\} = \mathbf{1}\{\widehat{T}_i' > t\}.\end{aligned}$$

This implies, with high probability,

$$\begin{aligned}&|\widehat{\mathrm{aLDG}}_t' - \widehat{\mathrm{aLDG}}_t| \\ &\leq \epsilon_n + \frac{1}{n}\sum_{i=d_n+1}^{n} \mathbf{1}\{t - \epsilon_n(c_{\max} + \eta_{d_n}) < \widehat{T}_i \leq t + \epsilon_n(c_{\max} + \eta_{d_n})\} \\ &= \epsilon_n + (1-\epsilon_n)\Big(\widehat{P}_{n-d_n}(\widehat{S}_{t-\epsilon_n(c_{\max}+\eta_{d_n})}) - \widehat{P}_{n-d_n}(\widehat{S}_{t+\epsilon_n(c_{\max}+\eta_{d_n})})\Big) \\ &\leq \epsilon_n + (1-\epsilon_n)\Big(\widehat{P}_{n-d_n}(S_{t-\epsilon_n(c_{\max}+\eta_{d_n})-c\eta_{n-d_n}}) \\ &\qquad\qquad - \widehat{P}_{n-d_n}(S_{t+\epsilon_n(c_{\max}+\eta_{d_n})+c\eta_{n-d_n}})\Big) \\ &\leq \epsilon_n + (1-\epsilon_n)\Big(P(D_t) + |\widehat{P}_{n-d_n}(D_t) - P(D_t)|\Big),\end{aligned}$$

where

$$\begin{aligned}&\widehat{S}_t := \{(x,y) : \widehat{T} > t\}, \quad S_t := \{(x,y) : T > t\}, \\ &D_t := S_{t-\epsilon_n(c_{\max}+\eta_n)-c\eta_{n-d_n}} \setminus S_{t+\epsilon_n(c_{\max}+\eta_n)+c\eta_{n-d_n}}.\end{aligned}$$

Since we assume that aLDG_t is L-Lipschitz smooth around t, therefore

$$P(D_t) \leq 2L(\epsilon_n(c_{\max} + \eta_{d_n}) + c\eta_{n-d_n}) \asymp O(\epsilon_n + \eta_{n-d_n}) \to 0.$$

Then using the Bernstein inequality for Bernoulli variable with mean $P(D_t) \ll 1$, with high probability we have

$$\begin{aligned}|\widehat{P}_{n-d_n}(D_t) - P(D_t)| &\leq \sqrt{\frac{P(D_t)\log(n-d_n)}{n-d_n}} \\ &\lesssim \sqrt{\frac{(\epsilon_n + \eta_{n-d_n})\log n}{n-d_n}} = \sqrt{\frac{\epsilon_n + \eta_{n-d_n}}{1-\epsilon_n}}\sqrt{\frac{\log n}{n}}.\end{aligned}$$

Combine the above results, with high probability we have,

$$\begin{aligned}|\widehat{\mathrm{aLDG}}_t' - \widehat{\mathrm{aLDG}}_t| &\lesssim \epsilon_n + (1-\epsilon_n)\Big(\epsilon_n + \eta_{n-d_n} + \sqrt{\frac{\epsilon_n + \eta_{n-d_n}}{1-\epsilon_n}}\sqrt{\frac{\log n}{n}}\Big) \\ &< 2\epsilon_n + \eta_{n-d_n} + 2\sqrt{\epsilon_n + \eta_{n-d_n}}\sqrt{\frac{\log n}{n}}.\end{aligned}$$

Finally, we can conclude, if the contamination mass $\epsilon_n \to 0$ as $n \to \infty$, and satisfy $\epsilon_n = O(\eta_n \vee \frac{\log n}{n})$, then with high probability, we have

$$\mathrm{MIF}((x', y'), \widehat{\mathrm{aLDG}}, F_n) < 2\epsilon_n + \eta_{n-d_n} + 2\sqrt{\epsilon_n + \eta_{n-d_n}}\sqrt{\frac{\log n}{n}} \ll 1,$$

which goes to zero as n goes to infinity. □

3.7.8 Discussion on thresholding methods

Another intuitive way we found for selecting t is based on the curve of $aLDG_t$ versus t. This function tends to decrease rapidly near zero and then reaches an inflection point, after which it declines very slowly (e.g., Figure 3.13). We propose selecting the threshold t to be the inflection point $t^\star$. Since the increment of t around $t^\star$ is suddenly unable to reduce further aLDG_t much, therefore, we expect this choice to strike a balance between robustness and sensitivity. To stabilize the estimation of such inflection point, we use the median of estimated inflection point from $\max\{\lfloor 1000/n \rfloor, 5\}$ different random shuffles as the final estimation. We call this t selection method the *inflection point* method.

In Figure 3.13 and Figure 3.14, we compare the above three proposed methods of selecting t. We use 18 different bivariate distributions to make the comparison (see Figure 3.3 for the explicit display of each distribution). We believe this series of distributions are representative enough as it covers cases from linear to nonlinear, monotone to nonmonotone, and also probabilistic mixtures. We find that the *asymptotic norm* method is often too conservative given the small sample size. In contrast, the *uniform error* and *inflection point* method are often similar to each other. On the other hand, Figure 3.14 shows that *uniform error* method gives more stable value than the *inflection point* method, while *asymptotic norm* is the most stabilised among the three. Therefore in practice, we recommend people use *uniform error* over *asymptotic norm* when the sample size is not too big (e.g., no bigger than 200); while using *asymptotic norm* when the sample size is big enough (e.g., bigger than 200) and the computation budget is limited.

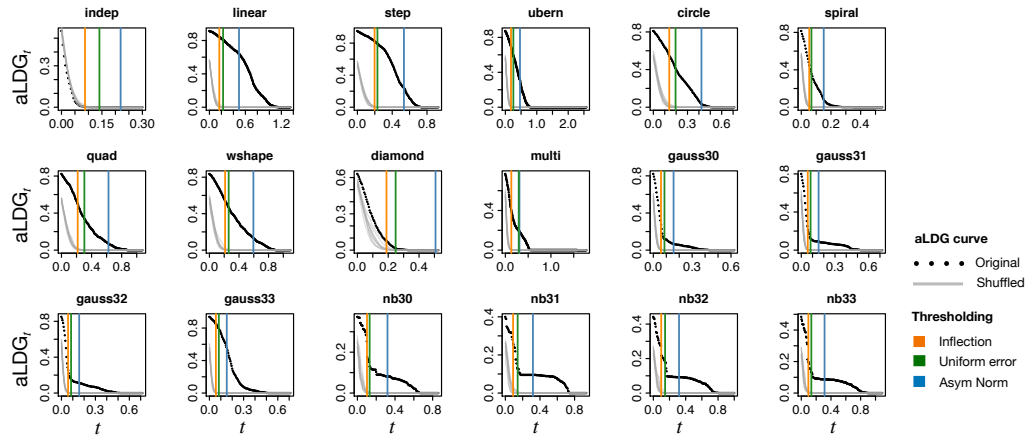


Figure 3.13: The curves of aLDG$_t$ versus t estimated by 1000 samples. Each plot represents different bivariate distribution annotated by the subtitle (see Figure 3.3 for explicit display of each distribution). In each plot, the black dot curve represents the aLDG$_t$ estimated using original data samples, and the gray dot curves represent the aLDG$_t$ estimated using shuffled data samples (one curve each random shuffle, 20 curves in total); The vertical lines represent different choices of the thresholding: the orange one represents the *inflect point* method; the green one represents the *uniform error* method; and the blue one represent the *asymptotic norm* one.

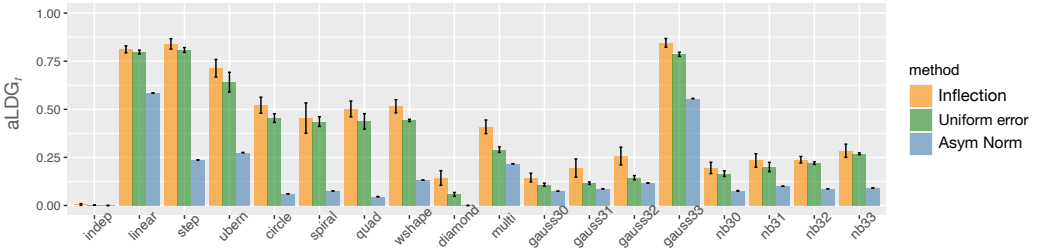


Figure 3.14: The value of aLDG$_t$ estimated by 1000 samples using different method of choosing t. The x-axis represents different bivariate distribution (see Figure 6 in the main paper for explicit display of each distribution). For each distribution, we show the mean value of aLDG$_t$ over 20 trials with error bar, where different thresholding method is annotated by different color.

3.7.9 Detailed example for merits of thresholding

Consider the following product kernel density mixture:

$$f_X(x) = \alpha k_{0,r}(x) + (1-\alpha)k_{0,1}(x), \quad f_Y(y) = k_{0,r}(y) + (1-\alpha)k_{0,1}(y),$$
$$f_{XY}(x,y) = \alpha k_{0,r}(x)k_{0,r}(y) + (1-\alpha)k_{0,1}k_{0,1},$$

where $\alpha \in (0,1)$, $0 < r \leq 1$ and $k_{\mu,r}(\cdot) := \frac{1}{r}k(\frac{\cdot-\mu}{r})$, with k as the density of a one dimensional uniform distribution supported on $[-1,1]$.

With $\alpha/r \to \infty$, $\alpha \to 0$ and $r \to 0$, we have

$$\mathbb{E}\left[\frac{f_{XY}(X,Y) - f_X(X)f_Y(Y)}{\sqrt{f_X(X)f_Y(Y)}} \;\middle|\; |X| < r \;\&\; |Y| < r\right] \approx \frac{\alpha(1-\alpha)/r^2}{\alpha/r} = \frac{1-\alpha}{r}$$

and

$$\mathbb{E}\left[\frac{f_{XY}(X,Y) - f_X(X)f_Y(Y)}{\sqrt{f_X(X)f_Y(Y)}} \;\middle|\; |X| > r \text{ or } |Y| > r\right] \approx -\frac{\alpha(1-\alpha)/r}{\alpha/r} = \alpha - 1,$$

$$\mathbb{E}\left[\frac{f_{XY}(X,Y) - f_X(X)f_Y(Y)}{\sqrt{f_X(X)f_Y(Y)}} \;\middle|\; |X| > r \;\&\; |Y| > r\right] \approx -\frac{(1-\alpha)\alpha}{1-\alpha} = -\alpha,$$

therefore using the law of total expectation, we finally have

$$\mathbb{E}\left[\frac{f_{XY}(X,Y) - f_X(X)f_Y(Y)}{\sqrt{f_X(X)f_Y(Y)}}\right] \approx p_1\frac{1-\alpha}{r} + p_2(\alpha - 1) + p_3\alpha, \tag{3.35}$$

where

$$\begin{aligned}
p_1 &:= \Pr\{|X| \leq r \;\&\; |Y| \leq r\} = \alpha + (1-\alpha)r^2,\\
p_2 &:= \Pr\{(|X| > r \;\&\; |Y| \leq r) \textit{ or } (|X| \leq r \;\&\; |Y| > r)\} = (1-\alpha)(2r - 2r^2),\\
p_3 &:= \Pr\{|X| > r \;\&\; |Y| > r\} = (1-\alpha)(1 - 2r + r^2).
\end{aligned}$$

Simplifying (3.35) we have,

$$\mathbb{E}\left[\frac{f_{XY}(X,Y) - f_X(X)f_Y(Y)}{\sqrt{f_X(X)f_Y(Y)}}\right] \approx \frac{\alpha}{r} \to \infty.$$

3.7.10 Supplementary figures

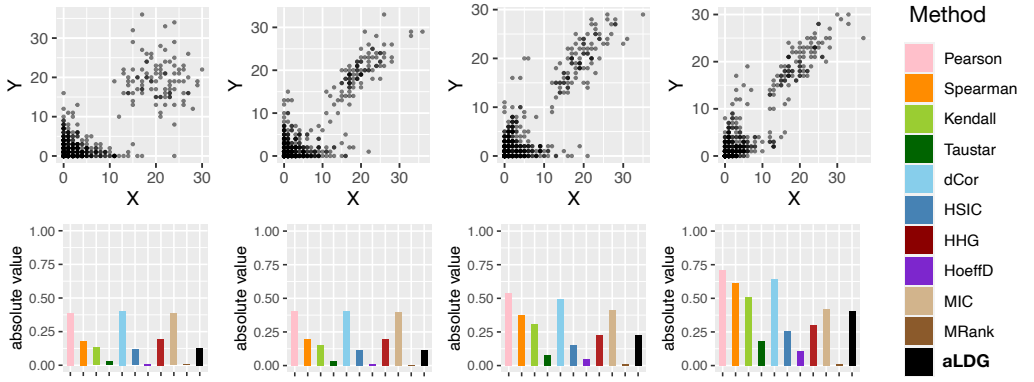


Figure 3.15: Empirical aLDG value for Negative Binomial mixture. The upper row shows the scatter plot, while the lower row shows the corresponding dependence level given by different measures. The data are generated as a three-component Negative Binomial mixture. From left to right there are 0,1,2,3 out of 3 components has correlation 0.8, while the rest has correlation 0, i.e. the dependence level increases from left to right.

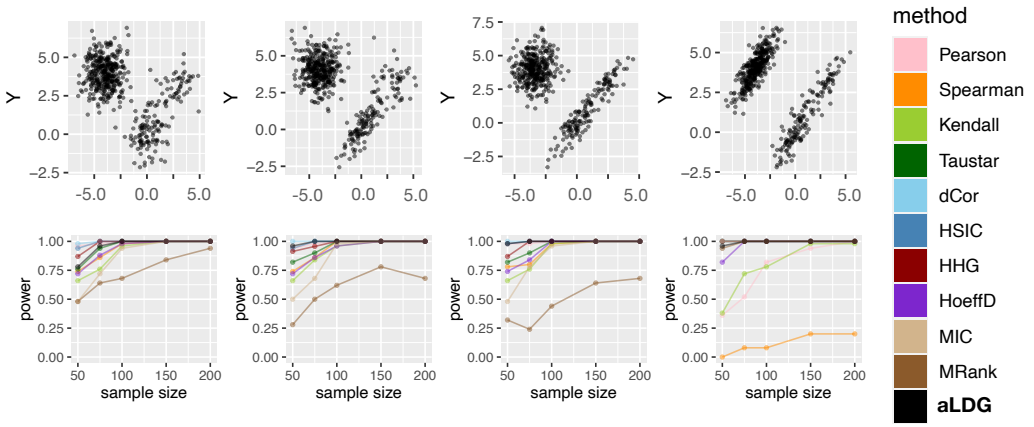


Figure 3.16: The empirical power of permutation test at level 0.05, based on different dependency measures under different Gaussian mixture distributions and sample sizes. The power is estimated using 50 independent trials. The data are generated as a three-component Gaussian mixture. From left to right the overall dependence level increases: specifically, 0,1,2 and 3 of the 3 components have correlation of 0.8, while the remaining components have no correlation.

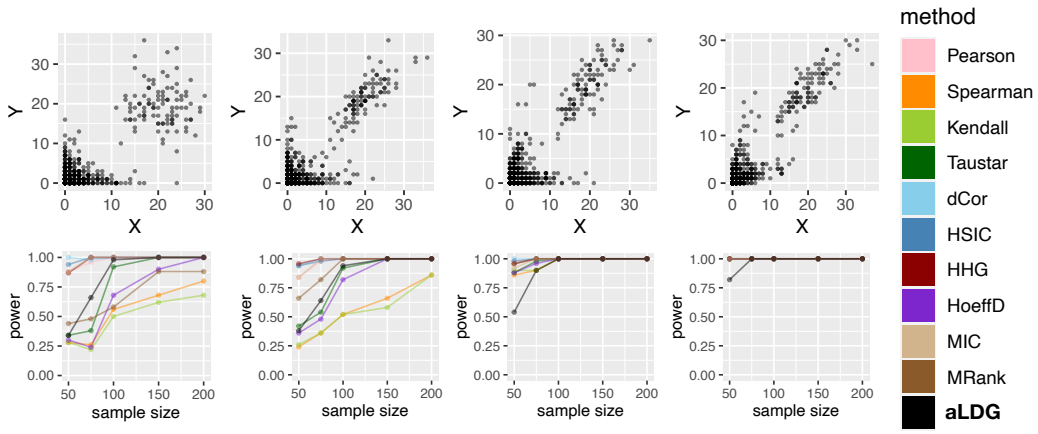


Figure 3.17: The empirical power of permutation test at level 0.05, based on different dependency measures under different negative binomial mixture distributions and sample sizes. The power is estimated using 50 independent trials. The data are generated as three-component Negative Binomial mixture. From left to right the overall dependence level increases: specifically, 0,1,2 and 3 of the 3 components have correlation of 0.8, while the remaining components have no correlation.

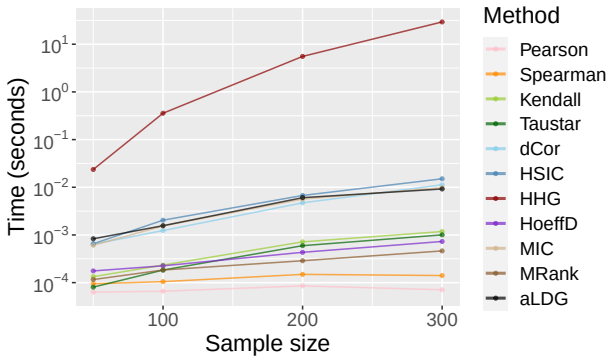


Figure 3.18: Computation time ($\log_{10}$ scaled) versus sample size for different methods, averaged over 10 independent trials. We can see that HHG is much slower than the others as sample size grows, while aLDG is roughly as fast as dCor, HSIC, MIC.

Four

Identifying active differential expression genes in Autism

In an effort to better understand autism spectrum disorders (ASD), we focus on two types of ASD-related genes: Differential Expression (DE) genes which are differentially expressed in ASD versus neurotypical brains, and TADA genes, which are identified by unusual patterns of genetic mutations. While TADA genes are thought to be "active", DE genes are thought to be either "active" (cause of ASD) or "reactive" (outcome of ASD). In this project, we aim to dive deep into the mechanism of DE genes: discriminating the "active" ones from "reactive". Relying on the conjecture that active DE gene modules are enriched with TADA genes, while reactive ones are not, we adopt a network-assisted approach to bridge these two sources of information and identify an assortment of unique "active" and "reactive" DE gene communities. Our work brings new insights toward understanding the role genes play in the development of ASD and how ASD affects gene expression as well.

4.1 Introduction

The autism spectrum disorder (ASD) is a clinically heterogeneous class of neurodevelopmental disorders that has a strong genetic basis. Over the past decade, extensive studies have led to the identification of numerous susceptibility genes. These genes were discovered through various approaches, including the analysis of unusual patterns of genetic mutations in DNA. The TADA method (He et al., 2013), in particular, has been instrumental in identifying ASD susceptibility genes by examining the frequency of de novo and transmitted mutations in parent-offspring trios. This approach has proven to be highly effective, especially when applied to exome or whole-genome sequencing data from large family samples. Still, among thousands of trios sequenced, only a few hundreds of genes were deemed as ASD risk genes, and preliminary studies suggest there should be nearly a thousand ASD risk genes (Neale et al., 2012; He et al., 2013). In this project, we look at this problem using a novel integrative perspective: with another source of information,

we utilize gene networks to bridge it with existing TADA results and aim to extract new insights about ASD mechanisms.

Specifically, we consider gene expression data as our additional information source. Gene expression data has revealed many (over four thousands) genes that are differentially expressed (DE) in ASD versus neuro typical (NT) or say control brains (Gandal et al., 2022). Interesting functional modules are identified for these DE genes, however, what roles these DE genes play in ASD remains unknown: a gene can be differentially expressed to cause the phenotype ("active"), or it can be differentially expressed because of the phenotype ("reactive"). On the contrary, analysis using genetic mutations in the DNA like TADA outputs genes known to be the cause of the phenotype. Therefore, one natural idea is to deconvolve the DE mechanism with the help of TADA results. The "active" DE genes might be a new group of candidates for ASD risk genes.

One might think about just doing a simple set diff on those two sets of significant genes, and directly treating DE genes that are also TADA significant as active and the rest as reactive. However, we find that there is not much overlap between the DE and TADA significant genes when just taking a cutoff of their q-values [1] to determine significance (see Table 4.1). So this simple approach is not good enough for interesting findings, and a bridge to connect these two information sources seems necessary.

cutoff	0.001	0.005	0.01	0.05	#TADA
0.001	1	1	2	3	72
0.005	6	8	10	15	97
0.01	9	12	14	22	111
0.05	19	26	31	48	185
#DE	720	1517	2077	4223	

Table 4.1: The overlaps between DE and TADA significant genes. The first row and first column indicate four different cutoffs on the q-values, and each cell in the table represents the number of overlapping DE and TADA significant genes when taking different cutoffs at their respective q-values to determine significance. The last row and last column represent the number of DE and TADA significant genes separately. We use DE results from Gandal et al. (2022), and TADA results from Fu et al. (2022).

Observations in previous studies motivate us to look at gene networks as the bridge and conduct analysis on the module level. Specifically, Willsey et al. (2013a) found that ASD-related genes tend to cluster meaningfully in a gene network derived from gene expression in the developing brain, compared with other genes. Later, many have conjectured that network derived from gene expression can be utilized

[1] A q-value is a p-values that has been adjusted for the False Discovery Rate (FDR) control.

to discover ASD risk genes (Liu et al., 2014, 2015; Xie et al., 2022), and indeed many novel ASD risk gene discoveries are made with joint modeling of gene network and gene risk using a Hidden Markov Random Field Model (HMRF). As for our task, we additionally observe that for gene networks appropriately constructed, there exist gene communities that are solely enriched with DE-significant genes; and also communities that are enriched with both TADA-significant genes (TADA genes for short) and DE-significant genes (DE genes for short). Naturally, we conjecture that the communities that have many TADA genes and DE genes as members tend to "affect" the etiology of ASD (i.e. active communities); whereas those communities solely comprised of DE genes tend to be an "outcome" of ASD ("reactive communities").

Based on these observations, we propose to first use gene network to regularize the DE and TADA information and then search for clusters that satisfy our target signal distribution. Figure 4.1 demonstrate the whole analysis pipeline of our method, which involves three major steps: 1) gene network construction 2) network regularization of DE and TADA signal 3) active and reactive community detection. We develop novel methods in each of these three steps, to address long-overlooked or newly-appeared challenges.

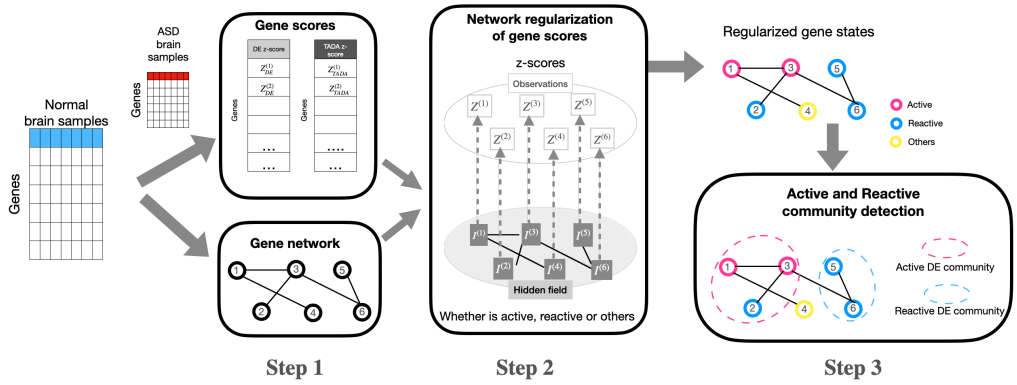


Figure 4.1: The whole pipeline of our method.

In the first step, the choice of gene network type is a crucial decision in the analysis of ASD risk gene modeling. Previous studies, such as Liu et al. (2014) and Liu et al. (2015), have focused on specific types of gene networks without conducting a systematic evaluation of different network concepts. For instance, (Liu et al., 2014) used Pearson correlation to measure gene co-expression and constructed gene networks using Weighted Gene Co-expression Network Analysis (WGCNA) (Langfelder and Horvath, 2008). On the other hand, Liu et al. (2015) used partial correlation as a gene co-expression measurement for a sparser and more interpretable graph structure. Xie et al. (2022), on the other hand, directly utilized

downstream protein-protein interaction networks from existing databases. However, these approaches have overlooked two important aspects in this domain. Firstly, they do not adequately capture the nonlinear nature of gene interactions, which is increasingly recognized as common in modern large heterogeneous datasets (Tian et al., 2022). Secondly, they do not effectively account for the fact that genes often work together in groups or pathways, with potential overlap between pathways (Wang et al., 2015). Assessing group-level interactions becomes challenging when the true gene groups are unknown. Motivated by these limitations, we contribute the first-ever systematic investigation of various gene network concepts in the application of ASD risk gene modeling, taking into account nonlinearity and group interaction. For completeness, we propose two novel gene network construction methods. One is the aLDG proposed in Chapter 3 which measures the nonlinear marginal gene relationship, another called Ensemble nonlinear Partial correlation using Additive CCA (EnPAC) is in Section 4.3.2 which extends an existing idea addressing the challenges of measuring gene group level interaction when the true gene groups are unknown to nonlinear setups.

To jointly model differential expression (DE) and Transmission and De Novo Association (TADA) signals in the context of gene networks, we build upon the original Hidden Markov Random Field (HMRF) approach proposed in Liu et al. (2014). In our work, we extend the hidden states of genes to represent three classes: "reactive", "active" and "others". We believe that true reactive and active genes are likely to be clustered together in the gene network, meaning that the "active" and "reactive" hidden state of a gene can be inferred from the states of its neighboring genes. To leverage this clustering tendency, we update the posterior probability of each gene's hidden status based on the observed DE and TADA z-scores of its neighbors. By incorporating information from the gene network, we obtain a more regularized estimate of hidden states of genes. After obtaining the jointly regularized gene states, we employ a graph clustering approach that we have developed by adapting a popular existing method to our specific setting. We specifically aim to find gene groups enriched with active genes (our target active gene communities) and enriched with reactive genes (our targeted reactive gene communities).

Using our method, we are able to identify a series of active and reactive gene communities that are biologically interesting. We find that the identified active clusters are related to synaptic and neuronal functions, and are enriched in neuron-type cells, which agrees with the common belief that ASD is caused by misfunction of neuronal activities. On the other hand, the reactive clusters are mostly related to responsive functions and are enriched in nonneuron-type cells, which brings new insights into the effect of ASD on the misfunction of nonneuronal activities.

In Section 4.2, we introduce the related work in more detail; then we describe the core methods we used in our analysis in Section 4.3. Finally, in Section 4.4, we present the results we get using two recently published real datasets.

4.2 Related work

Differential expression analysis. Differential expression analysis is a fundamental computational method extensively employed in the field of transcriptomics to elucidate gene expression variations across different experimental conditions. By comparing the transcriptome profiles of distinct biological samples, this analysis aims to identify genes whose expression levels significantly differ between groups, thereby revealing key regulatory mechanisms underlying biological processes. Differential expression analysis encompasses a series of statistical techniques, including normalization, data transformation, hypothesis testing, and multiple testing correction. For our study here, the key to DE analysis is just a two-sample testing problem. Gandal et al. (2022) represents the most recent quality work in this direction. In the study conducted by Gandal et al. (2022), the focus is on differential expression (DE) analysis, which involves comparing gene expression patterns between different groups. They collected bulk RNA-seq samples from various brain regions and performed a comprehensive investigation of ASD differential expression. To identify differentially expressed genes in individuals with Autism Spectrum Disorder (ASD) compared to neurotypical individuals, Gandal et al. (2022) employed a regression model. This model included both technical variables (e.g., batch effects) and biological variables (e.g, region, age, sex, diagnosis, etc) as covariates. One of the key covariates is the variable "diagnosis", which indicates whether an individual has ASD or not. The differential expression analysis was conducted by examining the significance of the coefficients associated with the "diagnosis" covariate in the regression model. If the coefficient is significantly nonzero, it indicates that the gene's expression is differentially expressed between individuals with ASD and neurotypical individuals.

TADA analysis. The method developed by He et al. (2013), known as Transmission And De novo Association (TADA), represents the most successful and widely used analysis framework in inferring ASD risk genes from genetic mutation patterns. ADA incorporates various types of genetic variations, including de novo mutations, inherited variants, and case-control variant data. TADA employs a gene-based parametric likelihood model that involves estimating parameters for allele frequencies and gene-specific penetrances. The inference process relies on a Hierarchical Bayes strategy, which leverages information across all genes to estimate parameters that would be challenging to determine for individual genes alone. By borrowing strength across genes, TADA improves the estimation of these parameters and enhances the power to identify ASD risk genes. Compared to other common methods used for gene-based association analysis, TADA demonstrates significantly higher statistical power. Its effectiveness has led to its widespread adoption in subsequent research studies, especially when multiple types of WES data are available.

Gene network estimation. The true biological networks are of form of a directed network, which describes how a collection of molecular regulators interact with each other and with other substances in the cell to govern the gene expression levels of mRNA and proteins which, in turn, determine the function of the cell. These networks, called genetic regulatory networks (GRNs), are central to all biological organisms, and their deciphering is crucial to understand the development, functioning, and pathology of these organisms. Once a remote theoretical possibility, this deciphering is now made possible by advances in genomics, most notably high-throughput profiling of gene expression patterns with DNA microarrays and RNA sequencing (Karlebach and Shamir, 2008; Delgado and Gómez-Vela, 2019; Mercatelli et al., 2020; Nguyen et al., 2021). These advances have prompted the development of a plethora of models of GRNs and algorithms to reverse-engineer them from expression data. On one aspect, there are physical models mimicking the biological mechanisms at play, including promoter recognition, mRNA transcription, and protein translation. These models, typically based on systems of ordinary or stochastic differential equations (Cao et al., 2012; Dibaeinia and Sinha, 2020a), can generate realistic behavior but a large number of experimental data since they tend to have high-dimensional parameter spaces.

On the other hand, models based on the statistical analysis of dependencies between expression patterns have intermediate complexity and have already been successfully applied to aid in the inference of large gene regulatory networks (GRNs). There are methods that utilize bivariate dependencies between the expression patterns of all pairs of genes to infer "coexpression networks" (Langfelder and Horvath, 2008; Reshef et al., 2011). However, pairwise (or marginal) gene relationships fail to capture more complex statistical dependencies, such as higher-order interactions. Various refinements have been proposed to measure group-level interactions in coexpression networks, where the relationship between a pair of genes is assessed after conditioning on a group of other known functional-related genes (i.e., pathways) (Toh and Horimoto, 2002; Kim et al., 2012; Wang et al., 2015). Although both pairwise and group-level gene relationships are directional, as the GRN should be, they provide reliable candidates for later causal structure discovery, which is often computationally expensive (Vowels et al., 2022). In this project, we focus on statistical approaches for estimating undirected gene networks, as the estimation of directional gene networks is currently beyond the scope of our computational resources.

Network assisted ASD risk gene identification. Gene-based tests, such as TADA, often yield only a small number of genes with p-values that meet the threshold for genome-wide significance. However, when considering the gene interaction network, it is observed that certain genes with low individual p-values tend to cluster together (Liu et al., 2014). Although these genes may not be individually significant, the presence of such clustering of small p-values in the network is unlikely to occur by

chance. To improve the detection of risk genes, (Liu et al., 2014, 2015; Xie et al., 2022) employ an HMRF model to identify risk genes by discovering those that are clustered with other known significant genes in the provided gene networks.

4.3 Methods

	Marginal	**Partial**	**Ensemble Partial**
Linear	Pearson	PNS (Liu et al., 2014)	EPC (Wang et al., 2015)
Nonlinear	aLDG (Chapter 3)	GENIE3 (Huynh-Thu et al., 2010)	EnPAC (Section 4.3.2)

Table 4.2: All the gene network concepts we considered.

4.3.1 Selective review of statistical gene network estimation

Marginal gene relationship. Most methods for inferring edges in gene networks are based on the notion of measuring pairwise gene expression profile similarity or co-expression, which aims to estimate marginal relationships between pairs of genes. For pair of genes X, Y, the simplest measure is the classical Pearson's correlation:

$$\text{Pearson's } \rho(X,Y) := \frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}. \tag{4.1}$$

Plugin the sample estimation of covariance and variance, consistency, and asymptotic normality can be proven using the law of large numbers and the central limit theorem, respectively. Pearson's ρ has been, and probably still is, the most extensively employed measure in genetic applications, due to its simplicity. However, it is known to detect only linear relationships.

There are many works in the literature for detecting nonlinear pairwise gene relationships (see (Tian et al., 2022) for a review), here we adopt the measure called averaged Local Density Gap due to its superior performance compared with others in the extensive simulations and real data analysis conducted by (Tian et al., 2022). For a pair of genes X, Y, assume joint and marginal densities both exist, denote f_{XY}, f_X, f_Y as their joint and marginal densities. Then averaged Local Density Gap (aLDG) (Tian et al., 2022) measure is then defined as

$$\begin{aligned} \text{aLDG}_t &:= \Pr_{X,Y}\left\{T(X,Y) > t\right\}, \\ &\text{where } T(X,Y) := \frac{f_{X,Y}(X,Y) - f_X(X)f_Y(Y)}{\sqrt{f_X(X)f_Y(Y)}} \end{aligned} \tag{4.2}$$

and $t \geq 0$ is a tunable hyper-parameter that can be set in a data-dependent way using the principle of eliminating estimation noise. (Tian et al., 2022) show that aLDG can accumulate local dependence and can detect any non-linear, non-monotone relationship. Together with a consistent nonparametric estimator, they also establish the robustness of aLDG on both the population and empirical levels.

Partial gene relationship. The previous approaches for estimating gene dependencies mainly focus on pairwise relationships, neglecting higher-level interactions. However, in biological pathways, genes can interact with groups of genes, even if their marginal relationships are weak. To capture these higher-level interactions, partial correlations have been utilized. In the current literature, partial correlations are typically calculated conditioned on either all available genes or a pre-defined subset that may contain biologically unrelated genes. Gaussian graphical models (GGMs) offer a more realistic approach to modeling these interactions in linear settings. By assuming that gene expression levels follow a multivariate normal distribution, the conditional independence structure can be inferred by estimating the support of the inverse covariance matrix of the expression data. One can prove that estimating such support is equivalent to the neighborhood selection task, which uses regression techniques to select pairs of genes with nonzero coefficients. However, a challenge arises when applying this model in genetic practice due to the limited number of expression samples compared to the number of genes[2]. To improve estimation precision in very high-dimensional cases, DAWN (Liu et al., 2014) introduced a neighborhood pre-screening approach along with the LASSO-based method called PNS (partial neighborhood selection). As a natural extension to linear partial relationships, GENIE3 (Huynh-Thu et al., 2010) decomposes the prediction of a network between p genes into p separate regression problems. Each regression problem predicts the expression pattern of a target gene using the expression patterns of all other genes as inputs, employing tree-based ensemble methods such as Random Forests or Extra-Trees. The importance of an input gene in predicting the target gene's expression pattern is considered an indication of a potential linkage. By aggregating these putative linkages across all genes, a ranking of interactions is generated, allowing for the reconstruction of the entire network.

Ensemble partial gene relationship. As highlighted by (De La Fuente et al., 2004; Kim et al., 2012), including irrelevant genes in the conditioning set during partial relationship estimation can introduce false dependencies and lead to erroneous edges in the estimated network. This is due to the fact that the underlying gene interaction mechanism follows a causal graph, and conditioning on a collider can introduce spurious dependencies. To address this issue, researchers have explored the use of lower-order partial correlations, which condition on one or two other genes or a small set of known pathway genes. The work by (Wang et al., 2015) represents the state-of-the-art in this field, addressing the challenge through an unsupervised approach involving sparse canonical correlation analysis combined with repeated random partition and subsampling. Their method aims to identify strong linear relationships among a small subset of candidate genes. By applying canonical

[2]In our analysis, we aim to estimate a network for approximately 8000 genes with around 300 samples.

correlation analysis to randomly partitioned gene groups and averaging the linear coefficients over iterations, an edge weight matrix is constructed to capture the aggregated level of partial gene interactions of different orders. Sparsity is further incorporated to filter out weak or noisy relationships. However, their approach is limited to linear settings, and no nonlinear variants have been proposed thus far. In Section 4.3.2, we extend their approach to nonlinear settings by adopting an additive canonical correlation analysis model with group sparsity.

4.3.2 EnPAC: Ensemble nonlinear partial relationship

To provide a more comprehensive understanding of the original linear method developed by Haiyan et al. (Wang et al., 2015), which serves as the foundation for our endeavor to extend it to nonlinear settings, it is necessary to delve into its specific details. Algorithm 4.1 outlines the main steps of their approach, which aims to capture the ensemble partial relationship by traversing through different conditional sets of genes and computing an aggregated measure of partial correlations of varying orders.

Their algorithm involves solving a sparse Canonical Correlation Analysis (CCA) problem, which we will briefly recap its most general non-sparse version here Hotelling (1936). Consider data matrices $\boldsymbol{X} \in \mathbb{R}^{n\times p}$, $\boldsymbol{Y} \in \mathbb{R}^{n\times q}$, CCA aim to find linear combinations of $\boldsymbol{X}$ and $\boldsymbol{Y}$ columns which have a maximum correlation with each other:

$$\max_{\boldsymbol{u}_X\in\mathbb{R}^{p\times 1},\boldsymbol{u}_Y\in\mathbb{R}^{q\times 1}} corr(X\boldsymbol{u}_X, Y\boldsymbol{u}_Y),$$

where $corr(a,b) := \frac{\mathrm{Cov}(a,b)}{\sqrt{\mathrm{Var}(a)\mathrm{Var}(b)}}$ is the correlation. Sparse CCA used in Wang et al. (2015) additionally imposes sparsity in $\boldsymbol{u}_X$ and $\boldsymbol{u}_Y$. For more discussion on computation and theory about sparse CCA, we refer readers to (Gao et al., 2015; GAO et al., 2017; Wang and Zhou, 2021).

The utilization of CCA in Algorithm 4.1 can be framed within a regression context. In the case of a functional gene group $(x_1, \ldots, x_k)$, where the expression levels are assumed to follow a multivariate normal distribution, regressing x_i on the remaining genes yields $x_i = \sum_{j\neq i} \beta_{ij} x_j$. Consequently, for partitions that result in such configurations (e.g., 1 versus $k-1$), the elements in the weight vector $\boldsymbol{u}$ are proportional to β_{ij}, and thus indicative of the correlations between pairs of genes conditioned on the other genes within the same group (and selected within the same subsample). When considering more general configurations, such as having l genes in one set versus $k-l$ genes in the other set, the weight vector $\boldsymbol{u}$ is proportional to the correlation between a gene and a linear combination of the genes in the other set, conditioned on the remaining genes in the same set. By averaging the weight vectors over multiple iterations of random subsampling and partitioning, an aggregated measure of partial correlations of different orders is obtained. This iterative process allows for the exploration of all possible dependent sets, resulting in a comprehensive assessment of gene interactions.

Algorithm 4.1 Aggregated partial correlation.

Input: Gene expression matrix $\boldsymbol{X} \in \mathbb{R}^{p\times n}$, subsampling ratio α, sampling times K, partition times M.

Output: A $p \times p$ matrix E indicating the pairwise ensemble partial correlation strength among all the genes.

for $k = 1, \ldots, K$ **do**
  Random sample $\lfloor \alpha p \rfloor$ genes;
  for $m = 1, \ldots, M$ **do**
    Random split the sampled genes into two sets $A^{(m)}$ and $B^{(m)}$.
    (Denote $|A^{(m)}| = p_A$, $|B^{(m)}| = p_B$, and submatrices as $X_{A^{(m)}} \in \mathbb{R}^{p_A\times n}$ and $X_{B^{(m)}}\mathbb{R}^{p_B\times n}$).
    Solve sparse CCA for $X_{A^{(m)}}$ and $X_{B^{(m)}}$ and get $\boldsymbol{u}_A^{(m)} \in \mathbb{R}^{p_A\times 1}, \boldsymbol{u}_B^{(m)} \in \mathbb{R}^{p_B\times 1}$.
    Recover gene-level coefficients $\boldsymbol{u}^{(m)} \in \mathbb{R}^{p\times 1}$:

$$\boldsymbol{u}^{(m)} = \boldsymbol{e}^{(m)}/||\boldsymbol{e}^{(m)}||_2, \quad \text{where } \boldsymbol{e}_i^{(m)} = \begin{cases} \boldsymbol{u}_{A,i_A}^{(m)}, & \text{if } i \in A^{(m)} \\ \boldsymbol{u}_{B,i_B}^{(m)}, & \text{if } i \in B^{(m)} \\ 0, & \text{otherwise} \end{cases}$$

    $E^{(m)} = \boldsymbol{u}^{(m)}(\boldsymbol{u}^{(m)})^\top$.
  end
  $E = \frac{1}{M}\sum_{m=1}^{M} E^{(m)}$.
end

Additive CCA formulation. Naturally, we can have a nonlinear version of Algorithm 4.1 by replacing the CCA step in Algorithm 4.1 with its nonlinear variant. There have been many works in nonlinear CCA, which mainly falls into three categories: Kernel CCA (Bach and Jordan, 2002; Chang et al., 2013; Yoshida et al., 2017), which transform the feature space to kernel space; Functional CCA (Balakrishnan et al., 2012), which extends the linear combination in CCA to nonlinear additive models; and HSIC CCA (Chang et al., 2013; Uurtio et al., 2018), which modifies the dependence measure in CCA from correlation to HSIC(Gretton et al., 2005). In this paper, we mainly follow the Functional CCA route, as kernel CCA lacks of interpretation in the original feature space, and HSIC CCA runs rather slowly in practice. In the following, we formulate our explicit objective function as well as optimization processes. Our approach can be treated as a simplified but more practical version of (Balakrishnan et al., 2012).

We consider the following family of non-linear transformations that transform each dimension using a finite set of uniformly bounded, orthonormal basis functions

$\{\eta_1, \ldots, \eta_L\}$:

$$\mathcal{F} := \left\{ f : \mathbb{R} \to \mathbb{R},\ x \mapsto \sum_{l=1}^{L} \beta_l \eta_l(x),\ \text{where } \beta_l \in \mathbb{R}\ \forall\ l \in [L] \right\}. \tag{4.3}$$

Then for $X_A \in \mathbb{R}^{n \times p_A}$, $X_B \in \mathbb{R}^{n \times p_B}$, we consider a variant of CCA where the linear combination is replaced by the additive model to combine all the features. That is, we are solving

$$\max_{\boldsymbol{\psi}, \boldsymbol{\phi}} corr\left(\boldsymbol{\psi}(X_A), \boldsymbol{\phi}(X_B)\right), \tag{4.4}$$

where

$$\boldsymbol{\psi}(X_A) := \left(\sum_{i=1}^{p_A} \psi^i(X_{A;1i}), \ldots, \sum_{i=1}^{p_A} \psi^i(X_{A;ni}) \right)^\top \in \mathbb{R}^{n \times 1},$$

$$\boldsymbol{\phi}(X_B) := \left(\sum_{j=1}^{p_B} \phi^j(X_{B;1j}), \ldots, \sum_{j=1}^{p_B} \phi^j(X_{B;nj}) \right)^\top \in \mathbb{R}^{n \times 1},$$

and ϕ^i and ψ^j are from function class $\mathcal{F}$ for all $i \in [p_A]$ and $j \in [p_B]$, and we take the representation

$$\psi^i(\cdot) := \sum_{l=1}^{L} \beta_{A,l}^i \eta_l(\cdot), \quad \phi^j(\cdot) := \sum_{l=1}^{L} \beta_{B,l}^j \eta_l(\cdot). \tag{4.5}$$

Remark 2. Using this finite set of basis yields a truncation bias if the true function is from a more generally assumed space like a second-order Sobolev space, however, the resulting CCA objective can be much easier to write out and the optimization process can be much faster and more stable. As for CCA, we are handling a nonconvex problem and using alternating updating for optimization, the convergence of the optimization process can be tricky, and therefore the estimation stability in each step is much more needed.

Convert to linear form. We show how we can solve (4.4) efficiently by converting it into a linear format, from where the appropriate way of introducing gene-level sparsity is clear. Define $\widetilde{X}_A \in \mathbb{R}^{n \times p_A L}$, and $\widetilde{X}_B \in \mathbb{R}^{n \times p_B L}$ as

$$\begin{aligned}
\widetilde{X}_A &:= \begin{bmatrix} \eta_1(X_{A;1,1}) & \ldots & \eta_1(X_{A;1,p}) & \ldots & \eta_L(X_{A;1,p}) \\ \eta_1(X_{A;2,1}) & \ldots & \eta_1(X_{A;2,p}) & \ldots & \eta_L(X_{A;2,p}) \\ & & & & \\ \eta_1(X_{A;n,1}) & \ldots & \eta_1(X_{A;n,p}) & \ldots & \eta_L(X_{A;n,p}) \end{bmatrix}, \\
\widetilde{X}_B &:= \begin{bmatrix} \eta_1(X_{B;1,1}) & \ldots & \eta_1(X_{B;1,p}) & \ldots & \eta_L(X_{B;1,p}) \\ \eta_1(X_{B;2,1}) & \ldots & \eta_1(X_{B;2,p}) & \ldots & \eta_L(X_{B;2,p}) \\ & & & & \\ \eta_1(X_{B;n,1}) & \ldots & \eta_1(X_{B;n,p}) & \ldots & \eta_L(X_{B;n,p}), \end{bmatrix}
\end{aligned} \tag{4.6}$$

and $\boldsymbol{w}_A \in \mathbb{R}^{pL\times 1}$, and $\boldsymbol{w}_B \in \mathbb{R}^{qL\times 1}$ as

$$\begin{aligned}\boldsymbol{w}_A &:= (\beta^1_{A,1},\dots\beta^1_{A,L},\dots,\beta^{p_A}_{A,1},\dots,\beta^{p_A}_{A,L})^\top,\\ \boldsymbol{w}_B &:= (\beta^1_{B,1},\dots\beta^1_{B,L},\dots,\beta^{p_B}_{B,1},\dots,\beta^{p_B}_{B,L})^\top.\end{aligned}$$

Also, define $\widetilde{C}_{AB} := \widetilde{X}_A^\top \widetilde{X}_B$, $\widetilde{C}_{AA} := \widetilde{X}_A^\top \widetilde{X}_A$ and $\widetilde{C}_{BB} := \widetilde{X}_B^\top \widetilde{X}_B$. Then we have problem (4.22) be rewritten as a classic CCA problem (constrained form):

$$\max_{\substack{\boldsymbol{w}_A\in\mathbb{R}^{p_A}\\ \boldsymbol{w}_B\in\mathbb{R}^{p_B}}} \boldsymbol{w}_A^\top \widetilde{C}_{AB}\boldsymbol{w}_B, \quad \text{subject to } \boldsymbol{w}_A^\top \widetilde{C}_{AA}\boldsymbol{w}_A = 1,\ \boldsymbol{w}_B^\top \widetilde{C}_{BB}\boldsymbol{w}_B = 1. \tag{4.7}$$

Add feature level sparsity. To introduce feature-level sparsity in the transformed problem (4.7), we partition $\boldsymbol{w}_A$ and $\boldsymbol{w}_B$ into p and q non-overlapping groups respectively:

$$\begin{aligned}\boldsymbol{w}_A^{(t)} &:= (\beta^t_{A,1},\dots,\beta^t_{A,L}) \in \mathbb{R}^{L\times 1}, t = 1,\dots,p\\ \boldsymbol{w}_B^{(s)} &: (\beta^s_{b,1},\dots,\beta^s_{B,L}) \in \mathbb{R}^{L\times 1}, s = 1,\dots,q.\end{aligned}$$

Then we consider a group Lasso (GL) penalty for $\boldsymbol{w}_A$ and $\boldsymbol{w}_B$ as follows:

$$\Omega_{GL}(\boldsymbol{w}_A) := \sum_{t=1}^{p}\sqrt{L}||\boldsymbol{w}_A^{(t)}||_2, \quad \text{and } \Omega_{GL}(\boldsymbol{w}_B) := \sum_{s=1}^{q}\sqrt{L}||\boldsymbol{w}_B^{(s)}||_2.$$

Now we can propose the following group sparse CCA:

$$\begin{aligned}\min_{\boldsymbol{w}_A,\boldsymbol{w}_B} & -\boldsymbol{w}_A^\top \widetilde{C}_{AB}\boldsymbol{w}_B \\ \text{subject to} & ||\widetilde{X}_A\boldsymbol{w}_A||^2 \le 1, \Omega_{GL}(\boldsymbol{w}_A) \le c_1,\\ & ||\widetilde{X}_B\boldsymbol{w}_B||^2 \le 1, \Omega_{GL}(\boldsymbol{w}_B) \le c_2.\end{aligned} \tag{4.8}$$

which will impose feature-level sparsity.

Optimization. The Lagrangian form of the above problem (4.8) is:

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}_A,\boldsymbol{w}_B) = -\boldsymbol{w}_A^\top \widetilde{C}_{AB}\boldsymbol{w}_B + \lambda_1\ \Omega_{GL}(\boldsymbol{w}_A) + \lambda_2\ \Omega_{GL}(\boldsymbol{w}_B)\\ + \eta_1||\widetilde{X}_A\boldsymbol{w}_A||^2 + \eta_2||\widetilde{X}_B\boldsymbol{w}_B||^2,\end{aligned} \tag{4.9}$$

where $\lambda_1 \ge 0, \lambda_2 \ge 0, \eta_1 \ge 0, \eta_2 \ge 0$ are Lagrange multipliers. To minimize $\mathcal{L}(\boldsymbol{w}_A,\boldsymbol{w}_B)$, we use the alternating iterative algorithm based on a block coordinate descent method to optimize $\boldsymbol{w}_A$ for a fixed $\boldsymbol{w}_B$ and vice versa. Particularly, following (Balakrishnan et al., 2012), we use the solution to a non-sparse variant as our initialization (warm start). The explicit processes are summarized as Algorithm 4.27.

The more detailed description is presented in Section 4.6.1, where we also include simulation studies to show the validity of our method.

Algorithm 4.2 Sparse additive CCA.

Input: Data matrixes: $X_A \in \mathbb{R}^{n\times p_A}$, $X_B \in \mathbb{R}^{n\times p_B}$;
a set of 1-D basis functions to be used in additive models: $\{\eta_1, \ldots, \eta_L\}$;
sparsity parameter $s \in (0, 1]$; norm penalty $\eta_1, \eta_2 > 0$;
maximum iteration T; maximum error tolerance $\epsilon > 0$.
Output: Canonical coefficients $\boldsymbol{u}_A \in \mathbb{R}^{p_A\times 1}$ and $\boldsymbol{u}_B \in \mathbb{R}^{p_B\times 1}$.

Data transformation:

$$\text{get } \widetilde{X}_A, \widetilde{X}_B \text{ using Equation 4.6}, \quad \widetilde{C}_{AB} \leftarrow \widetilde{X}_A\widetilde{X}_B^\top \in \mathbb{R}^{p_A L\times p_B L}$$

Warm start:
$$\boldsymbol{w}_A^{(0)}, \boldsymbol{w}_B^{(0)} = \arg\max_{\substack{\boldsymbol{w}_A\in\mathbb{R}^{p_A L\times 1}\\ \boldsymbol{w}_B\in\mathbb{R}^{p_B L\times 1}}} corr(\boldsymbol{w}_A^\top\widetilde{X}_A, \boldsymbol{w}_B^\top\widetilde{X}_B)$$

Optimization: $t \leftarrow 1$;
while $t < T$ *and* $\epsilon_t > \epsilon$ **do**
  Update λ_1, λ_2 as 100(1-s) quantile of grouped norm.
  Fix $\boldsymbol{w}_A$ update $\boldsymbol{w}_B$:

$$\boldsymbol{w}_B^{(t)} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^{p_B L\times 1}} -(\boldsymbol{w}_A^{(t-1)})^\top\widetilde{C}_{AB}\boldsymbol{w} + \eta_2|\widetilde{X}_B\boldsymbol{w}||^2 + \lambda_2\Omega_{GL}(\boldsymbol{w}).$$

  Fix $\boldsymbol{w}_B$ update $\boldsymbol{w}_A$:

$$\boldsymbol{w}_A^{(t)} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^{p_A L\times 1}} -\boldsymbol{w}^\top\widetilde{C}_{AB}\boldsymbol{w}_B^{(t)} + \eta_1|\widetilde{X}_A\boldsymbol{w}||^2 + \lambda_1\Omega_{GL}(\boldsymbol{w}).$$

  Compute error: $\epsilon_t = ||\boldsymbol{w}_A^t - \boldsymbol{w}_A^{t-1}||_2 + ||\boldsymbol{w}_B^t - \boldsymbol{w}_B^{t-1}||_2$
  $t \leftarrow t + 1$
end
Recover gene-level coefficients:

$$\boldsymbol{u}_{A,i} = \frac{1}{L}\sum_{l=1}^{L}|\boldsymbol{w}_{A,(i-1)L+l}^{(t)}| \quad \text{for } i \in [p_A]; \quad \boldsymbol{u}_{B,j} = \frac{1}{L}\sum_{l=1}^{L}|\boldsymbol{w}_{B,(j-1)L+l}^{(t)}| \quad \text{for } j \in [p_B]$$

return $\boldsymbol{u}_A, \boldsymbol{u}_B$.

4.3.3 Joint-HMRF: joint modeling of DE and TADA scores

Inspired by a series of previous works (Liu et al., 2014, 2015; Xie et al., 2022) which adopted a Hidden Markov Random Field (HMRF)(Rabiner, 1989) model to incorporate network into gene risk modeling, we model TADA score and DE score jointly with a four-states HMRF model.

Four-states HMRF model. Consider a list of genes of length p, denote their observed ASD DE z-scores as $Z_1^{DE}, \ldots, Z_p^{DE}$, and the observed ASD TADA z-scores as

$Z_1^{TADA}, \ldots, Z_p^{TADA}$. Also, we denote the true category of genes as hidden states $I_1, \ldots, I_p$, each takes values in 4 cases:

$$I_i = \begin{cases} 1, & \text{if gene } i \text{ is both DE and risk;} \\ 2, & \text{if gene } i \text{ is only DE;} \\ 3, & \text{if gene } i \text{ is only risk;} \\ 4, & \text{if gene } i \text{ is neither DE nor risk} \end{cases} \qquad \text{for all } i \in \{1, \ldots, p\}, \tag{4.10}$$

where value 1 and 2 indicate genes being "active" and "reactive" respectively, and value 3 and 4 are for the "other" genes that we do not care much about.

We then model $\boldsymbol{Z}^{TADA}, \boldsymbol{Z}^{DE}, \boldsymbol{I}$ using the following HMRF model. The probabilistic nature of Z_i^{DE} and Z_i^{TADA} is determined by the unobservable Markov random field on $\{I_i, i \in [p]\}$. That is, given the neighbors $\mathcal{N}_i$ of i, I_i is independent of all other I_j (Markov property). The model is formulated in such a way that conditioning on I_i, Z_i^{DE} and Z_i^{TADA} are independent of any other observable variables, also, we assume that Z_i^{DE} are independent of Z_i^{TADA}.

Therefore, the joint probability of $\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \boldsymbol{I}$ can be written as

$$p(\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \boldsymbol{I}) = p(\boldsymbol{I}) \prod_{i=1}^{p} p(Z_i^{DE}|I_i) p(Z_i^{TADA}|I_i),$$

and parameters can be estimated via iterating maximizing the three parts: $p(\boldsymbol{I})$, $\prod_{i=1}^p p(Z_i^{DE}|I_i)$, $\prod_{i=1}^p p(Z_i^{TADA}|I_i)$.

To model $p(Z_i^{DE}|I_i)$, similar as in (Liu et al., 2014), we consider each Z_i^{DE} follows a Gaussian mixture model:

$$Z_i^{DE} \sim \mathbf{1}_{I_i \in \{1,2\}} \cdot N(\mu_1, \sigma_1^2) + \mathbf{1}_{I_i \in \{3,4\}} \cdot N(0, \sigma_{01}^2). \tag{4.11}$$

And for Z_i^{TADA}, we assume

$$Z_i^{TADA} \sim \mathbf{1}_{I_i \in \{1,3\}} \cdot N(\mu_2, \sigma_2^2) + \mathbf{1}_{I_i \in \{2,4\}} \cdot N(0, \sigma_{02}^2). \tag{4.12}$$

To model $p(\boldsymbol{I})$, we design the potential function in the Markov random field such that $p(\boldsymbol{I})$ has the following representation:

$$p(\boldsymbol{I}) \propto \exp\left\{ \sum_{s=1}^{3} b_{0s} \left(\sum_i \mathbf{1}_{I_i = s} \right) + \sum_{s=1}^{2} b_{1s} \left(\sum_{i;j \in \mathcal{N}_i} \mathbf{1}_{I_i = s, I_j = s} \right) \right\} \tag{4.13}$$

For quick optimization, we instead optimize for the pseudo-likelihood

$$\widetilde{p}(\boldsymbol{I}) := \prod_i p(I_i | \boldsymbol{I}_{\mathcal{N}_i}), \tag{4.14}$$

where $p(I_i|\boldsymbol{I}_{\mathcal{N}_i})$ has

$$p(I_i|\boldsymbol{I}_{\mathcal{N}_i}) \propto \exp\Big(b_{01}\mathbf{1}_{I_i=1} + b_{02}\mathbf{1}_{I_i=2} + b_{03}\mathbf{1}_{I_i=3} + b_{11}\mathbf{1}_{I_i=1}\sum_{j\in\mathcal{N}_i}\mathbf{1}_{I_j=1} + b_{12}\mathbf{1}_{I_i=2}\sum_{j\in\mathcal{N}_i}\mathbf{1}_{I_j=2}\Big) := q(I_i). \tag{4.15}$$

Note that here the network structure only takes effect for active or reactive cases, as those are the only two we care about, and conjectured to be clustered together meaningfully in the network.

Optimization. For optimization, we follow an EM style approach: alternating between maximizing the pseudo-likelihood $\prod_i p(I_i|\boldsymbol{I}_{\boldsymbol{\mathcal{N}_i}})$ using coordinate descent, getting maximum a posteriori probability (MAP) estimation of $\boldsymbol{I}$ using iterative conditional mode (ICM) method (Besag, 1986), and then maximizing $\prod_i p(Z_i^{DE}|I_i)$ and $\prod_i p(Z_i^{TADA}|I_i)$ using the relationship between Gaussian MLE and moment estimation.

As for initialization, instead of random initialization, we initialize the hidden states of the node using

$$I_i^{(0)} = \begin{cases} 1 & \text{if } Z_i^{DE} > C \ \&\ Z_i^{TADA} > C; \\ 2 & \text{if } Z_i^{DE} > C \ \&\ Z_i^{TADA} \le C; \\ 3 & \text{if } Z_i^{DE} \le C \ \&\ Z_i^{TADA} > C; \\ 4 & \text{if } Z_i^{DE} \le C \ \&\ Z_i^{TADA} \le C, \end{cases} \tag{4.16}$$

where the cutoff C is pre-determined threshold; and initialize the values of the parameters using

$$\begin{aligned} &\widehat{b}_{01}^{(0)} = \widehat{b}_{02}^{(0)} = \widehat{b}_{03}^{(0)} = \widehat{b}_{11}^{(0)} = \widehat{b}_{12}^{(0)} = 0; \\ &\widehat{\mu}_1^{(0)} = \frac{\sum_i Z_i^{DE}\mathbf{1}_{I_i^{(0)}\in\{1,2\}}}{\sum_i \mathbf{1}_{I_i^{(0)}\in\{1,2\}}};\ \widehat{\sigma}_1^{(0)} = \frac{\sum_i (Z_i^{DE} - \widehat{\mu}_1)^2\mathbf{1}_{I_i^{(0)}\in\{1,2\}}}{\sum_i \mathbf{1}_{I_i^{(0)}\in\{1,2\}}}; \\ &\widehat{\mu}_2^{(0)} = \frac{\sum_i Z_i^{TADA}\mathbf{1}_{I_i^{(0)}\in\{1,3\}}}{\sum_i \mathbf{1}_{I_i^{(0)}\in\{1,3\}}};\ \widehat{\sigma}_2^{(0)} = \frac{\sum_i (Z_i^{TADA} - \widehat{\mu}_2^{(0)})^2\mathbf{1}_{I_i^{(0)}\in\{1,3\}}}{\sum_i \mathbf{1}_{I_i^{(0)}\in\{1,3\}}}; \\ &\widehat{\sigma}_{01}^{(0)} = \frac{\sum_i (Z_i^{DE})^2\mathbf{1}_{I_i^{(0)}\in\{3,4\}}}{\sum_i \mathbf{1}_{I_i^{(0)}\in\{3,4\}}};\ \widehat{\sigma}_{02}^{(0)} = \frac{\sum_i (Z_i^{TADA})^2\mathbf{1}_{I_i^{(0)}\in\{2,4\}}}{\sum_i \mathbf{1}_{I_i^{(0)}\in\{2,4\}}} \end{aligned} \tag{4.17}$$

Then the whole optimization process is summarized in Algorithm 4.3. After obtaining the estimated parameters, we use Gibbs Sampling to estimate the posterior distribution $p(I_i|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}})$ for each I_i, and take the value of the state which

maximizes such a marginal posterior probability for each gene as its estimated true states.

In Figure 4.2, we show the good convergence of model parameters our proposed computational approximations, when using the PNS network as the underlying gene network Ω, and the DE z-scores from Gandal et al. (2022), TADA z-scores from Fu et al. (2022), and set the threshold C as $\Phi^{-1}(1-0.01)$, with Φ^{-1} as the inverse CDF function of Gaussian $(N(0,1))$ distribution. Simulation studies in Section 4.6.2 also demonstrate the effectiveness of our proposed approximate learning algorithm.

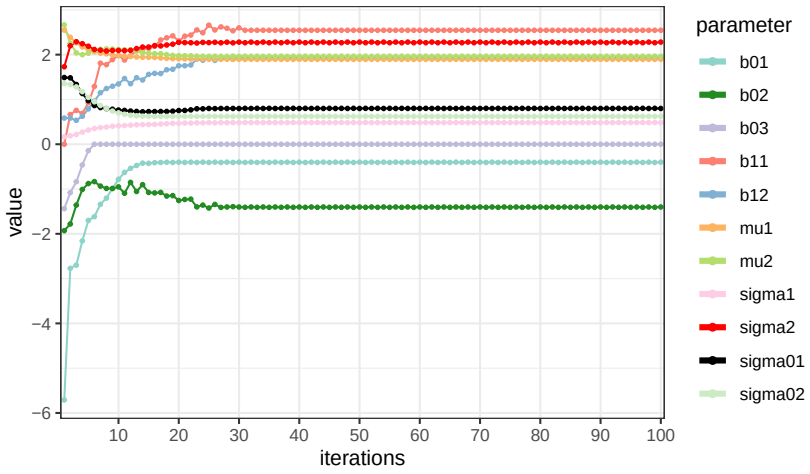


Figure 4.2: Evidence of convergence when applying Joint-HMRF algorithm on real data. For input, we use the PNS network as the underlying gene network Ω, and the DE z-scores from Gandal et al. (2022), TADA z-scores from Fu et al. (2022).

Algorithm 4.3 Joint-HMRF inference.

Input: Gene network: $\Omega \in \mathbb{R}^{p\times p}$; gene DE z-scores: $Z^{DE} \in \mathbb{R}^{1\times p}$; gene TADA z-scores: $Z^{TADA} \in \mathbb{R}^{1\times p}$; threshold C; max iteration T; max error tolerance $\epsilon > 0$.

Output: Estimated HMRF parameters:

$$\widehat{\boldsymbol{\theta}} := (\widehat{b}_{01}, \widehat{b}_{02}, \widehat{b}_{03}, \widehat{b}_{11}, \widehat{b}_{12}, \widehat{\mu}_1, \widehat{\mu}_2, \widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_{01}, \widehat{\sigma}_{02})$$

Initialization: Set the initial states of the node and the initial values for parameters using (4.16) and (4.17).

Optimization: $t \leftarrow 1$;

while $t < T$ *and* $\epsilon_t > \epsilon$ **do**

Update hidden variable related parameters:

$$(b_{01}^{(t)}, b_{02}^{(t)}, b_{03}^{(t)}, b_{11}^{(t)}, b_{12}^{(t)}) = \arg\max \prod_i \frac{q(I_i = I_i^{(t-1)})}{\sum_{s\in[4]} q(I_i = s)}$$

where q is the energy function in (4.15).

Apply a single cycle of ICM (Besag, 1986) to update the hidden states:

$$\begin{aligned} I_i^{(t)} &= \arg\max_{s\in[4]} p(I_i = s|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \boldsymbol{I}_{\mathcal{N}_i}^{(t-1)}, \widehat{\boldsymbol{\theta}}^{(t-1)}) \\ &\propto p(Z_i^{DE}|I_i) p(Z_i^{TADA}|I_i) p(I_i|\boldsymbol{I}_{\mathcal{N}_i}^{(t-1)}, \widehat{\boldsymbol{\theta}}^{(t-1)}) \end{aligned} \tag{4.18}$$

Update observable variables related parameters:

$$\begin{aligned}
\widehat{\mu}_1^{(t)} &= \frac{\sum_i Z_i^{DE} p(I_i^{(t)} \in \{1,2\}|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \widehat{\boldsymbol{\theta}}^{(t-1)})}{\sum_i p(I_i^{(t)} \in \{1,2\}|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \widehat{\boldsymbol{\theta}}^{(t-1)})}; \\
\widehat{\sigma}_1^{(t)} &= \frac{\sum_i (Z_i^{DE} - \widehat{\mu}_1^{(t)})^2 p(I_i^{(t)} \in \{1,2\}|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \widehat{\boldsymbol{\theta}}^{(t-1)})}{\sum_i p(I_i^{(t)} \in \{1,2\}|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \widehat{\boldsymbol{\theta}}^{(t-1)})}; \\
\widehat{\mu}_2^{(t)} &= \frac{\sum_i Z_i^{TADA} p(I_i^{(t)} \in \{1,3\}|\boldsymbol{Z}^{DE}, \boldsymbol{Z^{TADA}}, \widehat{\boldsymbol{\theta}}^{(t-1)})}{\sum_i p(I_i^{(t)} \in \{1,3\}|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \widehat{\boldsymbol{\theta}}^{(t-1)})}; \\
\widehat{\sigma}_2^{(t)} &= \frac{\sum_i (Z_i^{TADA} - \widehat{\mu}_2^{(t)})^2 p(I_i^{(t)} \in \{1,3\}|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \widehat{\boldsymbol{\theta}}^{(t-1)})}{\sum_i p(I_i^{(t)} \in \{1,3\}|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \widehat{\boldsymbol{\theta}}^{(t-1)})}; \\
\widehat{\sigma}_{01}^{(t)} &= \frac{\sum_i (Z_i^{DE})^2 p(I_i^{(t)} \in \{3,4\}|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \widehat{\boldsymbol{\theta}}^{(t-1)})}{\sum_i p(I_i^{(t)} \in \{3,4\}|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \widehat{\boldsymbol{\theta}}^{(t-1)})}; \\
\widehat{\sigma}_{02}^{(t)} &= \frac{\sum_i (Z_i^{TADA})^2 p(I_i^{(t)} \in \{2,4\}|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \widehat{\boldsymbol{\theta}}^{(t-1)})}{\sum_i p(I_i^{(t)} \in \{2,4\}|\boldsymbol{Z^{DE}}, \boldsymbol{Z^{TADA}}, \widehat{\boldsymbol{\theta}}^{(t-1)})}.
\end{aligned}$$

Compute error: $\epsilon_t = ||\widehat{\boldsymbol{\theta}}^{(t)} - \widehat{\boldsymbol{\theta}}^{(t-1)}||_\infty$

$t \leftarrow t + 1$

end

return $\widehat{\boldsymbol{\theta}}^{(t)}$.

4.3.4 TwoLeiden: two-step graph clustering

After the gene network construction and joint regularization of DE and TADA information, we now have the following network with attributes (or say covariates): for each node (gene) in the network, it is associated with a discrete value that indicates its estimated true states. Our goal is to identify clusters in the gene network that are enriched with genes in states 1 (active) or states 2 (reactive). This task falls into a more general topic: community detection with node attributes (Chunaev, 2020). Existing methods fall into three main categories: *pre-fusion*, *simultaneous-fusion*, and *post-fusion*, depending on whether to incorporate the attribute information *before*, *during*, or *after* the network community detection process. However, when applied in our specific setting, we observed that these methods failed to yield more meaningful results compared to clustering without covariates. This suggests a delicate situation where the integration of network structure and covariate information can lead to a performance collapse. While we acknowledge that we may not have explored all potential solutions, we leave this avenue for future research to address. In this study, we have employed a straightforward approach by conducting clustering solely based on the network information. Subsequently, we determine the categorization of clusters as "active" or "reactive" based on the proportions of active and reactive genes within each cluster. Despite its simplicity, this approach has yielded the most biologically meaningful results in our analysis.

Network community detection (or say graph clustering) itself has also been a long-studied problem (Mohamed et al., 2019; Su et al., 2022), and popular method in the genetic domain falls into three main categories: model-based, embedding-based, and modularity-based. Model-based methods are mostly based on the stochastic block model and probabilistic graphical model. Embedding methods focus on learning a node embedding for the graph and then conducting matrix clustering based on those embeddings. Modularity-based methods aim to find community partition that maximizes a measure called Modularity, which measures the density of links inside communities compared to links between communities. In this project, we follow the modularity-based route as it is the most widely used method in genetic research communities.

The most popular modularity-based method is the Louvain method (Blondel et al., 2008). However, it is known to fail to detect clusters smaller than some scale (Fortunato and Barthelemy, 2007), and can also yield arbitrarily badly connected communities (Traag et al., 2019), therefore we adopt a refined method called Leiden (Traag et al., 2019) which successfully addresses these two drawbacks. In Figure 4.3, we show evidence of the superiority of Leiden over Louvain: even with the highest resolution 1, Louvain fails to produce more small clusters; while for Leiden, more small-sized clusters can be captured as resolution goes up, and for the highest resolution 1, each gene forms its own clusters. In practice, we select the resolution

that gives us the highest modularity. Note that Leiden clustering cannot output clusters with a specified fixed number of outputs, instead, it determines the number of clusters that achieves the highest modularity.

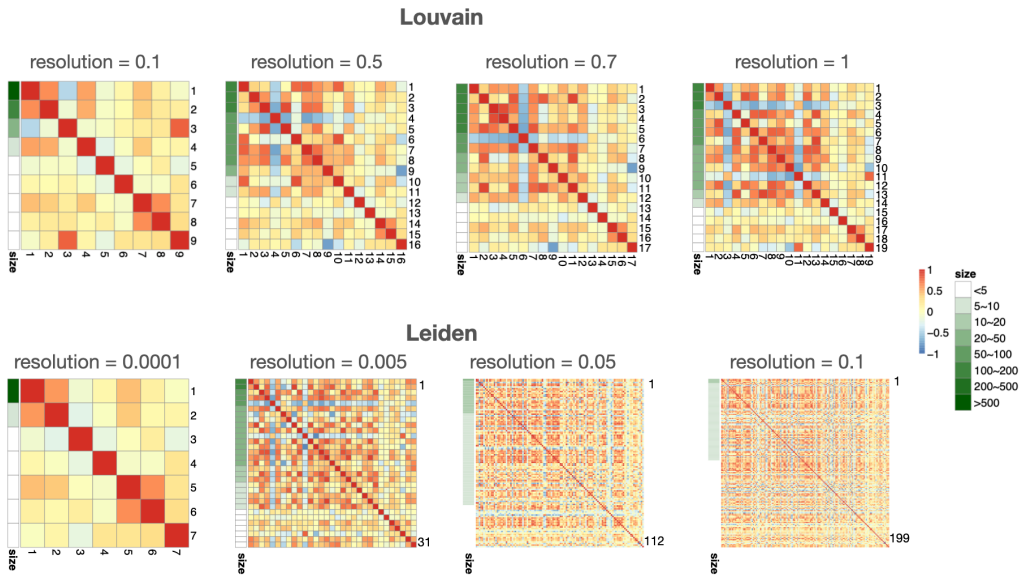


Figure 4.3: The correlation matrix between the first eigenvector of each cluster. We show results using the real data described in Section 4.4.1 with the PNS network.

A two-step variant of Leiden. We observe that clusters output by Leiden with a resolution that gives the highest modularity tend to still have many small-sized clusters that are too similar to each other (Figure 4.3). This inspires us to take the following two-step approach. We propose to first use Leiden to construct initial clusters; then we construct a similarity matrix among those clusters using information from each cluster's low-dimensional embeddings. Finally, we conduct a hierarchical clustering based on this similarity matrix, which merges the initial clusters into our final clusters. We select the number of final clusters in the merging step that leads to the highest stability like MRtree did. The detailed steps are presented in Algorithm 4.4

Algorithm 4.4 Two Step Leiden.

Input: Gene network $\Omega \in \mathbb{R}^{p\times p}$; gene expression matrix $X \in \mathbb{R}^{n\times p}$.

Output: A series of gene sets which represents a partition of all the genes.

31 **Step 1: Get the series of initial gene clusters of high resolution.**
Conduct Leiden clustering on network Ω with a resolution that gives the highest modularity. Denote the resulted clusters $C_1, \ldots, C_k$ as initial clusters.

32 **Step 2: Get the similarity measure according to gene expression.**
Compute the first eigenvector for each initial gene cluster using the expression matrix X, then compute the correlation matrix among those eigenvectors, and denote it as $\boldsymbol{S} \in \mathbb{R}^{k\times k}$.

33 **Step 4: Merge initial clusters.**
Convert the similarity matrix $\boldsymbol{S}$ to the distance matrix and conduct MRtree Peng et al. (2021) on it to obtain the final clusters.

4.4 Real data analysis

4.4.1 Datasets

The Gandal brain dataset. Gandal et al. (2022) perform bulk RNA-sequencing (RNA-seq) on 725 brain samples spanning 11 distinct cortical areas in 112 ASD cases and neurotypical controls. The authors have conducted several processing and analysis steps shown below, which prepare us for our analysis.

1. `Gene Filtering:` Genes were retained if they had a counts-per-million (CPM) value greater than 0.1 in at least 30% of the samples. Additionally, genes with an effective length (measured by RSEM) of less than 15 bp were removed. Following these filters, the dataset consisted of 24836 genes.

2. `Normalization:` To ensure comparability and eliminate potential biases, the remaining genes were subjected to further normalization using the limma-trend approach within the `limma` R package. This approach involved taking the log2(CPM+1) transformation of read counts, while accounting for variations in sample read depth. Additionally, a CQN-derived offset value was incorporated during the normalization process to address potential biases related to GC content and gene effective length. Collectively, these steps aimed to obtain normalized expression data suitable for downstream analysis.

3. `Outlier removal:` To identify sample outliers within each sequencing batch by cortical lobe group (frontal, parietal, temporal, and occipital), the normalized expression data underwent a two-step outlier detection process. First, samples were flagged as outliers if they met the following criteria: (1) an absolute z-score exceeding 3 for any of the top 10 expression principal components (PCs), and (2) a sample connectivity score below -2. The sample connectivity

score was computed using the fundamental NetworkConcepts function from the WGCNA R package, utilizing the signed adjacency matrix (soft power of 2) of the sample biweight midcorrelation. This procedure successfully identified 34 outliers in the dataset.

4. `Technical effects removal:` Next, to address technical effects, a regressed dataset was created using the lmerTest package in R. This involved subtracting the effects of 20 technical covariates from each gene, resulting in a dataset that retained only the random intercept, biological covariate effects, and the residual. The regressed gene expression dataset specifically captured the effects of biological covariates, including subject, diagnosis, region, sex, ancestry, and age.

5. `DE ASD genes identification:` Finally, the whole cortex differentially expressed (DE) genes were identified by examining the significant nonzero coefficients associated with the covariate "diagnosis" in the regression model.

The above processing output a final gene expression matrix with a total of 24836 genes and 725 samples (341=Control, 384=ASD) with technical noises removed. We will use this dataset for our analysis.

The Fu TADA dataset. We opted to utilize the TADA association results provided by Fu et al. (2022) as they represent the most cutting-edge findings in the field. The gene association results in their study were obtained through a joint statistical analysis of rare protein-truncating variants (PTVs), damaging missense variants, and copy number variants (CNVs) identified from exome sequencing data collected from a large cohort of 63237 individuals from ASD cohorts. These results serve as a robust and comprehensive resource for investigating gene associations in the context of our analysis.

4.4.2 Data preparation and preprocess

Our input data have three parts:

- Whole cortex gene expression data (bulk RNA-sequencing data) from (Gandal et al., 2022), which has 24836 genes and 725 samples(341=Control, 384=ASD).
- ASD DE q-values from (Gandal et al., 2022) for all the 24836 genes using the whole cortex data gene expression data.
- ASD TADA q-values from (Fu et al., 2022) for 18128 genes.

With these three parts of data, we did the following exploratory analysis and further processing to make it fit our task.

Recalibration. We first transform the TADA/DE q-values into p-values using the following formula for each gene $i \in [p]$:

$$p\text{-value}_i = \frac{q_i \times \text{rank}(q_i)}{p \times \max_{i \in [p]} q_i} \tag{4.19}$$

and check whether the p-values need re-calibration. Specifically, we consider the p-values to be well-calibrated if the distribution of the corresponding z-scores using

$$z\text{-score} = \Phi^{-1}(1 - p\text{-value}) \tag{4.20}$$

as if the p-value is a one-sided and is a mixture of $N(0,1)$ and $N(\delta, \tau)$, otherwise not. Below we can see that (Figure 4.4) TADA z-scores distributed nicely as a mixture of $N(0,1)$ and $N(\delta, \tau)$ (i.e. well-calibrated); while DE z-scores appear to have a shifted null distribution (Figure 4.4). Therefore we recalibrate the DE z-scores using Efron's method (Efron, 2004): estimating the mean μ and variance σ of the distribution of corresponding z-values and then adjusted it as $\widetilde{z} = \frac{z-\mu}{\sigma}$. We use these adjusted scores as our input z-scores to the joint-HMRF model.

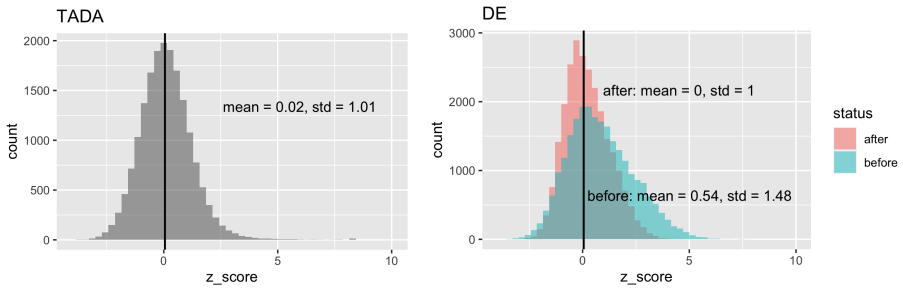


Figure 4.4: **(left)**The distribution of TADA z-score for all the genes. **(right)** The distribution of DE z-score for all the genes before and after calibration.

Filtration. We filter the genes to adapt to our analysis and focus on genes that have adequate expression, according to the prior knowledge that ASD risk genes are rarely very low-expressed. We first take the intersection of genes that are covered in both TADA results and DE results, which results in 15628 genes. Then we choose genes that exhibit non-zero expression in at least 50% of the samples. Among these genes, we further prioritize the top 8000 genes based on their mean expression levels. This step allows us to focus on a subset of genes that demonstrate sufficiently robust expression patterns across the samples.

4.4.3 Network estimation

We use all the samples from the neurotypical brain to estimate the gene network. To ensure the exclusion of gene connections influenced by ASD, we deliberately omit the

ASD brain samples during the network estimation process. This decision is based on the underlying assumption that the mutation causing ASD can lead to differential expression of reactive genes. Consequently, these reactive genes are likely to exhibit high correlation with causal (active) ASD genes, resulting in dense and inseparable connections between the active and reactive communities. By excluding ASD brain samples from network estimation, we aim to mitigate the confounding effects and focus on capturing biologically meaningful connections within the network.

We consider the following six types of network estimation methods:

	Marginal	**Partial**	**Aggregated Partial**
Linear	Pearson	PNS (Liu et al., 2014)	EPC (Wang et al., 2015)
Nonlinear	aLDG (Chapter 3)	GENIE3 (Huynh-Thu et al., 2010)	EnPAC (Section 4.3.2)

Table 4.3: The summary of gene network estimation methods of our consideration.

Following DAWN (Liu et al., 2014), we preselect a subset of core genes of potential interest and focus on estimating their connection with genes both within this subset and outside this subset. In contrast to DAWN, our approach involves including genes that belong to the union of DE genes and TADA genes. Specifically, we consider genes that fall within the highest 10% of either the TADA z-scores or the DE z-scores as "core genes". By combining these two sets, we obtain a total of 1532 core genes for further analysis. This integration allows us to capture genes that show strong evidence of association with the phenotype of interest, considering both differential expression and TADA analysis.

For EnPAC, EPC and PNS method that involves sparsity parameters, we refer to both the R^2 of the network fitting to the power law [3], and the visualization to choose the appropriate value. For example, for PNS, we end up choosing the lasso penalty parameter $\lambda = 0.2$, which has both a satisfactory R^2 (Figure 4.5) and a visually rich structure (Figure 4.6).

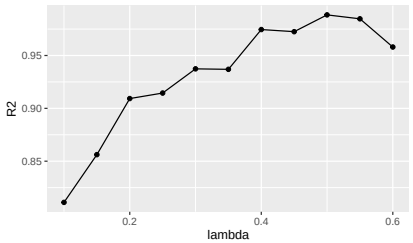


Figure 4.5: The R^2 of fitting power-law for networks estimated with different λ: $\lambda \in \{0.1, 0.15, 0.2, \ldots, 0.5\}$.

[3] In the network context, a power law refers to a specific pattern of connectivity or degree distribution within a network. Specifically, it implies the frequency of nodes with high degrees decreases exponentially as the degree increases.

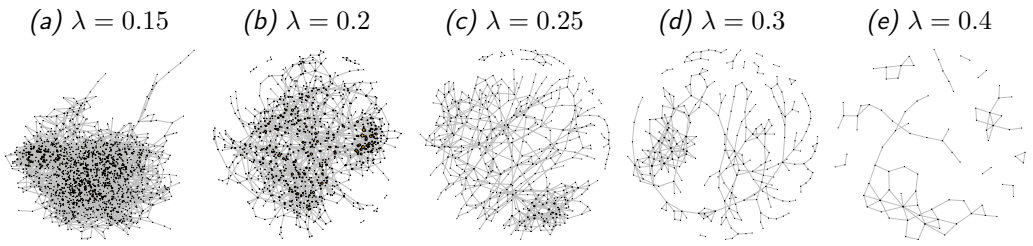


Figure 4.6: The visualization of the fitted networks with different λ: $\lambda \in \{0.1, 0.2, 0.3, \ldots, 0.5\}$.

For Pearson, aLDG, and GENIE3 which do not consider sparsity, we simply use a hard thresholding approach to sparsify and binarize the network. We set the threshold on edge weights such that only edges with the top 0.1% of weights are preserved. This cutoff gives us roughly the same sized network with PNS, so we can compare more fairly. A special note on aLDG is that most edges with high weights are still capturing linear relationships, therefore we take a transformation[4] such that the nonlinear ones are more upweighted.

After obtaining the estimated network following the above processes, we further prune the network by removing the genes with degrees smaller than 2 for better visualization and analysis. Figure 4.7 visualizes all the six estimated networks. We can see that some methods clearly produce networks with more structure than others: particularly, Pearson, PNS, GENIE3 and EnPAC are visually the most promising ones. EPC seems to collapse, probably due to the unsatisfactory sparse CCA optimization implemented in the paper. We leave the diagnoses for this method for future work.

All the networks under investigation have approximately 1000 nodes and exhibit varying degrees of overlap with each other. In Figure 4.8, the proportion of overlapping nodes and edges is depicted for each pair of networks. Notably, the Pearson, PNS, and EnPAC networks demonstrate the highest node overlap, suggesting the presence of a group of nodes that contribute significantly to the overall network structure. Furthermore, both Pearson and PNS networks exhibit substantial edge overlap, indicating similar gene relationships captured by these networks. However, EnPAC shows minimal edge overlap with Pearson and PNS, suggesting that it estimates distinct gene relationships compared to the other two networks. To evaluate the practical implications of these networks, we assess their usefulness in downstream applications such as joint-HMRF modeling and active/reactive cluster identification. Given the absence of ground truth in this domain, we omit simulation

[4]For a pair of genes, denote their original aLDG measure as a and their Pearson correlation as b, then we transform the aLDG measure as $\widetilde{a} = a(1 - |b|)^{1.2}$ such that linear relationship is being downscaled and nonlinear relationship are being upscaled.

studies and instead rely on the biological relevance and interpretability of the results obtained from these networks.

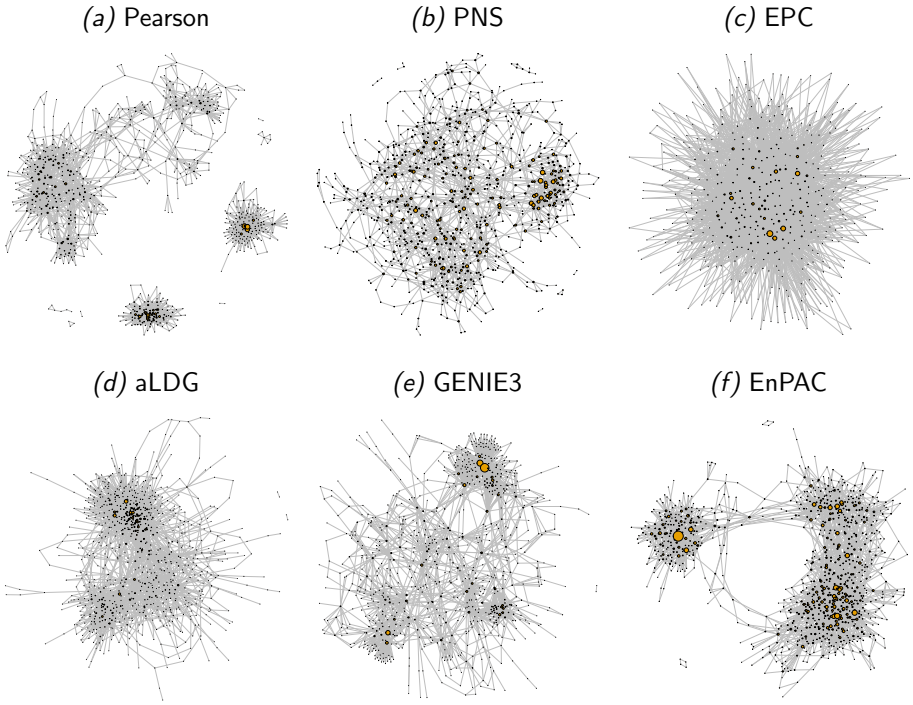


Figure 4.7: The visualization of the fitted networks with different methods. The hyperparameters in each method are chosen to have the best fit of the power law with visually clear interpretation.

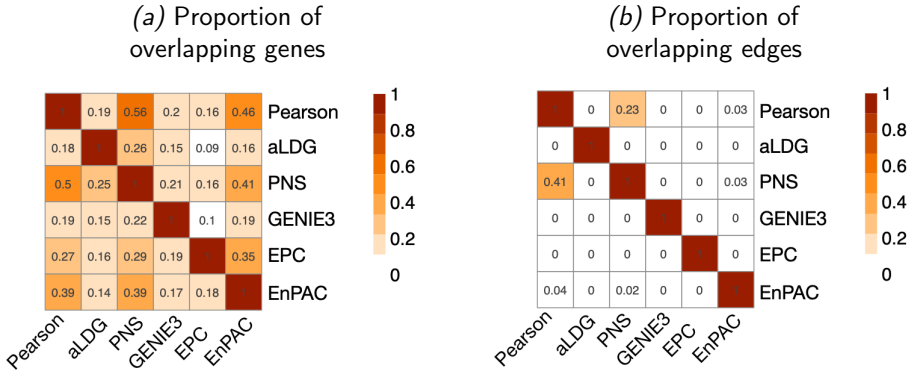


Figure 4.8: The proportion of overlapping genes and edges for each pair of estimated gene networks.

4.4.4 Network regularization of DE and TADA

Following the methodology outlined in Section 4.3.3, we proceed to regularize the differential expression (DE) and TADA information using the various estimated networks described in Section 4.4.3. To highlight the significance of our joint modeling approach, we compare it to a naive approach that treats DE and TADA signals separately, serving as a baseline for comparison. In this naive approach, DE and TADA information are independently modeled without considering their interplay. By isolating the DE and TADA information, we aim to assess the added value and necessity of our joint modeling framework.

A baseline model: separate HMRF model. To establish a baseline method, we employ the DAWN framework proposed by Liu et al. (2015) for modeling risk genes using an HMRF-based approach with binary hidden states (risk and non-risk). In this baseline approach, we run the DAWN algorithm twice: The first run of DAWN utilizes TADA z-scores as input, focusing on the TADA analysis. We employ the same estimated network for this run. The second run of DAWN employs DE z-scores, concentrating on the DE analysis. Again, we utilize the same estimated network as in the previous run. After obtaining the estimated parameters from each run, we employ Gibbs sampling to estimate the marginal posterior probability, denoted as $P(I_i = 0|\boldsymbol{Z})$. This probability reflects the likelihood of a gene being non-risk given the observed gene expression patterns. Finally, we update the p-values based on these posterior probabilities, following the approach described by Liu et al. (2014), for both the TADA analysis and the DE analysis. This involves establishing a connection between $P(I_i = 0|\boldsymbol{Z})$ and q-values, resulting in updated p-values.

After obtaining the DAWN TADA p-values and DAWN DE p-values for each node in the network, we proceed to classify the genes based on these values. Genes that have both DAWN TADA p-values and DAWN DE p-values below 0.01 are classified as "active". This implies that these genes exhibit significant associations with both the TADA risk analysis and the differential expression analysis. On the other hand, genes that have both DAWN TADA p-values and DAWN DE p-values above 0.01 are classified as "reactive". These genes demonstrate a lack of significant associations in both the TADA risk analysis and the differential expression analysis. Genes that do not fall into either of these categories are classified as "others".

By performing these steps, we establish a baseline method for analyzing TADA and DE information separately, while still incorporating the estimated network like our joint modeling framework.

our method: Joint-HMRF model. When applying our proposed method in Section 4.3.3, we set the threshold C on z-score in initialization as $\Phi^{-1}(1 - 0.01)$, which corresponds to p-value 0.01. We run the Algorithm 4.3 for each of the six gene networks in Section 4.4.3 till convergence, and they all converge within 100 iterations (except for EPC, which fails to converge). Then we classify genes based

on their most possible hidden states (MPHS) according to the marginal posterior $p(I_i|\boldsymbol{Z}^{DE}, \boldsymbol{Z}^{TADA})$: we call genes with MPHS=1 as "active", and MPHS=2 as "reactive", and the rest as "others".

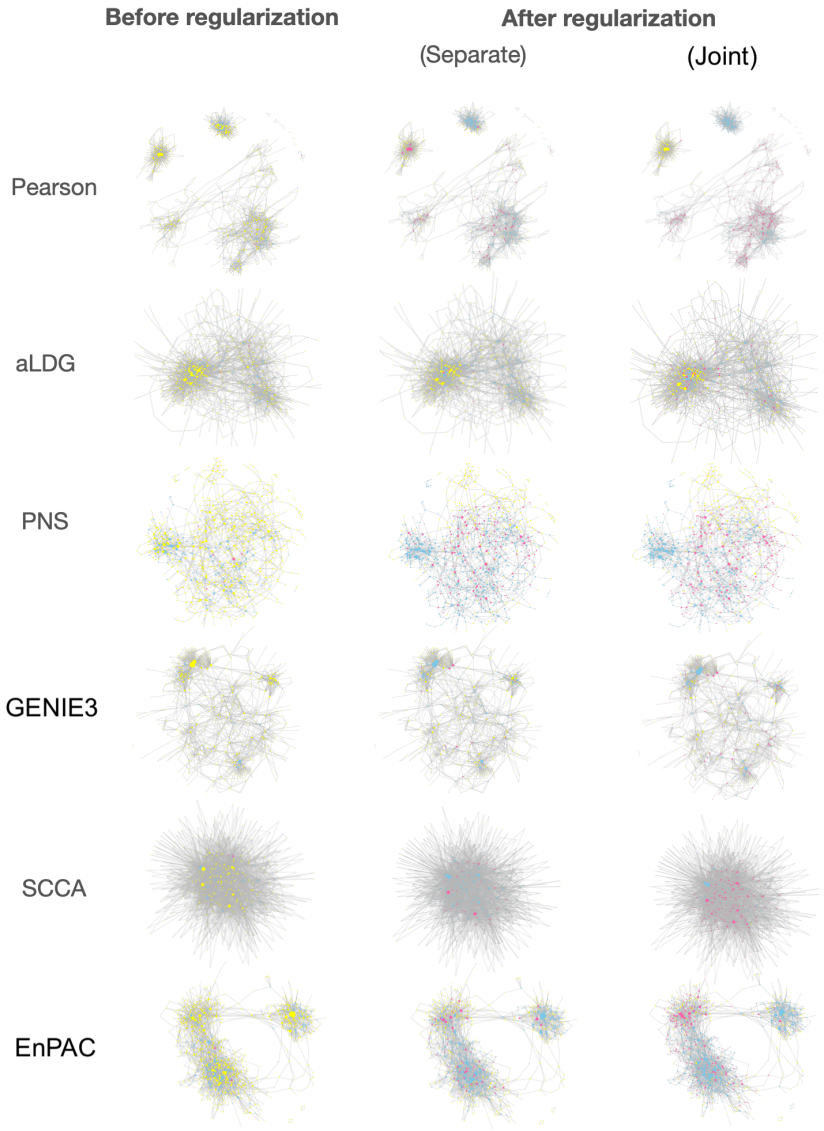


Figure 4.9: (Continue on next page)

Figure 4.9: The visualization of the fitted networks with different methods, and annotated with different states definition/estimation. **(left)** Node are colored by the original TADA and DE results. **(middle)** Node are colored by seperate-HMRF TADA and DE results (baseline). In both graphs, node are colored by a preliminary definition of active (hotpink), reactive (skyblue), and other (yellow) genes: where active-DE is genes with $p_{DE} < 0.01,\ p_{TADA} < 0.01$; reactive-DE is genes with $p_{DE} < 0.01,\ p_{TADA} \geq 0.01$; and others are the rest genes. **(right)** Node are colored by joint-HMRF TADA and DE results, according to the estimated genes states: active (hotpink), reactive (skyblue), and other (yellow).

Figure 4.9 demonstrates each gene's states before and after the network regulation. We can see that, before network regularization, DE and TADA significant genes have nearly no overlap, while after the network regularization, more overlap (i.e. "active" genes) is shown. Also, our joint method is better than the separate one as it visually gives more purified clusters: for the separate method, "active" genes tend to scatter over the network rather than clustered together.

4.4.5 Active and reactive DE gene modules identification:

Given the results from Section 4.4.4, we now have an associated state for each gene in the network. We then group genes into different clusters using the method described in Section 4.3.4.

As an example of the process, we demonstrate how we conduct the grouping for the PNS network. Specifically, we choose the resolution in the initial Leiden clustering that gives us the highest modularity. According to Figure 4.11, we choose the resolution as 0.005. With resolution 0.005, Leiden originally outputs 31 clusters. Then we merge these initial clusters for better interpretability using hierarchical clustering on the fused similarity described in Algorithm 4.4, where the final number of clustering 14 is chosen based on the stability of merged clusters. We can see that, after the merging process, the correlation among the cluster embeddings is much lower, indicating the clusters are more heterogenous to each other.

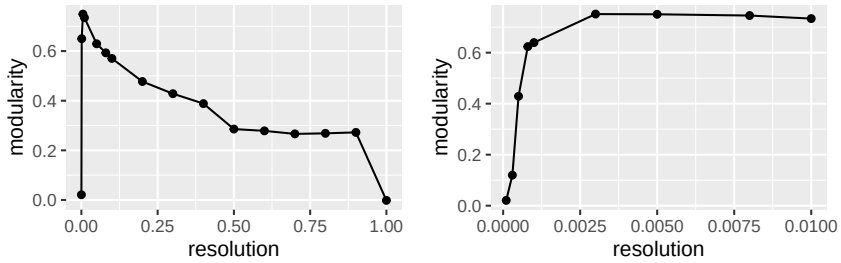


Figure 4.10: The modularity score using different resolution parameters in Leiden clustering.

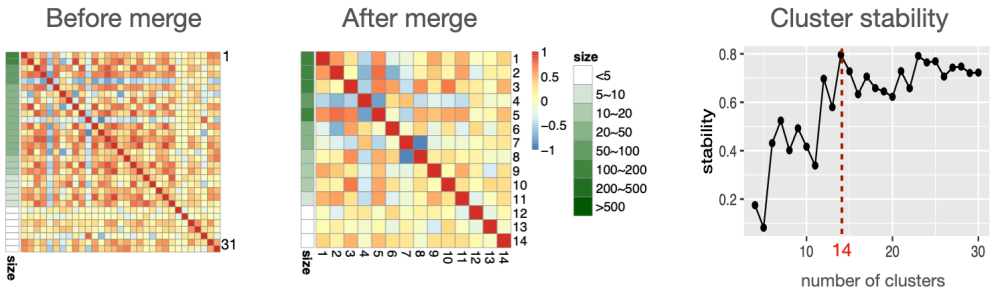


Figure 4.11: **(Left)** The correlation matrix of cluster eigengenes for all the clusters before merging. **(Middle)** The correlation matrix of cluster eigengenes for all the clusters after merging, given the chosen cluster number 14. **(Right)** The stability of cluster results given different cluster numbers. We choose 14 as our determined cluster numbers following MRtree (Peng et al., 2021).

Figure 4.12 visualize the results for the found 14 clusters for PNS network, and Table 4.4 summarizes cluster-specific statistics for all 14 clusters. We exclude the clusters with too small sizes (sizes smaller than 5 genes), which gives us only 11 clusters (C1-11) as a result. We can see a bunch of clusters that are potentially active (e.g. C2,C3,C4,C5,C8) and a bunch of clusters that are potentially reactive (e.g. cluster C1,C6,C9,C11). We further plot the proportion of active genes and reactive genes in each of these 9 clusters in Figure 4.13. We can see more clearly that these 9 clusters fall into three categories: "active"(C2, C3, C4, C5, C8): which have high proportions of active genes; "reactive"(C1, C6, C9, C11): have high proportions of reactive genes but low proportions of active genes; and "other" (C7, C10): have both low proportions of reactive gene ratio but low proportions of active genes. This distinction in clusters would not be observable without our network regularization process (i.e. our joint-HMRF method): in Figure 4.13, we also show the corresponding signal distribution for each cluster with just the raw DE and TADA information, where genes with both DE p-value$<$0.01 and TADA p-value$<$0.01 are regarded "active", and genes with DE p-value$<$0.01 and TADA p-value$\geq$ 0.01 are regarded "reactive". One can see that it's impossible to categorize these clusters into "active" or "reactive" without network regularization. In Table 4.4, we show the summary statistics for all 14 clusters, which makes the effect of our network regularization process (i.e. our joint-HMRF method) even more clear.

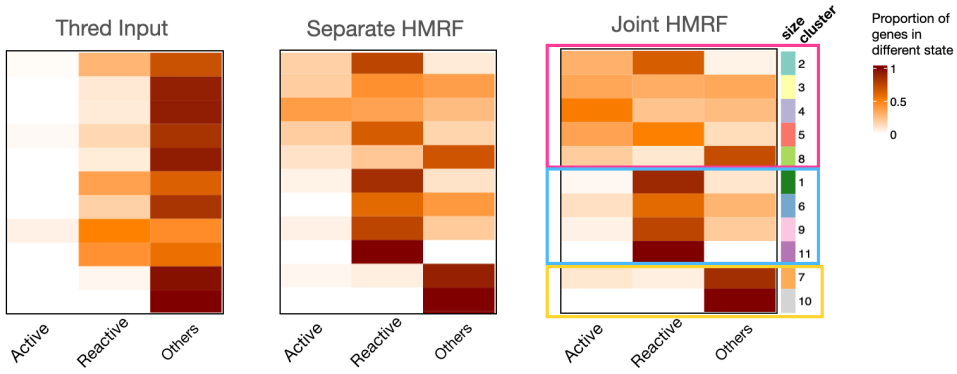


Figure 4.13: The proportion of active and reactive genes in each of the big 11 gene clusters before and after network regularization. We can see there is a clear distinction in the signal distribution after network regularization. The clusters in the blue block are just the "reactive" clusters we are looking for, and the clusters in the pink block are just the "active" clusters we are looking for.

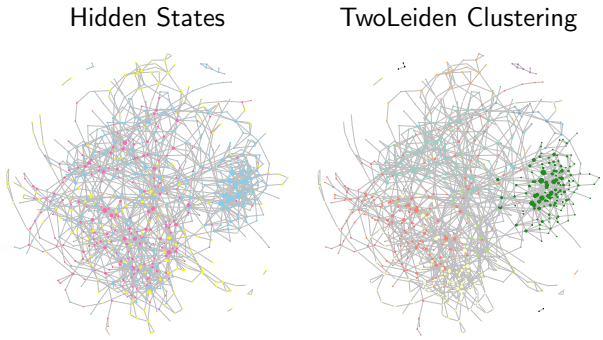


Figure 4.12: Results about the potential/identified reactive and active DE gene clusters. **(left)** For Hidden States, we mean nodes are colored by the estimated hidden states from our joint-HMRF model. The skyblue clusters are the potential "reactive" DE genes we are interested in; while the hotpink clusters are the potential "active" DE genes we are interested in, and the "yellow" genes are the other genes we don't care about. **(right)** We show the graph visualization with nodes colored by our two-step Leiden clustering.

		before regularization		after regularization		# total
		# active	# reactive	# active	# reactive	
active clusters	C2	2	47	50	107	165
	C3	0	11	49	42	133
	C4	0	5	35	15	67
	C5	4	30	73	102	202
	C8	0	4	11	5	57
reactive clusters	C1	0	46	3	111	126
	C6	0	6	4	21	35
	C9	1	10	1	15	20
	C11	0	3	0	7	7
others	C7	0	1	3	2	34
	C10	0	0	0	0	7
	C12	0	0	0	1	3
	C13	0	0	0	0	3
	C14	0	1	0	3	5

Table 4.4: The summary of statistics for all the PNS clusters.

We conduct a similar analysis using each of the six networks estimated in Section 4.4.3, however, only PNS gives the cleanest results and are most biologically interesting. This observation is reasonable as the PNS method shares similar edge interpretations with Markov Random Field (MRF). Our initial hope that the other network concept, though disagreeing with the MRF assumption, might still present useful results seems to fail. Nevertheless, these negative results provide useful insights/guidelines for future research.

In the following sections, we only show the interpretation of results for the PNS network, and leave those for the others in Section 4.6.3.

4.4.6 Interpretation of results

GO Enrichment analysis. The Gene Ontology (Ashburner et al., 2000) is a widely used standardized vocabulary that describes the functions, cellular locations, and biological processes associated with genes and gene products. It provides a structured framework to annotate genes based on their functional attributes. GO terms are organized in a hierarchical manner, with broad terms at the top (e.g., "biological process") and more specific terms below (e.g., "cell cycle" or "DNA repair"). Gene Ontology (GO) term enrichment is a computational method used in bioinformatics and genomics to analyze large sets of genes or proteins and determine whether specific GO terms are overrepresented within those sets.

To perform GO term enrichment analysis, researchers start with a set of genes or proteins of interest, typically derived from experimental data. The goal is to identify whether any particular GO terms are significantly enriched within this gene set compared to what would be expected by chance. The analysis involves statistical calculations to assess the significance of observed GO term enrichments. Various

statistical methods, such as hypergeometric or Fisher's exact test, are commonly used for this purpose. These methods take into account the total number of genes in the genome, the number of genes associated with each GO term, and the size of the gene set under investigation. The output of a GO term enrichment analysis typically includes a list of significantly enriched GO terms along with statistical measures, such as p-values, which help determine the level of confidence in the enrichment results.

We conduct GO enrichment analysis using the `enrichGO` function in "clusterProfiler" (Yu et al., 2012) R package. We set the p-value cutoff as 0.05 and used Benjamini-Hochberg (Benjamini and Hochberg, 1995) for multiplicity correction. We find out that, all of the five active DE gene clusters we identified (C2, C3, C4, C5, C8) are enriched with synaptic/neural related GO terms (Figure 4.14 first row), while all of the reactive DE gene clusters we identified (C1, C6, C9, C11) are enriched with responsive GO terms (Figure 4.15 first row). We also conduct the Synaptic GO enrichment analysis, and find that all of our identified active DE gene clusters are enriched with synaptic genes (Figure 4.14 second row); while all of our identified reactive DE gene clusters are **not** much enriched with synaptic genes.

We also show GO results for the "other" clusters (C7 and C10) we identified in Figure 4.16, and as we expected they are not enriched in synaptic/neural related functions, and they are not enriched in synaptic genes.

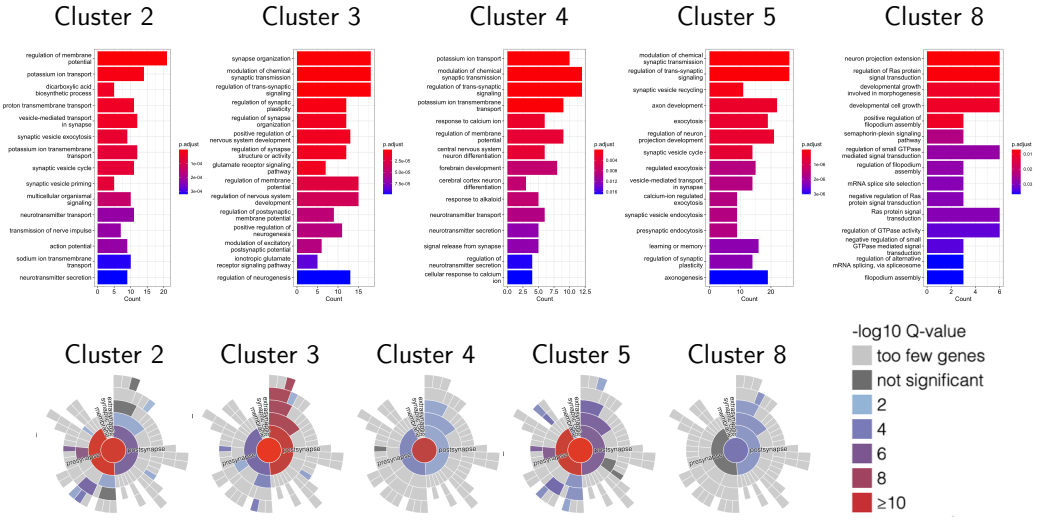


Figure 4.14: The ALL terms GO analysis (first row) and SynGO analysis (second row) results for our identified **active** DE clusters.

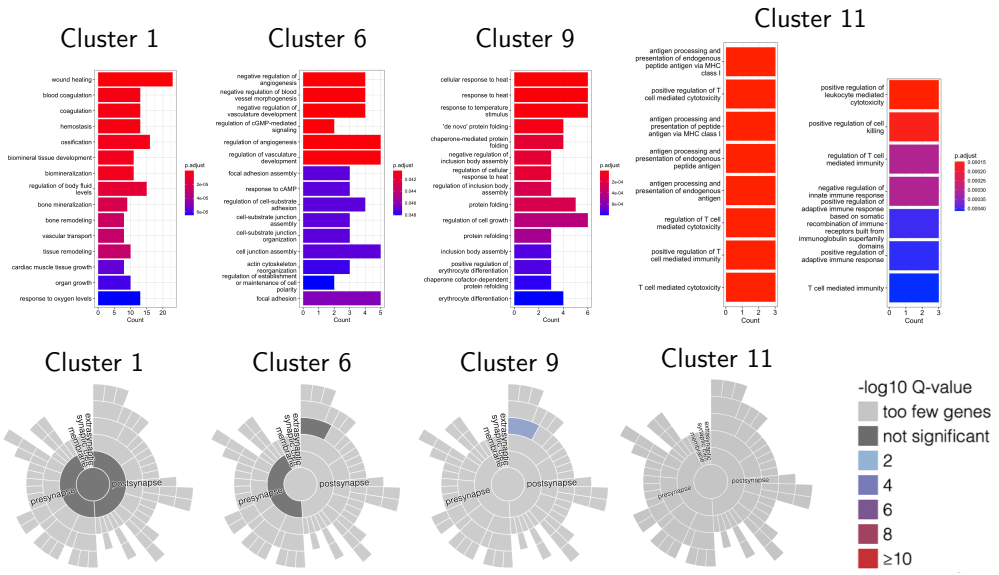


Figure 4.15: The ALL terms GO analysis (first row) and SynGO analysis (second row) results for our identified **reactive** DE clusters.

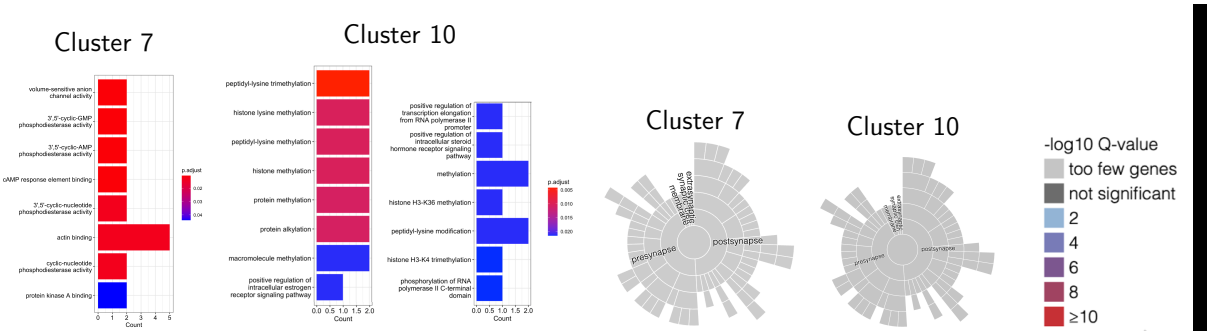


Figure 4.16: The ALL terms GO analysis (first row) and SynGO analysis (second row) results for our identified **other** DE clusters.

Comparing to WGCNA modules. We also compare our identified 11 clusters with the 35 identified WGCNA modules. Figure 4.17 shows the proportion of WGCNA module genes for each of our clusters. We can see that, our identified active DE gene clusters mostly lie in the neuron-type (e.g. ExNeuron and InNeuron) WGCNA modules, and the reactive DE gene clusters mostly lie in the non-neuron type (e.g. Astrocytes and Endothelial) WGCNA modules. It is found in previous studies that ASD risk genes are more enriched in neuron-type of cells (Fu et al., 2022), which just concurs with our results.

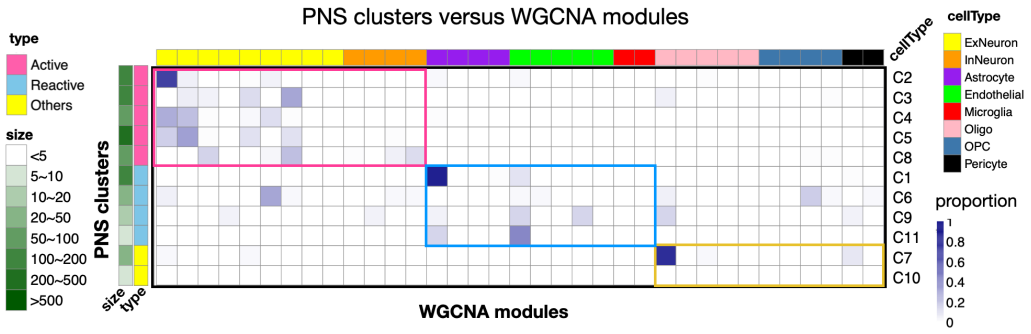


Figure 4.17: The comparison between our PNS clusters and the WGCNA modules found in (Gandal et al., 2022). The heatmap displays the proportion of genes from the PNS clusters that overlap with the corresponding WGCNA modules. The hotpink, skyblue, and yellow frames are used to highlight the prominent overlapping patterns for the "active," "reactive," and "other" PNS clusters, respectively.

Direction of differential expression. We also look into the direction of the differential expression for active and reactive DE genes for each gene cluster, in order to see whether the active and reactive DE tend to be more highly/lowly expressed in ASD cases (i.e. upregulated/downregulated). In Figure 4.18, where we plot the histogram of log fold change (logFC) for genes in each cluster, we can see that our reactive DE genes and also reactive DE gene clusters as a whole tend to be more upregulated in ASD, while active ones tend to be more downregulated in ASD. Note that this is not a result of our gene filtration: for the 8000 genes we selected to do the analysis, the mean of logFC is -0.034 and the standard deviation is 0.1. It is our PNS network and joint-HMRF analysis that makes most of the downregulated genes stand out. This result also concurs with previous findings: (Gandal et al., 2022) found that within the "Attenuation of Transcriptomic Regional Identity" genes [5], the downregulated gene set showed broad enrichment for neuronal cell-type-specific markers and RNA processing pathways, and contained many transcription factors (just like our active clusters); while the upregulated genes also contained many transcription factors and were enriched for oligodendrocyte progenitor cell (OPC) and astrocyte cell-type markers along with metabolic and development pathways (just like our reactive clusters).

[5] Genes that associated with an attenuation of typical gene expression differences between two regions frontal and temporal lobe in ASD.

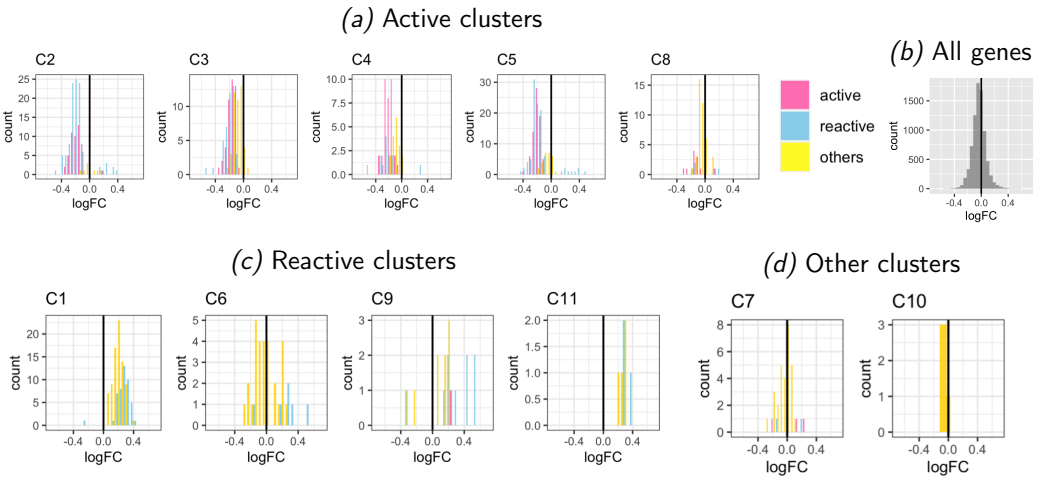


Figure 4.18: The whole cortex differential direction of active and reactive DE genes in different modules. In each plot we show the histogram of log fold change (logFC) for genes in the corresponding cluster. Positive logFC means a gene is more expressed in ASD, while negative means less expressed.

4.5 Conclusion

In this project, we study the problem of the differential expression mechanism in Autism (ASD). Though lots of genes that are differentially expressed between ASD and neurotypical brains are identified, their role in ASD development remains a mystery. A gene can be differentially expressed to cause the phenotype ("active"), or it can be differentially expressed because of the phenotype ("reactive"). To deconvolve the DE mechanism, we integrate information from other sources, the Transmission And De novo Association (TADA) analysis, which directly measures how likely a gene is to be the cause of ASD. The integration task is nontrivial as we found that TADA-significant genes (genes that carry mutations significantly associated with ASD) have nearly no overlap with DE-significant genes (genes that are significantly differentially expressed). Therefore, to bridge them together, we resort to their common underlying biological mechanism: the interaction among genes (i.e. gene network). Our method involves three major steps: 1) gene network construction 2) network regularization of DE and TADA signal 3) active and reactive community detection. We develop novel methods in each of these three steps, to address long-overlooked or newly-appeared challenges. Our contributions and findings can be summarized in the following two aspects.

First, we contribute the first-ever effort to systematically investigate various gene network concepts in the application of ASD risk gene modeling, with special inclusion of nonlinearity and group interaction. For completeness, we also developed a novel nonlinear variant of the ensemble group interaction concept. Instead of

far-from-truth simulation, we directly evaluate the usefulness of each gene network concept by their ability to induce biological meaningful results in the downstream analysis of the DE mechanism. We found that a linear type of group interaction appears most useful in our task. This surprising finding also concurs with recent findings in (Manicka et al., 2023), that biological regulation tends to be less nonlinear than expected. Our effort benefits beyond just the problem we study here: the collection and implementation of various gene network concepts provide convenience for other scientific studies. Also, our findings provide insights into the correct direction for the development of gene network estimation: nonlinearity, though interesting, might often not be worth the trouble due to the estimation difficulty and the rarity of the case.

Second, we propose a novel Hidden Markov Random Field (HMRF) model to model DE and TADA signals jointly with the gene network information. Our approach expands the overlap between DE and TADA signals in a meaningful way: the signals are regularized together carefully by "message passing" while traversing the gene network. With this network regularization of the two sources of information, we are able to identify a collection of "active" and "reactive" DE gene clusters. We find that the identified active clusters are related to synaptic and neuronal functions, and are enriched in neuron-type cells, which agrees with the common belief that ASD is caused by misfunction of neuronal activities. On the other hand, the reactive clusters are mostly related to responsive functions and are enriched in nonneuron-type cells, which brings new insights into the effect of ASD on the misfunction of nonneuronal activities.

4.6 Appendices

4.6.1 Details on sparse additive CCA

Consider the following family of non-linear transformations that transform each dimension using a set of basis functions $\{\eta_1, \ldots, \eta_L\}$:

$$\mathcal{S} := \left\{ s : \mathbb{R} \to \mathbb{R},\ x \mapsto \sum_{l=1}^{L} \beta_l \eta_l(x),\ \text{where } \beta_l \in \mathbb{R}\ \forall\ l \in [L] \right\}. \tag{4.21}$$

For $X \in \mathbb{R}^{n\times p}$, $Y \in \mathbb{R}^{n\times q}$, we consider additive model to combine all the features. That is, we are solving

$$\max_{\substack{\boldsymbol{\beta}_X^i \in \mathbb{R}^p,\ i\in[p] \\ \boldsymbol{\beta}_Y^j \in \mathbb{R}^q,\ j\in[q]}} corr\left(\boldsymbol{\psi}(X), \boldsymbol{\phi}(Y)\right), \tag{4.22}$$

where

$$\boldsymbol{\psi}(X) := \left(\sum_{i=1}^{p} \psi^i(X_{1i}), \ldots, \sum_{i=1}^{p} \psi^i(X_{ni}) \right)^\top \in \mathbb{R}^{n\times 1},$$

$$\boldsymbol{\phi}(Y) := \left(\sum_{j=1}^{q} \phi^j(Y_{1j}), \ldots, \sum_{j=1}^{p} \phi^j(Y_{nj}) \right)^\top \in \mathbb{R}^{n \times 1},$$

and ϕ^i and ψ^j are from function class $\mathcal{S}$ for all $i \in [p]$ and $j \in [q]$, and we take the representation

$$\psi^i(\cdot) := \sum_{l=1}^{L} \beta_{X,l}^i \eta_l(\cdot), \quad \phi^j(\cdot) := \sum_{l=1}^{L} \beta_{Y,l}^j \eta_l(\cdot). \tag{4.23}$$

Write in terms of β coefficient, we get:

$$\boldsymbol{\psi}(X)^\top \boldsymbol{\phi}(Y) := \sum_{k=1}^{n} \left(\sum_{i=1}^{p} \sum_{l=1}^{L} \beta_{X,l}^i \eta_l(X_{ki}) \right) \left(\sum_{j=1}^{q} \sum_{l=1}^{L} \beta_{Y,l}^j \eta_l(Y_{kj}) \right). \tag{4.24}$$

If we define $M_X \in \mathbb{R}^{n \times pL}$, and $M_Y \in \mathbb{R}^{n \times qL}$ where

$$M_X := \begin{bmatrix} \eta_1(X_{1,1}) & \ldots & \eta_1(X_{1,p}) & \ldots & \eta_L(X_{1,p}) \\ \eta_1(X_{2,1}) & \ldots & \eta_1(X_{2,p}) & \ldots & \eta_L(X_{2,p}) \\ & & & & \\ \eta_1(X_{n,1}) & \ldots & \eta_1(X_{n,p}) & \ldots & \eta_L(X_{n,p}) \end{bmatrix},$$

$$M_Y := \begin{bmatrix} \eta_1(Y_{1,1}) & \ldots & \eta_1(Y_{1,p}) & \ldots & \eta_L(Y_{1,p}) \\ \eta_1(Y_{2,1}) & \ldots & \eta_1(Y_{2,p}) & \ldots & \eta_L(Y_{2,p}) \\ & & & & \\ \eta_1(Y_{n,1}) & \ldots & \eta_1(Y_{n,p}) & \ldots & \eta_L(Y_{n,p}), \end{bmatrix}$$

and $\boldsymbol{w}_X \in \mathbb{R}^{pL \times 1}$, and $\boldsymbol{w}_Y \in \mathbb{R}^{qL \times 1}$ where

$$\boldsymbol{w}_X := (\beta_{X,1}^1, \ldots \beta_{X,L}^1, \ldots, \beta_{X,1}^p, \ldots, \beta_{X,L}^p)^\top,$$
$$\boldsymbol{w}_Y := (\beta_{Y,1}^1, \ldots \beta_{Y,L}^1, \ldots, \beta_{Y,1}^p, \ldots, \beta_{Y,L}^p)^\top.$$

Then we can write (4.24) as

$$\boldsymbol{\psi}(X)^\top \boldsymbol{\phi}(Y) := \boldsymbol{w}_X^\top M_X^\top M_Y \boldsymbol{w}_Y := \boldsymbol{w}_X^\top \widetilde{C}_{XY} \boldsymbol{w}_Y, \tag{4.25}$$

where $\widetilde{C}_{XY} := M_X^\top M_Y$.

Similarly, define $\widetilde{C}_{XX} := M_X^\top M_X$, and $\widetilde{C}_{YY} := M_Y^\top M_Y$, we have problem (4.22) be rewritten as a classic CCA problem:

$$\max_{\substack{\boldsymbol{w}_X \in \mathbb{R}^p \\ \boldsymbol{w}_Y \in \mathbb{R}^q}} \boldsymbol{w}_X^\top \widetilde{C}_{XY} \boldsymbol{w}_Y, \quad \text{subject to } \boldsymbol{w}_X^\top \widetilde{C}_{XX} \boldsymbol{w}_X = 1,\ \boldsymbol{w}_Y^\top \widetilde{C}_{YY} \boldsymbol{w}_Y = 1. \tag{4.26}$$

Suppose $\boldsymbol{w}_X$ and $\boldsymbol{w}_Y$ can be respectively divided into L and M non-overlapping groups: $\boldsymbol{w}_X^{(g)} \in \mathbb{R}^{p_g \times 1}, g \in 1, \ldots, G$; and $\boldsymbol{w}_Y^{(m)} \in \mathbb{R}^{q_m \times 1}, q \in 1, \ldots, M$. Then

consider the (adaptive) group Lasso (GL1) penalty for $\boldsymbol{w}_X$ and $\boldsymbol{w}_Y$ as follows:

$$\Omega_{GL_1}(\boldsymbol{w}_X) := \sum_{g=1}^{G} \sqrt{p_g}||\boldsymbol{w}_X^{(g)}||_2, \quad \text{and } \Omega_{GL_1}(\boldsymbol{w}_Y) := \sum_{m=1}^{G} \sqrt{q_m}||\boldsymbol{w}_Y^{(m)}||_2.$$

Based on the definition of GL_1 penalty, we propose the following group sparse CCA:

$$\begin{aligned} \min_{\boldsymbol{w}_X, \boldsymbol{w}_Y} & - \boldsymbol{w}_X^\top \widetilde{C}_{XY} \boldsymbol{w}_Y \\ & \text{subject to} ||\widetilde{X}\boldsymbol{w}_X||^2 \leq 1, \Omega_{GL1}(\boldsymbol{w}_X) \leq c_1, \\ & \quad ||\widetilde{Y}\boldsymbol{w}_Y||^2 \leq 1, \Omega_{GL1}(\boldsymbol{w}_Y) \leq c_2. \end{aligned} \tag{4.27}$$

The Lagrangian form of the above problem is:

$$\begin{aligned} & \mathcal{L}(\boldsymbol{w}_X, \boldsymbol{w}_Y) \\ & = -\boldsymbol{w}_X^\top \widetilde{C}_{XY} \boldsymbol{w}_Y + \lambda_1 \, \Omega_{GL1}(\boldsymbol{w}_X) + \lambda_2 \, \Omega_{GL1}(\boldsymbol{w}_Y) + \eta_1 ||X\boldsymbol{w}_X||^2 + \eta_2 ||Y\boldsymbol{w}_Y||^2, \end{aligned}$$

where $\lambda_1 \geq 0, \lambda_2 \geq 0, \eta_1 \geq 0, \eta_2 \geq 0$ are Lagrange multipliers. To minimize $\mathcal{L}(\boldsymbol{w}_X, \boldsymbol{w}_Y)$, we use the alternating iterative algorithm based on a block coordinate descent method to optimize $\boldsymbol{w}_X$ for a fixed $\boldsymbol{w}_Y$ and vice versa.
Specifically, to learn $\boldsymbol{w}_X$: fix $\boldsymbol{w}_Y$ and let $\boldsymbol{z} = \widetilde{C}_{XY}\boldsymbol{w}_Y$, then the target function become:

$$\begin{aligned} \mathcal{L}_{w_X}(\boldsymbol{w}_X, \lambda_1, \eta_1) &= -\boldsymbol{w}_X^\top \boldsymbol{z} + \lambda_1 \sum_{g=1}^{G} \sqrt{p_g}||\boldsymbol{w}_X^{(g)}||_2 + \eta_1 ||X\boldsymbol{w}_X||_2^2 \\ &= -\sum_{g=1}^{G} \left(\boldsymbol{w}_X^{(g)\top} \boldsymbol{z}^{(g)} + \lambda_1 \sqrt{p_g}||\boldsymbol{w}_X^{(g)}||_2 \right) - \sum_{i=1}^{n} \sum_{g=1}^{G} \eta_1 \left(\boldsymbol{w}_X^{(g)^\top} \boldsymbol{x}_i^{(g)} \right)^2, \end{aligned}$$

where $\boldsymbol{x}_i \in \mathbb{R}^{p \times 1}$ is the i-th row of X. Since $\mathcal{L}_{w_X}(\boldsymbol{w}_X)$ is strictly convex and separable, the block coordinate descent algorithm must converge to its optimal solution (Tseng, 2001).

Then the subgradient of $\mathcal{L}_{w_X}$ with respect to $\boldsymbol{w}_X^{(g)}$ is

$$\begin{aligned} \partial_{w_X^{(g)}} \mathcal{L}_{w_X} &= -\boldsymbol{z}^{(g)} + \lambda_1 \sqrt{p_g} \boldsymbol{s}^{(g)} + 2\eta \sum_{i=1}^{n} \boldsymbol{x}_i^{(g)} (\boldsymbol{x}_i^{(g)\top} \boldsymbol{w}_X^{(g)}) \\ &= -\boldsymbol{z}^{(g)} + \lambda_1 \sqrt{p_g} \boldsymbol{s}^{(g)} + 2\eta \left(\sum_{i=1}^{n} \boldsymbol{x}_i^{(g)} \boldsymbol{x}_i^{(g)\top} \right) \boldsymbol{w}_X^{(g)} \\ &= -\boldsymbol{z}^{(g)} + \lambda_1 \sqrt{p_g} \boldsymbol{s}^{(g)} + 2\eta (X^\top X)^{(g)} \boldsymbol{w}_X^{(g)} \end{aligned}$$

where

$$\boldsymbol{s}^{(g)} = \begin{cases} \frac{\boldsymbol{w}_X^{(g)}}{||\boldsymbol{w}_X^{(g)}||_2}, & \text{if } \boldsymbol{w}_{\boldsymbol{X}}^{(\boldsymbol{g})} \neq \mathbf{0} \\ \in \{\boldsymbol{s} \in \mathbb{R}^{p_g \times 1} : ||s||_2 \leq 1\} & \text{otherwise.} \end{cases}$$

Then according to KKT condition, the solution needs to satisfy $0 \in \partial_{w_X^{(g)}} \mathcal{L}_{w_X}$, and therefore we get

$$2\eta(X^\top X)^{(g)} \boldsymbol{w}_X^{(g)} = \boldsymbol{z}^{(g)} - \lambda_1 \sqrt{p_g} \boldsymbol{s}^{(g)}.$$

Then a necessary and sufficient condition for θ to be zero is that the system of equations

$$\boldsymbol{z}^{(g)} = \lambda_1 \sqrt{p_g} \boldsymbol{s}^{(g)} \quad \text{have a solution with } ||\boldsymbol{s}^{(g)}||_2 \leq 1,$$

or equivalently, whether $||\boldsymbol{z}^{(g)}||_2 \leq \lambda_1 \sqrt{p_g}$.

Then if $||\boldsymbol{z}^{(g)}||_2 \geq \lambda_1 \sqrt{p_g}$, then we have $\boldsymbol{w}_X^{(g)} \neq 0$. Therefore the target function we optimize over is the sum of convex differentiable functions, and hence we can use gradient descent to obtain the global minimum. The gradient is just

$$\nabla_{w_X^{(g)}} \mathcal{L}_{w_X} = -\boldsymbol{z}^{(g)} + \lambda_1 \sqrt{p_g} \frac{\boldsymbol{w}_X^{(g)}}{||\boldsymbol{w}_X^{(g)}||_2} + 2\eta(X^\top X)^{(g)} \boldsymbol{w}_X^{(g)}.$$

We use backtracking line search to determine the step size. We use warm start, that is set initialization as the solution of the nonsparse variant. Also, for fast computation, we only take a few (like 10) gradient steps in each iteration (instead of solving it towards convergence). This turns out to give a good enough solution. We use cosine and sine basis functions with order 3 and normalize the data to range [0,1] before putting them into the additive sparse CCA solver.

Simulations. In the following, we use simulations to show the validity of our propose Sparse Additive CCA. We consider $p = q = 15$, $n = 50$, and each sample in X and Y are generated by the following:

$$\begin{aligned} &X_i \sim N(0,1) \text{ for all } i = 1, \ldots, p; \\ &Y_1 = f(X_2); \quad Y_i \sim N(0,1) \text{ for all } i \neq 1; \end{aligned}$$

That is, there is only one dimension in each set that is relevant to the others, and the true canonical weights should be $\boldsymbol{w}_X = (0, 1, 0, \ldots, 0)$ and $\boldsymbol{w}_Y = (1, 0, 0, \ldots, 0)$. We consider various type f, both linear and nonlinear, and run 10 repeats of the experiment. In Figure 4.19, we plot the average of the estimated canonical weights over the 10 experiments for each method: CCA; Sparse CCA and our propose Sparse Additive CCA. We can see that only our method can recover the truth with

the highest precision across all the settings, while the other method collapses when handling some nonlinear relationships.

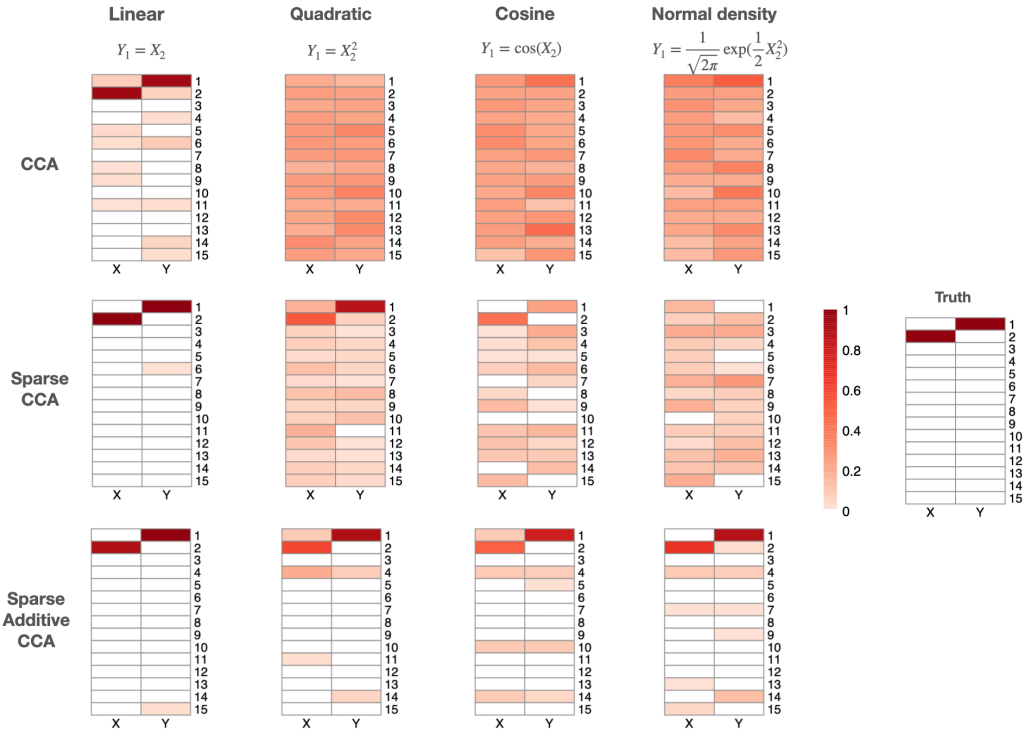


Figure 4.19: Summary of simulation results for CCA, Sparse CCA and our propose Sparse Additive CCA.

4.6.2 Simulations on joint HMRF

In this section, we conduct a simple simulation study to further justify for our joint-HMRF method. To simulate the data, we first consider the true graph is Figure 4.20, which contains 654 nodes and presents a nice cluster structure and a nice power-law distribution of edge degree; and we set the rest parameters in the MRF model (4.13) as $b_{01} = b_{02} = -3$, $b_{03} = 0$, $b_{11} = b_{12} = 2$.

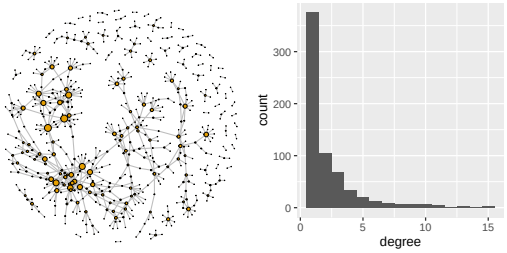


Figure 4.20: The summary of the true graph we considered.

Initial values of $\boldsymbol{I}$ are randomly assigned to each node in the simulated graph and we let each class of the nodes have the same size. We then use Gibbs sampling to update $\boldsymbol{I}$ for 1000 iterations and withdraw the first 200 iterations to avoid nonstationarity in the initial period. Then we compute marginal probabilities for each node and set the true hidden states as the states that give the maximum marginal probabilities for each node. Then we generate z-scores for DE and TADA using (4.11) and (4.12) given the true hidden states, with parameters $\mu_1 = \mu_2 = 2$, $\sigma_{01} = \sigma_{02} = \sigma_{11} = \sigma_{12} = 1$. Figure 4.21(a) visualize the true hidden states, and Figure 4.21(b,c) visualize the simulated DE and TADA z-scores.

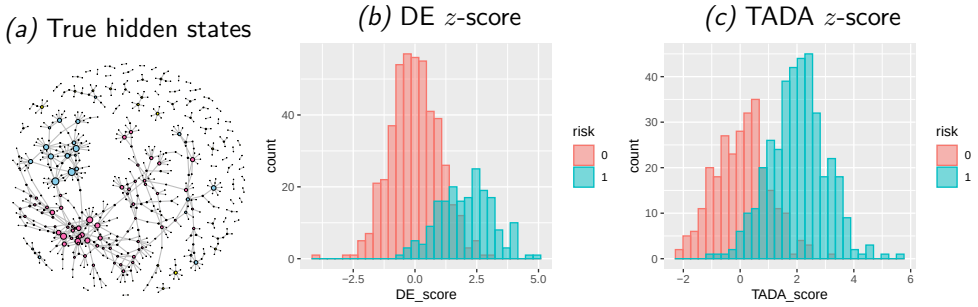


Figure 4.21: The summary of the sampled data from the true joint HMRF model. **(a)** The sampled true hidden states, where the color of the node indicates the hidden states of the node: hotpink for "active"; skyblue for "reactive"; yellow for "others". **(b)** The sampled DE z-score. **(c)** The sampled TADA z-score.

Then we estimate the gene hidden states using the sampled DE and TADA z-score and the graph as input. We consider the following three methods: (1) Thred Input: Directly thresholding on the input DE and TADA z-score to determine gene states (like (4.16)). (2) Separate HMRF: the method we described as the baseline for comparison in Section 4.4.4, which run DAWN twice, one using DE z-score as input and another using TADA z-score as input, and then threshold the output DAWN-DE and DAWN-TADA p-values to determine gene states. (3) Joint HMRF: our proposed method described in Section 4.3.3.

In Figure 4.22, we visualize the estimated hidden states using the three different methods described above, and we can see that our method performs the best. To quantitatively evaluate the results, we use a weighted F1 score, to account for the fact that in practice we care more about nodes with high degrees in the graph.

$$F_1(\boldsymbol{I}, \widehat{\boldsymbol{I}}; w) := \frac{1}{4}\sum_{s=1}^{4} 2\frac{\text{precision}(\boldsymbol{I}, \widehat{\boldsymbol{I}}; w, s) * \text{recall}(\boldsymbol{I}, \widehat{\boldsymbol{I}}; w, s)}{\text{precision}(\boldsymbol{I}, \widehat{\boldsymbol{I}}; w, s) + \text{recall}(\boldsymbol{I}, \widehat{\boldsymbol{I}}; w, s)};$$

where

$$\text{precision}(\boldsymbol{I}, \widehat{\boldsymbol{I}}; w, s) = \frac{\sum_i w_i \mathbf{1}_{I_i=\widehat{I}_i=s}}{\sum_i w_i \mathbf{1}_{\widehat{I}_i=s}}; \quad \text{recall}(\boldsymbol{I}, \widehat{\boldsymbol{I}}; w, s) = \frac{\sum_i w_i \mathbf{1}_{I_i=\widehat{I}_i=s}}{\sum_i w_i \mathbf{1}_{I_i=s}}.$$

In Figure 4.23, we show the summarized weighted F1 score for the three methods we considered across 10 repetitions, where we set weights w as the node degree.

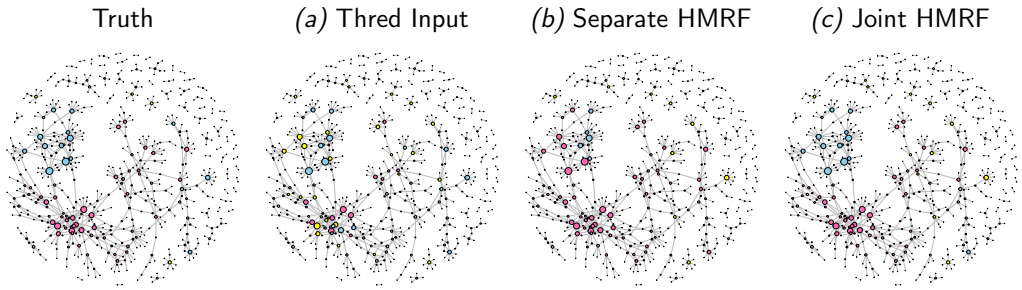


Figure 4.22: Visualization of hidden states using different estimation methods. In each plot, the color of the node indicates the hidden states of the node: hotpink for "active"; skyblue for "reactive"; yellow for "others". **(a)** The hidden states are estimated by direct thresholding the input DE and TADA z-scores using cutoff $\Phi^{-1}(1-0.05)$; **(b)** The hidden states are estimated by direct thresholding the DAWN DE and TADA p-values using cutoff 0.05; **(c)** The hidden states are estimated using our proposed Joint-HMRF method with cutoff $C = \Phi^{-1}(1-0.05)$ in initialization.

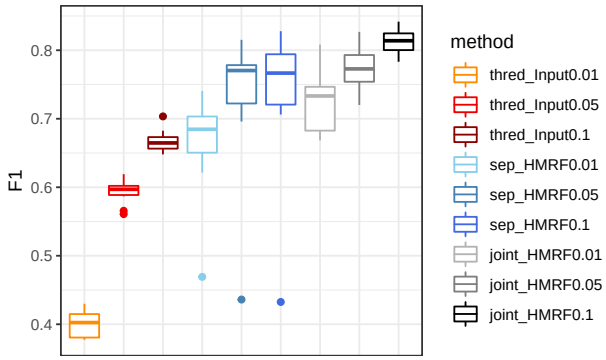


Figure 4.23: The weighted F1 for different hidden states estimation methods across 10 experiment repetitions, where the node weights are proportional to the node degrees. `thred-Input0.01`,`thred-Input0.05`,`thred-Input0.1` represents Thred Input with cutoff $\Phi^{-1}(1-0.01)$, $\Phi^{-1}(1-0.05)$, $\Phi^{-1}(1-0.1)$ respectively; `sep-HMRF0.01`,`sep-HMRF0.05`,`sep-HMRF0.1` represents Separate HMRF with cutoff 0.01, 0.05, 0.1 respectively; `joint-HMRF0.01`,`joint-HMRF0.05`,`joint-HMRF0.1` represents Joint HMRF with cutoff $C = \Phi^{-1}(1-0.01)$, $\Phi^{-1}(1-0.05)$, $\Phi^{-1}(1-0.1)$ in the initialization.

4.6.3 Additional plots

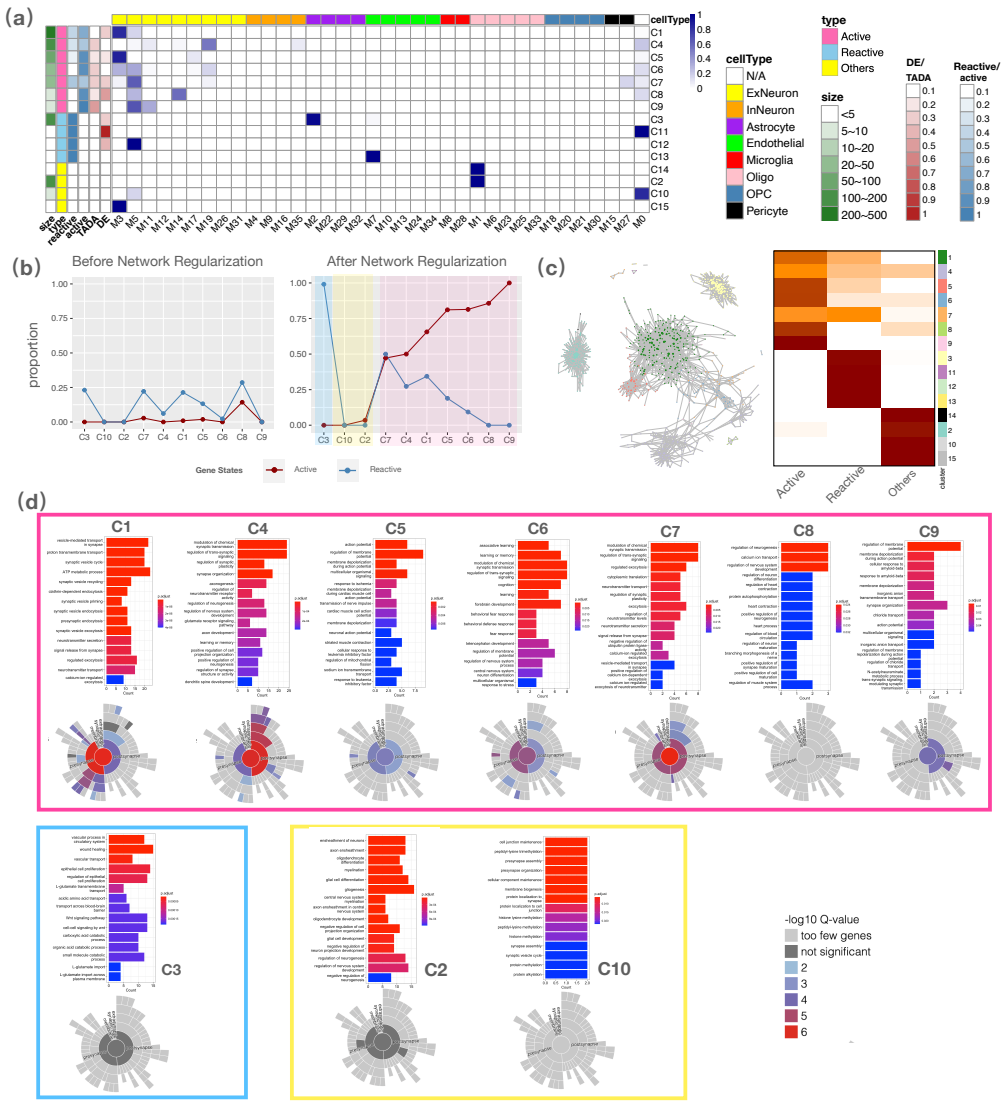


Figure 4.24: The summary of results using Pearson network. (a) the comparison of found clusters with the WGCNA module as in Figure 4.17. (b) The proportion of active and reactive genes in each cluster before and after network regularization; (c) The clusters were visualized on the network, and a heatmap was generated to display the proportions of active, reactive, and other genes within each cluster. (d) The GO and SynGO results for the identified gene clusters are presented below. Active clusters are highlighted with a hotpink frame, reactive clusters with a skyblue frame, and other clusters with a yellow frame.

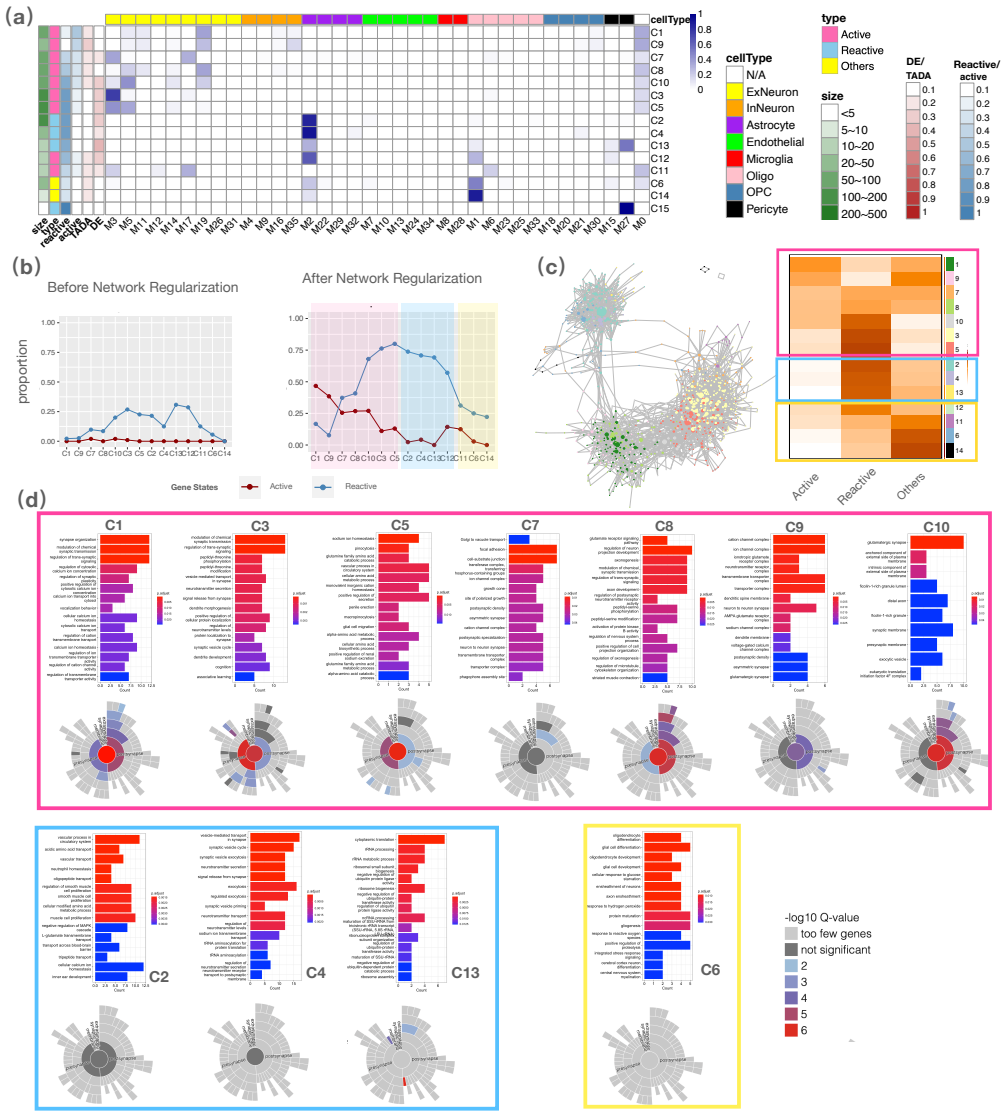


Figure 4.25: The summary of results using EnPAC network. (a) the comparison of found clusters with the WGCNA module as in Figure 4.17. (b) The proportion of active and reactive genes in each cluster before and after network regularization; (c) The clusters were visualized on the network, and a heatmap was generated to display the proportions of active, reactive, and other genes within each cluster. (d) The GO and SynGO results for the identified gene clusters are presented below. Active clusters are highlighted with a hotpink frame, reactive clusters with a skyblue frame, and other clusters with a yellow frame.

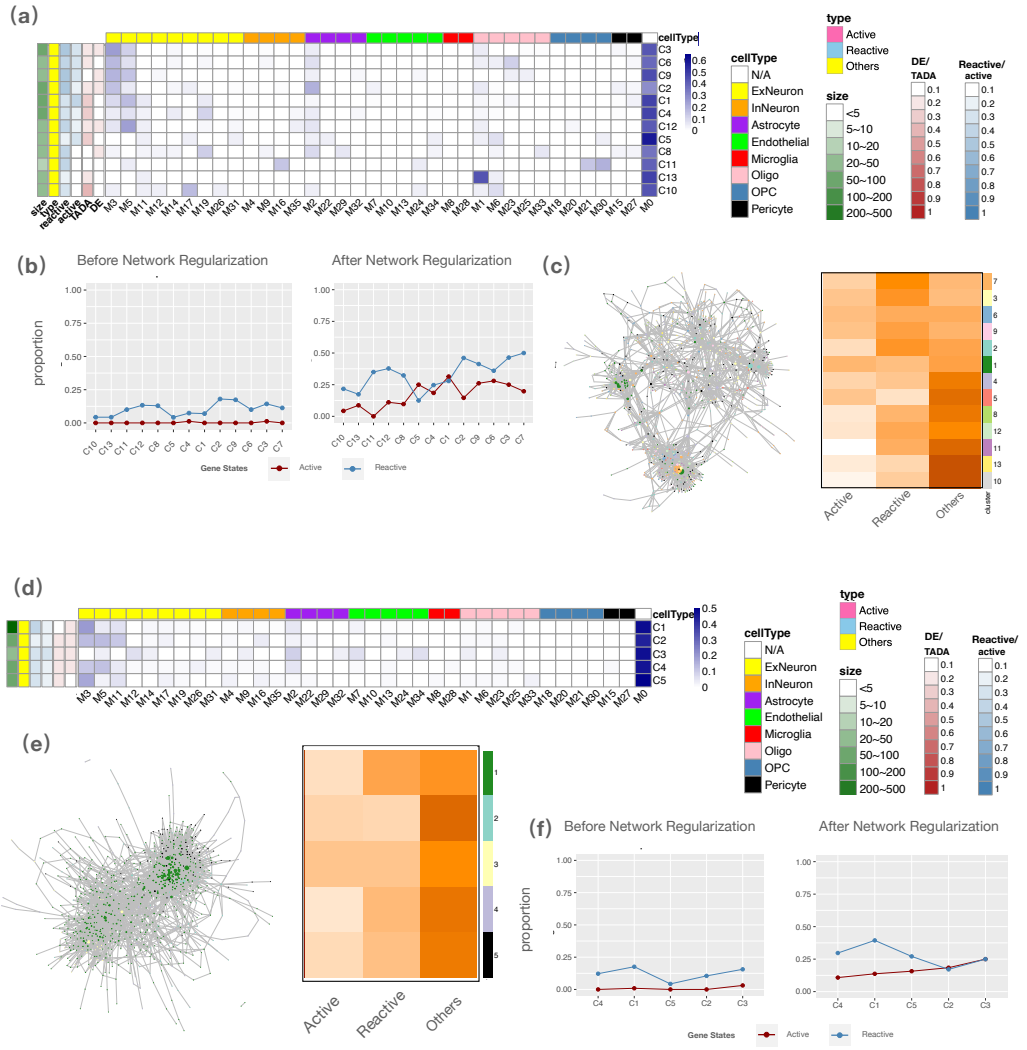


Figure 4.26: The summary of results for networks that failed to give meaningful active and reactive clusters. (a-c) are for GENIE3 networks; and (d-f) are for aLDG networks. EPC even failed to give meaningful clusters, therefore we omit for analysis. (a)/(d) the comparison of found clusters using GENIE3/aLDG networks with the WGCNA module as in Figure 4.17. (b)/(f) The proportion of active and reactive genes in each cluster from GENIE3/aLDG networks before and after network regularization; (c)/(e) The clusters were visualized on the GENIE3/aLDG networks, and a heatmap was generated to display the proportions of active, reactive, and other genes within each cluster.

Five

Conclusions and Future work

This thesis focuses on the study of gene networks, which describe the collaborative relationships among genes. It addresses several challenges in the field of statistical gene network analysis, including benchmarking gene network estimation methods, developing nonlinear gene network estimation techniques, and exploiting gene networks to understand genes associated with Autism Spectrum Disorders (ASD).

In Chapter 2, the thesis introduces a benchmarking tool for evaluating imputation methods on gene coexpression estimation. It presents a new simulation tool capable of generating realistic data for homogeneous and heterogeneous cell groups, as well as complex cell group relationships such as tree structures and cell trajectories. The tool specifically incorporates gene coexpression patterns, allowing for the assessment of the impact of gene expression denoising methods on downstream gene coexpression estimation.

Chapter 3 addresses the limitations of existing gene coexpression estimation methods in capturing nonlinear relationships. It proposes a novel dependence measure, the averaged Local Density Gap (aLDG), which accumulates local dependence and can detect non-linear, non-monotone, and non-global relationships. The chapter establishes the consistency and robustness of the proposed measure and demonstrates its superiority over a wide range of existing dependence measures.

In Chapter 4, the thesis explores the application of different types of gene network concepts in identifying active genes associated with ASD. Particularly, it introduces a novel gene group interaction measure to address challenges when the true gene groups are unknown in nonlinear setups. By employing a unified network-assisted modeling approach, the chapter identifies distinct "active" and "reactive" gene communities associated with ASD, shedding light on the biological mechanisms underlying the disorder.

Overall, the thesis contributes to the field of gene network analysis by developing innovative methods, benchmarking tools, and applying gene networks to gain insights into complex biological processes, such as gene coexpression and the etiology of ASD.

5.1 Future directions

Higher level higher resolution cell heterogeneity characterization. In Chapter 3, the thesis explores the concept of cell-specific gene co-expression and its potential applications in single-cell data analysis. While the chapter focuses on aggregating cell-specific gene co-expression to obtain a global gene co-expression measure, there is an interesting direction to investigate the utility of cell-specific gene co-expression without aggregation. In single-cell data analysis, the heterogeneity of cells is commonly captured through marginal or first-moment information, such as low-dimensional embeddings and clustering analysis. However, these approaches often overlook the higher-order interactions among genes. For instance, if two cell clusters differ only in their covariance structure (i.e., gene-by-gene interactions) but not in their marginal distribution levels, existing clustering methods may fail to differentiate these two clusters. To address this limitation, some researchers Ghazanfar et al. (2020); Dai et al. (2019) have been advocating a higher-order and higher-resolution analysis that focus on differentiating cells based on gene relational information. By considering the interplay and relationships among genes within individual cells, these methods can uncover novel cell heterogeneity that may not be captured by traditional approaches.

Exploring the potential of cell-specific gene co-expression without aggregation opens up new avenues for understanding cell heterogeneity and its underlying molecular mechanisms. By directly leveraging the gene relational information within cells, it may be possible to discover previously hidden subpopulations, identify unique cell states, and gain insights into the regulatory dynamics of cellular processes. This direction suggests a shift towards a more comprehensive analysis of single-cell data, considering both the marginal distributions and the higher-order gene-gene interactions. By integrating these aspects, researchers can potentially uncover more nuanced and detailed information about cellular heterogeneity and its functional implications.

In Chapter A we make several attempts in this direction: we generalize Dai et al. (2019)'s work, which we characterize as a non-linear data transformation, and subsequently define a general class of data transformation that allows capturing pairwise local distributional (PLoD) information. Initial results suggest that PLoD allows novel characterization of cell heterogeneity from the gene-gene interaction perspective, as well as localization of gene pairs whose relation drives the observed cell heterogeneity.

First, in the second-order clustering problem, where clusters differ only in terms of covariance, the PLoD transformation-induced clustering method achieves near-optimal performance compared to other competitive methods that collapse in this scenario. This demonstrates the effectiveness of PLoD in capturing the covariance structure and enabling accurate clustering. Second, in a nontraditional signal-noise mixture model where the signal is determined by variance rather than the mean,

the PLoD transformation-induced feature selection method enables exact signal recovery. This is in contrast to canonical methods like sparse PCA, which struggle in such scenarios. These results highlight the potential of PLoD-induced methods in capturing and extracting meaningful information from complex data settings.

However, when applied to real data, the PLoD-induced methods do not outperform existing methods. This suggests that the nontraditional problem settings investigated here may not commonly emerge in real single-cell RNA-seq (scRNA-seq) data. Nevertheless, this finding opens up interesting avenues for further exploration. It raises the question of whether cell clusters in scRNA-seq data are determined predominantly by marginal information alone or if gene-gene interactions also play a prominent role.

Region-specifc partial gene network estimation. In the application described in Chapter 4, we conducted the analysis using the whole-cortex gene network. However, an interesting follow-up direction would be the analysis on a region-specific level. Previous studies, such as Gandal et al. (2022), have identified region-specific ASD DE genes, with the smallest region, BA17, showing the highest number of DE genes associated with ASD. However, these region-specific DE genes did not show enrichment in known ASD genetic risk genes identified through methods like TADA.

To gain a better understanding of these region-specific ASD differentially expressed genes, the proposed analysis framework in the thesis can be extended to incorporate a region-specific gene network. Constructing a region-specific Partial Network Similarity (PNS) network is challenging due to the limited sample size for each region. For instance, BA17 has only 28 samples from neurotypical brains, making it difficult to estimate reliable networks for specific regions. Simulations conducted in Chapter B reveal that with only 100 samples, a PNS network for 3000 genes can result in a high False Discovery Rate (FDR) of 50%. Therefore, a joint analysis considering all the regions becomes necessary. Assuming that the region-specific networks share a substantial proportion of edges while the region-specific edges are sparse, modeling and estimating them jointly can leverage the shared information and improve the accuracy of the analysis.

In Chapter B, we delves further into this direction and proposes two methods for joint estimation. While these methods perform better than estimating each region separately, they still exhibit a relatively high FDR. The thesis also explores procedures for network estimation with FDR control, such as the high-dimensional graphical knockoff filter proposed in recent studies (Li and Maathuis, 2021; Zhou et al., 2022). However, the performance improvement is limited as the FDR control results are guaranteed under strict conditions that may not hold in practical scenarios. Therefore, additional efforts are needed to achieve accurate region-specific PNS network estimation given the extremely small sample size.

One potential approach to address this challenge is to switch to a marginal gene network, such as Pearson correlation, which requires fewer samples to obtain a

reliable estimation. By adapting the methodology and considering marginal gene networks instead of PNS, it may be possible to overcome the limitations posed by the small sample size and improve the accuracy of the region-specific analysis.

5.2 Next Fronteriors

Over the past two decades, significant progress has been made in characterizing gene expression data, both in terms of marginal analysis and relational analysis. Marginal analysis techniques, such as low-dimensional embedding, differential expression, and cell clustering, have been widely used to understand gene expression patterns at the individual gene and cell levels. On the other hand, relational analysis approaches, such as gene regulatory and co-expression networks, have provided insights into the interactions /relationships between genes.

However, there has been relatively less effort in integrating these two aspects together. Researchers have recognized the power of gene relational information when it comes to analyzing gene expression data in a marginal context. For example, in genome-wide association studies (GWAS), linkage disequilibrium (LD) among SNPs in a gene have been leveraged to correct for dependencies among tests in the analysis Yurko et al. (2021). In differential expression analysis, gene covariance information has been utilized for improved sample selection Lin et al. (2021). In the integration of multi-omics data, gene regulatory networks have been employed to align information from different omics modalities Cao and Gao (2022).

This thesis highlights three key scientific questions that merit further exploration. Firstly, it is crucial to investigate what types of gene relational information are most useful in different application scenarios. Understanding the specific types of gene interactions and relationships that are relevant to various analytical tasks can guide the development of appropriate methodologies and approaches.

Secondly, there is a need to explore how to effectively incorporate gene relational information into the existing analysis pipeline. Developing methodologies that can seamlessly integrate relational information with marginal analysis techniques will enable a more comprehensive understanding of gene expression patterns and their biological significance.

Lastly, the possibility of jointly learning both marginal and relational information should be examined. Existing approaches routinely learn marginal structure (cell structure) first and then learn relational structure (gene relationship). By simultaneously modeling and estimating the marginal characteristics of gene expression data along with their relational aspects, it may be possible to uncover novel insights and improve the overall analysis outcomes.

While this thesis represents a small step towards addressing these questions, further research is required to fully answer them and advance the integration of marginal and relational analysis in the field of gene expression data analysis. Except for considering the traditional statistical approaches, modern machine learning,

and even deep learning approaches should be given more attention. Those modern techniques have shown amazing performances among many scientific domains or real-world applications (Zhan et al., 2021, 2022; Ma, 2022; Ma et al., 2020; Chen and Ahn, 2020). Particularly, the graph neural network has made surprising progress in relational data modeling (Dong et al., 2021; Chen et al., 2022; Duan et al., 2022; Dong et al., 2023); the higher-order modeling idea has shown effectiveness in the mechanical system science as well (Wu and Du, 2020; Wu et al., 2021; Wu and Du, 2020).

Bibliography

Allen, J. D., Xie, Y., Chen, M., Girard, L., and Xiao, G. (2012). Comparing statistical methods for constructing large scale gene networks. *PloS one*, 7(1):e29348.

Andrews, T. S. and Hemberg, M. (2018). False signals induced by single-cell imputation. *F1000Research*, 7.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29.

Bach, F. R. and Jordan, M. I. (2002). Kernel independent component analysis. *Journal of machine learning research*, 3(Jul):1–48.

Bairamov, I., Kotz, S., and Kozubowski, T. (2003). A new measure of linear local dependence. *Statistics: A Journal of Theoretical and Applied Statistics*, 37(3):243–258.

Balakrishnan, S., Puniyani, K., and Lafferty, J. D. (2012). Sparse additive functional and kernel cca. In *ICML*.

Baran, Y., Bercovich, A., Sebe-Pedros, A., Lubling, Y., Giladi, A., Chomsky, E., Meir, Z., Hoichman, M., Lifshitz, A., and Tanay, A. (2019). Metacell: analysis of single-cell rna-seq data using k-nn graph partitions. *Genome biology*, 20(1):1–19.

Bell, C. (1962). Mutual information and maximal correlation as measures of dependence. *The Annals of Mathematical Statistics*, pages 587–595.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

Bergsma, W., Dassios, A., et al. (2014). A consistent test of independence based on a sign covariance related to kendall's tau. *Bernoulli*, 20(2):1006–1028.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society: Series B (Methodological)*, 48(3):259–279.

Bjerve, S. and Doksum, K. (1993). Correlation curves: measures of association as functions of covariate values. *The Annals of Statistics*, pages 890–902.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Blum, J. R., Kiefer, J., and Rosenblatt, M. (1961). Distribution free tests of independence based on the sample distribution function. *The annals of mathematical statistics*, pages 485–498.

Blyth, S. (1994a). Karl pearson and the correlation curve. *International Statistical Review/Revue Internationale de Statistique*, pages 393–403.

Blyth, S. J. (1994b). Measuring local association: an introduction to the correlation curve. *Sociological Methodology*, pages 171–197.

Cao, J., Qi, X., and Zhao, H. (2012). Modeling gene regulation networks using ordinary differential equations. *Next generation microarray bioinformatics: methods and protocols*, pages 185–197.

Cao, Z.-J. and Gao, G. (2022). Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. *Nature Biotechnology*, pages 1–9.

Chang, B., Kruger, U., Kustra, R., and Zhang, J. (2013). Canonical correlation analysis based on hilbert-schmidt independence criterion and centered kernel target alignment. In *International Conference on Machine Learning*, pages 316–324. PMLR.

Chen, C., Wu, C., Wu, L., Wang, X., Deng, M., and Xi, R. (2020). scrmd: Imputation for single cell rna-seq data via robust matrix decomposition. *Bioinformatics*, 36(10):3156–3161.

Chen, T., Zhou, K., Duan, K., Zheng, W., Wang, P., Hu, X., and Wang, Z. (2022). Bag of tricks for training deeper graph neural networks: A comprehensive benchmark study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2769–2781.

Chen, Z. and Ahn, H. (2020). Item response theory based ensemble in machine learning. *International Journal of Automation and Computing*, 17:621–636.

Chen, Z., Chen, C., Zheng, Z., and Zhu, Y. (2019). Tensor decomposition for multilayer networks clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3371–3378.

Cheng, J., Li, T., Levina, E., and Zhu, J. (2017). High-dimensional mixed graphical models. *Journal of Computational and Graphical Statistics*, 26(2):367–378.

Chu, L.-F., Leng, N., Zhang, J., Hou, Z., Mamott, D., Vereide, D. T., Choi, J., Kendziorski, C., Stewart, R., and Thomson, J. A. (2016). Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome biology*, 17(1):1–20.

Chunaev, P. (2020). Community detection in node-attributed social networks: a survey. *Computer Science Review*, 37:100286.

Dai, H., Li, L., Zeng, T., and Chen, L. (2019). Cell-specific network constructed by single-cell rna sequencing data. *Nucleic acids research*, 47(11):e62–e62.

Daub, C. O., Steuer, R., Selbig, J., and Kloska, S. (2004). Estimating mutual information using b-spline functions–an improved similarity measure for analysing gene expression data. *BMC bioinformatics*, 5(1):1–12.

De La Fuente, A., Bing, N., Hoeschele, I., and Mendes, P. (2004). Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18):3565–3574.

Delgado, F. M. and Gómez-Vela, F. (2019). Computational methods for gene regulatory networks reconstruction and analysis: A review. *Artificial intelligence in medicine*, 95:133–145.

Dhar, S. S., Dassios, A., Bergsma, W., et al. (2016). A study of the power and robustness of a new test for independence against contiguous alternatives. *Electronic Journal of Statistics*, 10(1):330–351.

Dibaeinia, P. and Sinha, S. (2020a). Sergio: a single-cell expression simulator guided by gene regulatory networks. *Cell systems*, 11(3):252–271.

Dibaeinia, P. and Sinha, S. (2020b). Sergio: A single-cell expression simulator guided by gene regulatory networks. *Cell Systems*, 11(3):252–271.

Doksum, K., Blyth, S., Bradlow, E., Meng, X.-L., and Zhao, H. (1994). Correlation curves as local measures of variance explained by regression. *Journal of the American Statistical Association*, 89(426):571–582.

Dong, G., Boukhechba, M., Shaffer, K. M., Ritterband, L. M., Gioeli, D. G., Reilley, M. J., Le, T. M., Kunk, P. R., Bauer, T. W., and Chow, P. I. (2021). Using graph representation learning to predict salivary cortisol levels in pancreatic cancer patients. *Journal of Healthcare Informatics Research*, 5:401–419.

Dong, G., Tang, M., Wang, Z., Gao, J., Guo, S., Cai, L., Gutierrez, R., Campbel, B., Barnes, L. E., and Boukhechba, M. (2023). Graph neural networks in iot: a survey. *ACM Transactions on Sensor Networks*, 19(2):1–50.

Duan, K., Liu, Z., Wang, P., Zheng, W., Zhou, K., Chen, T., Hu, X., and Wang, Z. (2022). A comprehensive study on large-scale graph training: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 35:5376–5389.

Efron, B. (2004). Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *Journal of the American Statistical Association*, 99(465):96–104.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868.

Emmert-Streib, F., Dehmer, M., and Haibe-Kains, B. (2014). Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. *Front Cell Dev Biol*, 2:38.

Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S., and Theis, F. J. (2019). Single-cell rna-seq denoising using a deep count autoencoder. *Nature communications*, 10(1):390.

Fischer, D. S., Schaar, A. C., and Theis, F. J. (2021). Learning cell communication from spatial graphs of cells. *BioRxiv*.

Fortunato, S. and Barthelemy, M. (2007). Resolution limit in community detection. *Proceedings of the national academy of sciences*, 104(1):36–41.

Fu, J. M., Satterstrom, F. K., Peng, M., Brand, H., Collins, R. L., Dong, S., Wamsley, B., Klei, L., Wang, L., Hao, S. P., et al. (2022). Rare coding variation provides insight into the genetic architecture and phenotypic context of autism. *Nature genetics*, 54(9):1320–1331.

Gandal, M. J., Haney, J. R., Wamsley, B., Yap, C. X., Parhami, S., Emani, P. S., Chang, N., Chen, G. T., Hoftman, G. D., de Alba, D., Ramaswami, G., Hartl, C. L., Bhattacharya, A., Luo, C., Jin, T., Wang, D., Kawaguchi, R., Quintero, D., Ou, J., Wu, Y. E., Parikshak, N. N., Swarup, V., Belgard, T. G., Gerstein, M., Pasaniuc, B., and Geschwind, D. H. (2022). Broad transcriptomic dysregulation occurs across the cerebral cortex in asd. *Nature*, 611(7936):532–539.

Gao, C., Ma, Z., Ren, Z., and Zhou, H. H. (2015). Minimax estimation in sparse canonical correlation analysis. *The Annals of Statistics*, pages 2168–2197.

GAO, C., MA, Z., and ZHOU, H. H. (2017). Sparse cca: Adaptive estimation and computational barriers. *The Annals of Statistics*, 45(5):2074–2101.

Gebelein, H. (1941). Das statistische problem der korrelation als variations-und eigenwertproblem und sein zusammenhang mit der ausgleichsrechnung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 21(6):364–379.

Ghazanfar, S., Lin, Y., Su, X., Lin, D. M., Patrick, E., Han, Z.-G., Marioni, J. C., and Yang, J. Y. H. (2020). Investigating higher-order interactions in single-cell data with schot. *Nature Methods*, 17(8):799–806.

Giné, E. and Guillou, A. (2002). Rates of strong uniform consistency for multivariate kernel density estimators. In *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, volume 38, pages 907–921. Elsevier.

Gong, W., Kwak, I.-Y., Pota, P., Koyano-Nakagawa, N., and Garry, D. J. (2018). Drimpute: imputing dropout events in single cell rna sequencing data. *BMC bioinformatics*, 19(1):220.

Gorfine, M., Heller, R., and Heller, Y. (2012). Comment on detecting novel associations in large data sets. *Science*, pages 1–6.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer.

Grün, D., Kester, L., and Van Oudenaarden, A. (2014). Validation of noise models for single-cell transcriptomics. *Nature methods*, 11(6):637.

Haque, A., Engel, J., Teichmann, S. A., and Lönnberg, T. (2017). A practical guide to single-cell rna-sequencing for biomedical research and clinical applications. *Genome Med*, 9(1):75.

He, X., Sanders, S. J., Liu, L., De Rubeis, S., Lim, E. T., Sutcliffe, J. S., Schellenberg, G. D., Gibbs, R. A., Daly, M. J., Buxbaum, J. D., et al. (2013). Integrated model of de novo and inherited genetic variants yields greater power to identify risk genes. *PLoS genetics*, 9(8):e1003671.

Heller, R., Heller, Y., and Gorfine, M. (2013). A consistent multivariate test of association based on ranks of distances. *Biometrika*, 100(2):503–510.

Heller, R., Heller, Y., Kaufman, S., Brill, B., and Gorfine, M. (2016). Consistent distribution-free k-sample and independence tests for univariate random variables. *The Journal of Machine Learning Research*, 17(1):978–1031.

Hoeffding, W. (1948). A non-parametric test of independence. *The annals of mathematical statistics*, pages 546–557.

Holland, P. W. and Wang, Y. J. (1987). Dependence function for continuous bivariate densities. *Communications in Statistics-Theory and Methods*, 16(3):863–876.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(3/4):321–377.

Huang, M., Wang, J., Torre, E., Dueck, H., Shaffer, S., Bonasio, R., Murray, J. I., Raj, A., Li, M., and Zhang, N. R. (2018). Saver: gene expression recovery for single-cell rna sequencing. *Nature methods*, 15(7):539.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.

Huynh-Thu, V. A., Irrthum, A., Wehenkel, L., and Geurts, P. (2010). Inferring regulatory networks from expression data using tree-based methods. *PloS one*, 5(9):e12776.

Iacono, G., Massoni-Badosa, R., and Heyn, H. (2019). Single-cell transcriptomics unveils gene regulatory network plasticity. *Genome biology*, 20(1):110.

Inouye, D. I., Yang, E., Allen, G. I., and Ravikumar, P. (2017). A review of multivariate distributions for count data derived from the poisson distribution. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(3):e1398.

Janson, S. (2004). Large deviations for sums of partly dependent random variables. *Random Structures & Algorithms*, 24(3):234–248.

Jones, M. C. (1998). Constant local dependence. *Journal of Multivariate Analysis*, 64(2):148–155.

Karlebach, G. and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature reviews Molecular cell biology*, 9(10):770–780.

Kim, H. J., Wang, K., Chen, C., Lin, Y., Tam, P. P. L., Lin, D. M., Yang, J. Y. H., and Yang, P. (2021). Uncovering cell identity through differential stability with cepo. *Nature Computational Science*, 1(12):784–790.

Kim, J. K. and Marioni, J. C. (2013). Inferring the kinetics of stochastic gene expression from single-cell rna-sequencing data. *Genome biology*, 14(1):R7.

Kim, K., Jiang, K., Teng, S. L., Feldman, L. J., and Huang, H. (2012). Using biologically interrelated experiments to identify pathway genes in arabidopsis. *Bioinformatics*, 28(6):815–822.

Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., Natarajan, K. N., Reik, W., Barahona, M., Green, A. R., et al. (2017). Sc3: consensus clustering of single-cell rna-seq data. *Nature methods*, 14(5):483–486.

Kriege, N. M., Johansson, F. D., and Morris, C. (2020). A survey on graph kernels. *Applied Network Science*, 5(1):1–42.

Langfelder, P. and Horvath, S. (2008). Wgcna: an r package for weighted correlation network analysis. *BMC bioinformatics*, 9(1):1–13.

Li, J. and Maathuis, M. H. (2021). Ggm knockoff filter: False discovery rate control for gaussian graphical models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(3):534–558.

Li, W. V. and Li, J. J. (2018). An accurate and robust imputation method scimpute for single-cell rna-seq data. *Nature communications*, 9(1):997.

Li, Z., Mccormick, T., and Clark, S. (2019). Bayesian joint spike-and-slab graphical lasso. In *International Conference on Machine Learning*, pages 3877–3885. PMLR.

Lin, K. Z., Liu, H., and Roeder, K. (2021). Covariance-based sample selection for heterogeneous data: Applications to gene expression and autism risk gene detection. *Journal of the American Statistical Association*, 116(533):54–67.

Lin, Y., Ghazanfar, S., Wang, K. Y., Gagnon-Bartsch, J. A., Lo, K. K., Su, X., Han, Z.-G., Ormerod, J. T., Speed, T. P., Yang, P., et al. (2019). scmerge leverages factor analysis, stable expression, and pseudoreplication to merge multiple single-cell rna-seq datasets. *Proceedings of the National Academy of Sciences*, 116(20):9775–9784.

Linderman, G. C., Zhao, J., and Kluger, Y. (2018). Zero-preserving imputation of scrna-seq data using low-rank approximation. *bioRxiv*, page 397588.

Liu, H., Han, F., Yuan, M., Lafferty, J., Wasserman, L., et al. (2012). High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326.

Liu, L., Lei, J., and Roeder, K. (2015). Network assisted analysis to reveal the genetic basis of autism. *The annals of applied statistics*, 9(3):1571.

Liu, L., Lei, J., Sanders, S. J., Willsey, A. J., Kou, Y., Cicek, A. E., Klei, L., Lu, C., He, X., Li, M., et al. (2014). Dawn: a framework to identify autism genes and subnetworks using gene expression and genetics. *Molecular autism*, 5:1–18.

Lun, A. T., Bach, K., and Marioni, J. C. (2016). Pooling across cells to normalize single-cell rna sequencing data with many zero counts. *Genome biology*, 17(1):75.

Ma, X. (2022). *Traffic Performance Evaluation Using Statistical and Machine Learning Methods*. PhD thesis, The University of Arizona.

Ma, X., Karimpour, A., and Wu, Y.-J. (2020). Statistical evaluation of data requirement for ramp metering performance assessment. *Transportation Research Part A: Policy and Practice*, 141:248–261.

Manicka, S., Johnson, K., Levin, M., and Murrugarra, D. (2023). The nonlinearity of regulation in biological networks. *NPJ Systems Biology and Applications*, 9(1):10.

McCarthy, D. J., Chen, Y., and Smyth, G. K. (2012). Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. *Nucleic acids research*, 40(10):4288–4297.

Mercatelli, D., Scalambra, L., Triboli, L., Ray, F., and Giorgi, F. M. (2020). Gene regulatory network inference resources: A practical overview. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, 1863(6):194430.

Mohamed, E.-M., Agouti, T., Tikniouine, A., and El Adnani, M. (2019). A comprehensive literature review on community detection: Approaches and applications. *Procedia Computer Science*, 151:295–302.

Neale, B. M., Kou, Y., Liu, L., Ma'Ayan, A., Samocha, K. E., Sabo, A., Lin, C.-F., Stevens, C., Wang, L.-S., Makarov, V., et al. (2012). Patterns and rates of exonic de novo mutations in autism spectrum disorders. *Nature*, 485(7397):242–245.

Nguyen, H., Tran, D., Tran, B., Pehlivan, B., and Nguyen, T. (2021). A comprehensive survey of regulatory network inference methods using single cell rna sequencing data. *Briefings in bioinformatics*, 22(3):bbaa190.

Pang, K., Wang, L., Wang, W., Zhou, J., Cheng, C., Han, K., Zoghbi, H. Y., and Liu, Z. (2020). Coexpression enrichment analysis at the single-cell level reveals convergent defects in neural progenitor cells and their cell-type transitions in neurodevelopmental disorders. *Genome Res*, 30(6):835–848.

Papadopoulos, N., Gonzalo, P. R., and Söding, J. (2019). Prosstt: probabilistic simulation of single-cell rna-seq data for complex differentiation processes. *Bioinformatics*, 35(18):3517–3519.

Parikshak, N. N., Luo, R., Zhang, A., Won, H., Lowe, J. K., Chandran, V., Horvath, S., and Geschwind, D. H. (2013). Integrative functional genomic analyses implicate specific molecular pathways and circuits in autism. *Cell*, 155(5):1008–21.

Parikshak, N. N., Swarup, V., Belgard, T. G., Irimia, M., Ramaswami, G., Gandal, M. J., Hartl, C., Leppa, V., de la Torre Ubieta, L., Huang, J., et al. (2016). Genome-wide changes in lncrna, splicing, and regional gene expression patterns in autism. *Nature*, 540(7633):423.

Peng, M., Wamsley, B., Elkins, A. G., Geschwind, D. H., Wei, Y., and Roeder, K. (2021). Cell type hierarchy reconstruction via reconciliation of multi-resolution cluster tree. *Nucleic acids research*, 49(16):e91–e91.

Polioudakis, D., de la Torre-Ubieta, L., Langerman, J., Elkins, A. G., Shi, X., Stein, J. L., Vuong, C. K., Nichterwitz, S., Gevorgian, M., Opland, C. K., Lu, D., Connell, W., Ruzzo, E. K., Lowe, J. K., Hadzic, T., Hinz, F. I., Sabri, S., Lowry, W. E., Gerstein, M. B., Plath, K., and Geschwind, D. H. (2019). A single-cell transcriptomic atlas of human neocortical development during mid-gestation. *Neuron*, 103(5):785–801.e8.

Pratapa, A., Jalihal, A. P., Law, J. N., Bharadwaj, A., and Murali, T. (2020). Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature Methods*, 17(2):147–154.

Quinn, T. P., Erb, I., Richardson, M. F., and Crowley, T. M. (2018). Understanding sequencing data as compositions: an outlook and review. *Bioinformatics*, 34(16):2870–2878.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Raj, A., Peskin, C. S., Tranchina, D., Vargas, D. Y., and Tyagi, S. (2006). Stochastic mrna synthesis in mammalian cells. *PLoS biology*, 4(10).

Rényi, A. (1959). On measures of dependence. *Acta Mathematica Academiae Scientiarum Hungarica*, 10(3-4):441–451.

Reshef, D. N., Reshef, Y. A., Finucane, H. K., Grossman, S. R., McVean, G., Turnbaugh, P. J., Lander, E. S., Mitzenmacher, M., and Sabeti, P. C. (2011). Detecting novel associations in large data sets. *science*, 334(6062):1518–1524.

Ríos, O., Frias, S., Rodríguez, A., Kofman, S., Merchant, H., Torres, L., and Mendoza, L. (2015). A boolean network model of human gonadal sex determination. *Theoretical Biology and Medical Modelling*, 12(1):26.

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

Sedgewick, A. J., Shi, I., Donovan, R. M., and Benos, P. V. (2016). Learning mixed graphical models with separate sparsity parameters and stability-based model selection. *BMC bioinformatics*, 17:307–318.

Shahrezaei, V. and Swain, P. S. (2008). Analytical distributions for stochastic gene expression. *Proceedings of the National Academy of Sciences*, 105(45):17256–17261.

Shen, C., Priebe, C. E., and Vogelstein, J. T. (2020). From distance correlation to multiscale graph correlation. *Journal of the American Statistical Association*, 115(529):280–291.

Simon, N. and Tibshirani, R. (2014). Comment on" detecting novel associations in large data sets" by reshef et al, science dec 16, 2011. *arXiv preprint arXiv:1401.7645*.

Song, L., Langfelder, P., and Horvath, S. (2012). Comparison of co-expression measures: mutual information, correlation, and model based indices. *BMC bioinformatics*, 13(1):1–21.

Steuer, R., Kurths, J., Daub, C. O., Weise, J., and Selbig, J. (2002). The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics*, 18(suppl_2):S231–S240.

Street, K., Risso, D., Fletcher, R. B., Das, D., Ngai, J., Yosef, N., Purdom, E., and Dudoit, S. (2018). Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC genomics*, 19:1–16.

Su, X., Xue, S., Liu, F., Wu, J., Yang, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Jin, D., et al. (2022). A comprehensive survey on community detection with deep learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Székely, G. J., Rizzo, M. L., Bakirov, N. K., et al. (2007). Measuring and testing dependence by correlation of distances. *The annals of statistics*, 35(6):2769–2794.

Tian, J., Lei, J., and Roeder, K. (2022). From local to global gene co-expression estimation using single-cell rna-seq data. *arXiv preprint arXiv:2203.01990*.

Tian, J., Wang, J., and Roeder, K. (2021). Esco: single cell expression simulation incorporating gene co-expression. *Bioinformatics*, 37(16):2374–2381.

Tjøstheim, D. and Hufthammer, K. O. (2013). Local gaussian correlation: A new measure of dependence. *Journal of Econometrics*, 172(1):33–48.

Toh, H. and Horimoto, K. (2002). Inference of a genetic network by a combined approach of cluster analysis and graphical gaussian modeling. *Bioinformatics*, 18(2):287–297.

Traag, V. A., Waltman, L., and Van Eck, N. J. (2019). From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233.

Trinh, D. T. (2019). Volume of sublevel sets versus area of level sets via gelfand-leray form. *Acta Mathematica Vietnamica*, 44(4):915–922.

Tseng, P. (2001). Convergence of a block coordinate descent method for non-differentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494.

Uurtio, V., Bhadra, S., and Rousu, J. (2018). Sparse non-linear cca through hilbert-schmidt independence criterion. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1278–1283. IEEE.

Van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A. J., Burdziak, C., Moon, K. R., Chaffer, C. L., Pattabiraman, D., et al. (2018). Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729.

Velmeshev, D., Schirmer, L., Jung, D., Haeussler, M., Perez, Y., Mayer, S., Bhaduri, A., Goyal, N., Rowitch, D. H., and Kriegstein, A. R. (2019). Single-cell genomics identifies cell type–specific molecular changes in autism. *Science*, 364(6441):685–689.

Vowels, M. J., Camgoz, N. C., and Bowden, R. (2022). D'ya like dags? a survey on structure learning and causal discovery. *ACM Computing Surveys*, 55(4):1–36.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press.

Wang, D., Yu, Y., and Rinaldo, A. (2021a). Optimal change point detection and localization in sparse dynamic networks. *The Annals of Statistics*, 49(1):203–232.

Wang, W. and Zhou, Y.-H. (2021). Eigenvector-based sparse canonical correlation analysis: Fast computation for estimation of multiple canonical vectors. *Journal of Multivariate Analysis*, 185:104781.

Wang, X., Choi, D., and Roeder, K. (2021b). Constructing local cell sepcific networks from single cell data. *bioRxiv*.

Wang, X. and Leng, C. (2015). High Dimensional Ordinary Least Squares Projection for Screening Variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(3):589–611.

Wang, Y. R., Waterman, M. S., and Huang, H. (2014). Gene coexpression measures in large heterogeneous samples using count statistics. *Proceedings of the National Academy of Sciences*, 111(46):16371–16376.

Wang, Y. X. R., Jiang, K., Feldman, L. J., Bickel, P. J., and Huang, H. (2015). Inferring gene–gene interactions and functional modules using sparse canonical correlation analysis. *The Annals of Applied Statistics*, 9(1):300 – 323.

Willsey, A. J., Sanders, S. J., Li, M., Dong, S., Tebbenkamp, A. T., Muhle, R. A., Reilly, S. K., Lin, L., Fertuzinhos, S., Miller, J. A., et al. (2013a). Coexpression networks implicate human midfetal deep cortical projection neurons in the pathogenesis of autism. *Cell*, 155(5):997–1007.

Willsey, A. J., Sanders, S. J., Li, M., Dong, S., Tebbenkamp, A. T., Muhle, R. A., Reilly, S. K., Lin, L., Fertuzinhos, S., Miller, J. A., Murtha, M. T., Bichsel, C., Niu, W., Cotney, J., Ercan-Sencicek, A. G., Gockley, J., Gupta, A. R., Han, W., He, X., Hoffman, E. J., Klei, L., Lei, J., Liu, W., Liu, L., Lu, C., Xu, X., Zhu, Y., Mane, S. M., Lein, E. S., Wei, L., Noonan, J. P., Roeder, K., Devlin, B., Sestan, N., and State, M. W. (2013b). Coexpression networks implicate human midfetal deep cortical projection neurons in the pathogenesis of autism. *Cell*, 155(5):997–1007.

Wu, H. and Du, X. (2020). System reliability analysis with second-order saddlepoint approximation. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 6(4):041001.

Wu, H., Hu, Z., and Du, X. (2021). Time-dependent system reliability analysis with second-order reliability method. *Journal of Mechanical Design*, 143(3):031101.

Xie, Y., Jiang, W., Dong, W., Li, H., Jin, S. C., Brueckner, M., and Zhao, H. (2022). Network assisted analysis of de novo variants using protein-protein interaction information identified 46 candidate genes for congenital heart disease. *PLoS genetics*, 18(6):e1010252.

Yoshida, K., Yoshimoto, J., and Doya, K. (2017). Sparse kernel canonical correlation analysis for discovery of nonlinear interactions in high-dimensional data. *BMC bioinformatics*, 18(1):1–11.

Yu, G., Wang, L.-G., Han, Y., and He, Q.-Y. (2012). clusterprofiler: an r package for comparing biological themes among gene clusters. *Omics: a journal of integrative biology*, 16(5):284–287.

Yurko, R., Roeder, K., Devlin, B., and G'Sell, M. (2021). An approach to gene-based testing accounting for dependence of tests among nearby genes. *Briefings in Bioinformatics*, 22(6):bbab329.

Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell rna sequencing data. *Genome biology*, 18(1):174.

Zeisel, A., Muñoz-Manchado, A. B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., et al. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347(6226):1138–1142.

Zhan, C., Ghaderibaneh, M., Sahu, P., and Gupta, H. (2021). Deepmtl: Deep learning based multiple transmitter localization. In *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 41–50. IEEE.

Zhan, C., Ghaderibaneh, M., Sahu, P., and Gupta, H. (2022). Deepmtl pro: Deep learning based multiple transmitter localization and power estimation. *Pervasive and Mobile Computing*, 82:101582.

Zhang, L. and Zhang, S. (2018). Comparison of computational methods for imputing single-cell rna-sequencing data. *IEEE/ACM transactions on computational biology and bioinformatics*.

Zhang, M., Eichhorn, S. W., Zingg, B., Yao, Z., Cotter, K., Zeng, H., Dong, H., and Zhuang, X. (2021). Spatially resolved cell atlas of the mouse primary motor cortex by merfish. *Nature*, 598(7879):137–143.

Zhang, X., Xu, C., and Yosef, N. (2019a). Simulating multiple faceted variability in single cell rna sequencing. *Nature communications*, 10(1):1–16.

Zhang, X.-F., Ou-Yang, L., Yang, S., Zhao, X.-M., Hu, X., and Yan, H. (2019b). Enimpute: imputing dropout events in single-cell rna-sequencing data via ensemble learning. *Bioinformatics*, 35(22):4827–4829.

Zheng, Z., Shi, H., Li, Y., and Yuan, H. (2020). Uniform joint screening for ultra-high dimensional graphical models. *Journal of Multivariate Analysis*, 179:104645.

Zhou, J., Li, Y., Zheng, Z., and Li, D. (2022). Reproducible learning in large-scale graphical models. *Journal of Multivariate Analysis*, 189:104934.

Zhu, L., Lei, J., Devlin, B., and Roeder, K. (2017). Testing high-dimensional covariance matrices, with application to detecting schizophrenia risk genes. *The annals of applied statistics*, 11(3):1810.

A

PLoD: pairwise local distributional information

A.1 Introduction

Chapter 3 depicts the relationship between two random variables with one single metric. However, there is often more structure in how two random variables are associated. For example, it is common among finance analysts and econometricians that the dependency between financial objects becomes stronger as the market goes down and approaches one when the market crashes. In genetics, people Wang et al. (2021b); Ghazanfar et al. (2020) found that gene coexpression changes with cell development: some genes tend to be actively co-expressed with each other in the cell developmental phase while being independent when the cells hit puberty. Characterization of a sort of *local dependence* becomes more essential than a single scalar dependence measure.

A.2 Related work

A.2.1 Local dependence quantifier

The curve of correlation. Bjerve and Doksum (1993), Doksum et al. (1994) and Blyth (1994a,b) introduce and discuss a "correlation curve", which is defined as how the amount of variance explained by a regression curve. Suppose random variables X and Y have joint density function $f(x, y)$, then the correlation curve $c(x)$ is defined as

$$c^2(x) := \frac{\sigma_x^2 d\mu(x)^2}{\sigma_x^2 d\mu(x)^2 + \sigma^2(x)},$$

where $\mu(x) := \mathbb{E}[Y \mid X = x]$ and $\sigma^2(x) := \mathrm{Var}[Y \mid X = x]$, and $\sigma_x^2 = \mathrm{Var}[X]$. However, this definition is really a regression concept rather than association concept as it does not treat X and Y on an equal footing.

Local linear dependence. To address the asymmetric issue of correlation curve, (Bairamov et al., 2003) describe a symmetrized variant whose expected value is

approximately equal to the Pearson correlation coefficient. Specifically, let X and Y be random variables with marginal distribution functions and densities F_X, f_X and F_Y, f_Y, respectively, then the local linear dependency function is defined as

$$l(x,y) := \frac{\mathbb{E}\{(X - \mathbb{E}\{X|Y=y\})(Y - \mathbb{E}\{Y|X=x\})\}}{\sqrt{\mathbb{E}\{(X - \mathbb{E}\{X|Y=y\})^2\}}\sqrt{\mathbb{E}\{(Y - \mathbb{E}\{Y|X=y\})^2\}}}. \tag{A.1}$$

With simple algebra, the local linear dependency function can be rewritten as

$$l(x,y) = \frac{\rho + r_x(y)r_y(x)}{\sqrt{1 + r_x(y)^2\}}\sqrt{1 + r_y(x)^2}}, \tag{A.2}$$

where ρ is the Pearson correlation coefficient, and $r_x(y)$ and

$$r_y(x) = \frac{1}{\sigma_X}\left(\mathbb{E}\{X\} - \mathbb{E}\{X|Y=y\}\right)$$
$$r_y(x) = \frac{1}{\sigma_X}\left(\mathbb{E}\{X\} - \mathbb{E}\{X|Y=y\}\right),$$

and $\frac{1}{\sigma_X}\left(\mathbb{E}\{Y\} - \mathbb{E}\{Y|X=x\}\right)$ respectively, which can be seen as the normalized residuals.

Local Gaussian dependence. Tjøstheim and Hufthammer (2013) proposed to depict local dependence by fitting a series of bivariate Gaussian locally. Specifically, for a general bivariate density f for the variables (X_1, X_2), locally in a neighbourhood of each position $\boldsymbol{x} = (x_1, x_2)$, they fit a bivariate Gaussian density $\mathcal{N}_2(\boldsymbol{\mu}_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}})$, where $\boldsymbol{\mu}_{\boldsymbol{x}} := (\mu_1(\boldsymbol{x}), \mu_2(\boldsymbol{x}))$ is the local mean vector and $\Sigma_{\boldsymbol{x}} := (\sigma_{ij}(\boldsymbol{x}))$ is the local covarance matrix. Then the local correlation measure is defined as

$$\rho(\boldsymbol{x}) := \frac{\sigma_{12}(\boldsymbol{x})}{\sqrt{\sigma_{11}(\boldsymbol{x})\sigma_{22}(\boldsymbol{x})}}.$$

Local dependence function. The notion of cross-product ratio for discrete two-way contingency table is extended to the case of continuous bivariate densities, which results in the "local dependence function" (Holland and Wang, 1987) that measures the margin-free dependence between bivariate random variables:

$$\gamma(x,y) := \frac{\partial \log f(x,y)}{\partial x \partial y} = \frac{1}{f^2(x,y)}\Big(\frac{\partial f(x,y)}{\partial x \partial y} f(x,y) - \frac{\partial f(x,y)}{\partial x}\frac{\partial f(x,y)}{\partial y}\Big).$$

Bjerve and Doksum (1993) provides a different derivation of this definition from a weighted Pearson correlation, stating that using a special product kernel weighting mechanism, the weighted Pearson correlation around (x,y) approximates $\gamma(x,y)$ as the kernel bandwidth goes to zero.

Properties of $\gamma(x, y)$ include: 1) it is a strong measure of dependence; 2) it is invariant to monotone marginal transformation; 3) for a bivariate normal, it is constant everywhere and takes the value $\frac{\rho}{1-\rho^2}$ where ρ is the Pearson correlation coefficient, and later Jones (1998) identified a family of distributions satisfying the property of constant local dependence, a family he called the exponential family of the conditional distribution.

The local contingency table test. Given n bivariate samples $(x_1, y_1), \ldots, (x_n, y_n)$, denote

$$T(x,y) := \frac{f_{X,Y}(x,y) - f_X(x)f_Y(y)}{\sqrt{f_X(x)f_Y(y)}},\ \widehat{T}(x,y) := \frac{\widehat{f}_{X,Y}(x,y) - \widehat{f}_X(x)\widehat{f}_Y(y)}{\sqrt{\widehat{f}_X(x)\widehat{f}_Y(y)}},$$

where $\widehat{f}_{X,Y}, \widehat{f}_X, \widehat{f}_Y$ are some density estimator for $f_{X,Y}, f_X, f_Y$. And for simplicity we write $\widehat{T}_i := \widehat{T}(x_i, y_i)$. Then the empirical aLDG can be written as

$$\widehat{\mathrm{aLDG}}_t = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{\widehat{T}_i \geq t\right\}. \tag{A.3}$$

If using boxcar kernel density estimator with bandwidth h, and set $t = \frac{\Phi^{-1}(1-\alpha)}{h\sqrt{n}}$, then it is easy to derive that,

$$\mathbf{1}\left\{\widehat{T}_i \geq t\right\} \equiv \phi_\alpha(S_i),$$

which is the α level Pearson's chi square test for 2×2 contingency tables defined by partitioning the sample space based on whether $x \in [x_i - h, x_i + h]$, and $y \in [y_i - h, y_i + h]$.

Remark 3. Note that the local dependence function in (Holland and Wang, 1987) is derived from the odds ratio of local contingency table tests. Specifically, at position (x, y), consider the $m \times m$ contingency tables defined by slicing the sample space to $m \times m$ rectangulars with sides dx, dy, such that $[x, x + dx]$, and $[y, y + dy]$ is one cell among this large contingency table. Then as $dx, dy \to 0$, the odds ratio for cell $[x, x + dx] \times [y, y + dy]$ and $[x + dx, x + 2dx] \times [y + dy, y + 2dy]$

$$r(x,y) \approx \frac{f(x,y)dxdyf(x+dx,y+dy)dxdy}{f(x,y+dy)dxdyf(x+dx,y)dxdy} \tag{A.4}$$

$$= \frac{f(x,y)f(x+dx,y+dy)}{f(x+dx,y)f(x,y+dy)} \tag{A.5}$$

when $dx, dy \to 0$, the limitation

$$\lim_{dx\to 0, dy\to 0} \frac{\log r(x,y)}{dxdy} = \frac{\partial \log f(x,y)}{\partial x \partial y} := \gamma(x,y), \tag{A.6}$$

which is just the local dependence function. In conclusion, our aLDG induced local statistics $T(x, y)$ can be thought as a variant of the local dependence function $\gamma(x, y)$: both are based on a 2-way local contingency table test, but the former is using Pearson Chi-square statistics for 2×2 table, and the latter is using odds ratio statistics for $m \times m$ table.

A.2.2 Synthetic example

In the following, we investigate the behaviors of above mentioned local dependence quantifier using synthetic examples. Specifically, we color each sample point based on the value of the local dependence quantifier at its position. Particularly we compare our local contingency table test statistics $T(x, y)$ with several others: the local linear dependence $l(x, y)$, the local Gaussian correlation $\rho(x, y)$, the local dependence function $\gamma(x, y)$, and the local density ratio test statistics $r(x, y)$. In Figure A.1, we plot the landscape of different local dependence quantifiers for different bivariate distributions, and we spot that $T(x, y)$ tends to highlight the part that contributes the most to the global dependence such as the boundaries and shapes. In Figure A.2, we plot similar metrics for three-component Gaussian mixture. We spot that only $\gamma(x, y)$ can give a sensible measure of dependence in restricted regions.

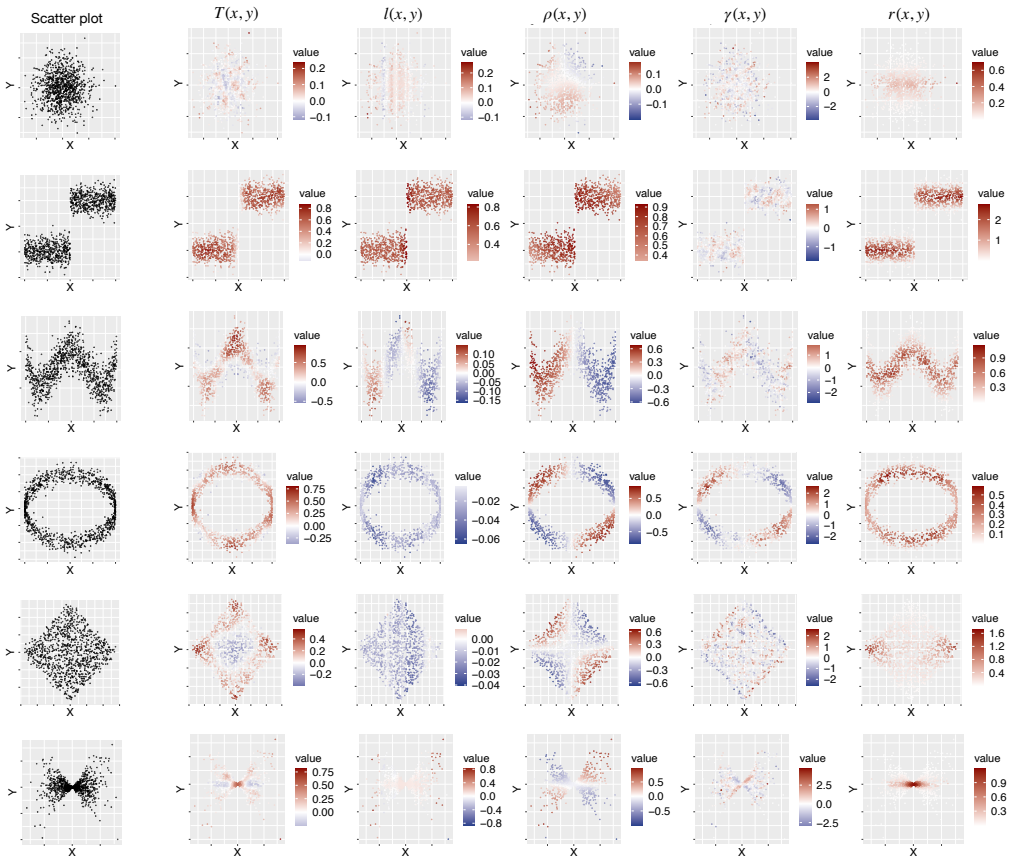


Figure A.1: Pattern given by different local dependence quantifier under different bivariate distribution.

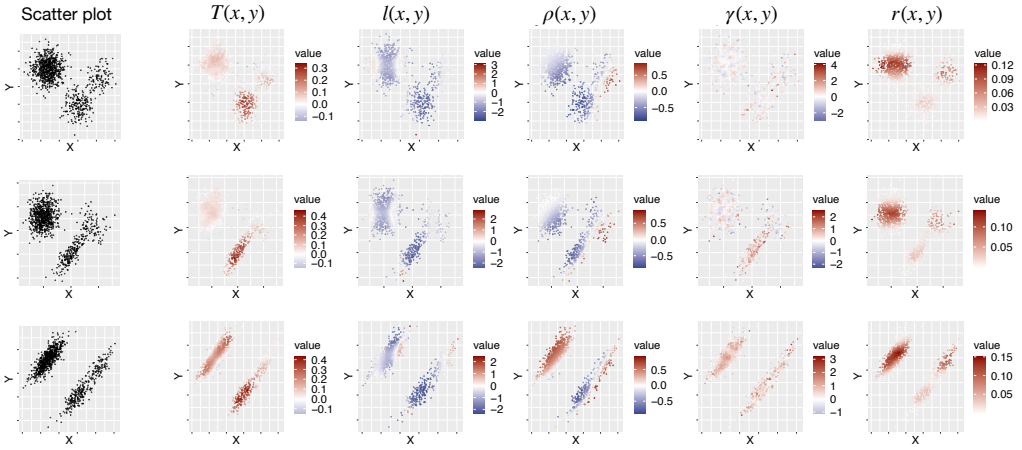


Figure A.2: Pattern given by different local dependence quantifier under Gaussian mixture.

A.3 Application of PLoD

As we discussed in Section A.2.1, for n bivariate samples $\mathcal{D} := \{(x_1, y_1), \ldots, (x_n, y_n)\}$, and $\widehat{f}_{X,Y}, \widehat{f}_X, \widehat{f}_Y$ are some density estimator for $f_{X,Y}, f_X, f_Y$ based on data $\mathcal{D}$, and the aLDG induced local statistics

$$\widehat{T}_{\mathcal{D}}(x, y) := \frac{\widehat{f}_{X,Y}(x, y) - \widehat{f}_X(x)\widehat{f}_Y(y)}{\sqrt{\widehat{f}_X(x)\widehat{f}_Y(y)}}, \tag{A.7}$$

can highlight subtle local dependence pattern. In multivariate case, where we observe data $\boldsymbol{\mathcal{D}} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ with $\boldsymbol{z}_k = (z_k^1, \ldots, z_k^p) \in \mathbb{R}^p$ i.i.d drawn from a p-dimensional distribution, we define the following data transformation motivated by the aLDG induced local statistics:

$$\psi : \mathbb{R}^{p\times n} \to \mathbb{R}^{p\times p\times n}, \quad \mathcal{D} \mapsto \mathcal{T}, \tag{A.8}$$

where $\mathcal{T}$ is a 3-way tensor with entry

$$T_{ijk} := \widehat{T}_{\{(z_1^i, z_1^j), \ldots, (z_n^i, z_n^j)\}}(z_k^i, z_k^j),$$

for $i, j \in [p], k \in [n]$. In words, we convert each p-dimensional observation vector to a $p \times p$ matrix that measures a local dependence level around this specific observation for each pair of features.

Transformation ψ can be defined more generally, with a different type of pairwise local measure T_{ijk} being plugged in, and the measure need not be about dependence: it can be any type of distributional information, i.e. the local measures mentioned

in Section A.2.1 can now being plugged in. We call this class of data transformation as Pairwise Local Distributional (PLoD) transformation, which is formally defined as the following:

Definition A.1. (**PLoD transformation**) Consider a bivariate local measure l, such that $l(\mathcal{D}; k)$ depicts some local distributional information at the k-th observation among all observations $\mathcal{D}$. For n observations of p-dimensional random vectors $\mathcal{M}$, we call the transformation

$$\psi : \mathbb{R}^{p\times n} \to \mathbb{R}^{p\times p\times n}, \quad \mathcal{M} \mapsto \mathcal{T} = \{T_{ijk}\}_{\substack{i,j\in[p],\\ k\in[n]}}, \quad T_{ijk} := l(\mathcal{M}_{\{i,j\}}; k) \tag{A.9}$$

as PLoD (*Pairwise Local Distributional*) transformation, where $\mathcal{M}_{\{i,j\}}$ stands for the subset of data that contains observations only for the i, j-th dimension of the random vector.

In the following, we investigate the application of this data transformation in various data tasks that admit a pattern of sparse mixed signals among enormous background noise. We show that, for this type of data, PLoD transformation captures subtle signals as it highlights local distributional information. In contrast, canonical global summary statistics cannot as the subtle signals are often averaged out with the enormous background noise.

A.3.1 A general model set-up

First we describe a general non-parametric mixture model, of which most of our theoretical analysis is a special case. Consider a random vector $Y = (Y_1, Y_2, \ldots, Y_p) \in \mathbb{R}^p$ with mixture density

$$g = \alpha_0 \otimes_{j\in[p]} h_0 + \sum_{m=1}^{M} \alpha_m f^m \otimes_{j\notin S_m} h_0, \tag{A.10}$$

where the first component is a p-dimensional density representing the pure background noise and the second component is a M-mixture density, with $S_m \subset [p]$, and f_m as $|S_m|$-dim density representing the low-dim signals for each mixture component. The proportion of group m is represented by $\alpha_m \in [0, 1]$, with $\sum_{m=0}^{M} \alpha_m = 1$.

We call S_m as the *signal dimensions* for the m-th mixture, and denote G_i as the group indexes that contains i as signals. We denote $S = \cup_{m\in[M]} S_m$ as the set of all signals, and $A_{ij} := \sum_{m\in G_i\cap G_j} \alpha_m$ as the total group proportion where dimensions i and j are both signals, and correspondingly $A_{i\setminus j} = \sum_{m\in G_i\setminus G_j} \alpha_m$, $A_{j\setminus i} :== \sum_{m\in G_j\setminus G_i} \alpha_m$, $A_{ij^c} := \sum_{m\in(G_i\cup G_j)^c} \alpha_m$. Note that $A_{ij} + A_{i\setminus j} + A_{j\setminus i} + A_{ij^c} = 1$.

A.3.2 Second-order population detection

In this section, we consider the task of clustering. The distributional differences in each cluster lie in the second-order information, e.g., covariance, rather than

the first-order information, e.g., mean. To give an example of such data structure, we first describe a data generation model in the following. We empirically show that canonical clustering methods like kMeans, Louvain, hierarchical clustering fail when directly applied to this kind of data, while they succeed after the PLoD transformation.

Problem description. We consider a special case of the general model in Section A.3.1, to make the dependence on covariance more clear. Specifically, we generate data for one cluster via the following Gaussian copula model:

$$\begin{aligned} Y_{ik} &= Q_i\left(\Phi^{-1}(X_{ik})\right) \quad \text{for } i = 1, 2, \ldots, p, \\ &\text{where } \ (X_{1k}, X_{2k}, \ldots, X_{pk}) \sim N(\mathbf{0}, \Sigma); \quad \text{for } k = 1, 2, \ldots, n, \end{aligned} \tag{A.11}$$

where Q_i is the quantile function of a complex distribution with parameters indexed by feature i, and Σ is a correlation matrix to impose dependency among features. We consider the clustering problem when two clusters are different only in Σ.

Given such data matrix $Y := \{Y_{ik}\}_{i\in[p],k\in[n]}$, applying PLoD transformation gives the three-way tensor $\mathcal{T} := \{T_{ijk}\}_{i,j\in[p],k\in[n]}$, and now the clustering task becomes inferring the group index for each sample using a $p \times p$ symmetric matrix. We call this symmetric matrix as *sample specific matrix* to set apart from original data matrix.

Tentative methods. Dai et al. (2019) propose to directly aggregate the sample specific matrix column wise, which results in a $p \times n$ dimensional matrix again. They call this matrix *network degree matrix* (NDM), for which they demonstrate better clustering results on various real data set. In the following simulation study, however, we show that this naive aggregation is not sufficient: it breaks down whenever the signal is sparse or overlapping. As for a concrete example of its weakness, note that, changes in any sub-modules in sample specific matrix of equal-sized 2-block structure $\begin{bmatrix} \mathbf{0} & \boldsymbol{A} \\ \boldsymbol{A} & \mathbf{0} \end{bmatrix}$ with $\boldsymbol{A} = \boldsymbol{A}^T$ to $\begin{bmatrix} \boldsymbol{A} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A} \end{bmatrix}$ in a symmetric matrix does not change its NDM.

Therefore we seek methods that preserve more information of the sample-specific matrix. We currently have explored the following three directions: **(1)** One simple but effective approach is vectorization: i.e. flatten the sample-specific matrix to one long vector of length $p(p-1)$. **(2)** One natural but more complex way is treating the sample-specific matrix as a weighted graph and utilizing graph clustering algorithms (e.g. Kriege et al. (2020)). **(3)** Moreover, when the signal is low-rank, methods based on tensor SVD may be more appropriate. Tensor SVD (e.g. Chen et al. (2019)) decomposes the noisy $\mathcal{T}$ to low-dimensional feature and sample loadings, and then matrix clustering algorithms can be applied on the sample loadings.

Synthetic experiments. With the motivation of application in single cell RNA-seq data, we simulate the data using a marginal distribution based on Negative Binomial, following the general modeling of single cell RNA-seq data in the literature (e.g. Tian et al. (2021)). And we simulate the correlation matrix $\Sigma \in \mathbb{R}^{p\times p}$ to follow the following block structure: given a set $S \subseteq [p]$,

$$\Sigma_{ij} = \Sigma_{ji} = \rho > 0, \quad \text{if } \{i,j\} \subset S; \qquad \Sigma_{ij} = \Sigma_{ji} = 0 \quad \text{otherwise}, \tag{A.12}$$

as it is often the case with gene interaction. We set $\rho = 0.99$ throughout the following experiments.[1] We simulate two cluster, each has $n = 200$, $p = 100$. We denote signal feature set for cluster 1 as S_1, and S_2 for cluster 2. We use the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) score to evaluate the performance of clustering[2]. In Figure A.5 we plot the boxplot of ARI from 20 runs using different methods under different setting of S_1 and S_2. We can see that clustering directly on data fails (`Data`), as well as the previous attempts (Dai et al., 2019) that aggregates the sample-specific matrix (`Avg.Tensor`), while our newly proposed ones do not. Particularly, the flattened tensor (`Flatten.Tensor`) and tensor SVD (`SVD.Tensor`) achieve near oracle performance, while the one based on graph clustering (`Graph.Tensor`) needs further adjustment. More supporting results are shown in Appendix A.4.

[1]With this strong correlation ($\rho = 0.99$), the model can also be seen as $Y = \begin{bmatrix} \boldsymbol{U}Z^{(1)} \\ Z^{(2)} \end{bmatrix}$, where $U \in \mathbb{R}^{|S|\times 1}$, $Z^{(1)} \in \mathbb{R}^{1\times n}$, $Z^{(2)} \in \mathbb{R}^{(p-|S|)\times n}$. Both $Z^{(1)}$ and $Z^{(2)}$ have independent rows.

[2]The ARI score lies in [0,1] where 0 corresponds to random assignment, and 1 corresponds to perfect assignment. A higher score implies better alignment of the estimated group labels to the true group labels.

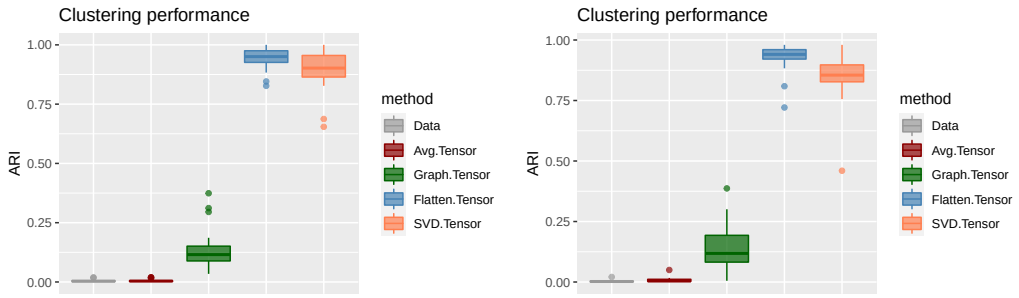


Figure A.3: The clustering performance over the 20 independent trials. The left plot shows results for non-overlapping communities: $S_1 = \{1, \ldots, 10\}; S_2 = \{11, \ldots, 20\}$; the right plot shows results for overlapping communities: $S_1 = \{1, \ldots, 30\}; S_2 = \{11, \ldots, 40\}$. Specifically, `Data` indicates clustering on original data matrix; and `Avg.Tensor` indicates clustering on the NDM of the tensor from the PLoD transformation; `Flatten.Tensor` indicates clustering on the vectorized tensor; `Graph.Tensor` indicates clustering using the graph view of sample specific matrix; `SVD.Tensor` indicates clustering on the sample loadings from tensor SVD.

Future work.

- We observe that (see Appendix A.4), when the covariance is weak, even PLoD induced methods failed. So one interesting direction is to find out when it fails, and whether we could circumvent it.
- We aim to find published data with signals that we can discover with our proposed techniques. To promote progress we need to scale the computations to handle realistic large scale data. We also wish to validate our findings, but this is difficult because at this time there are few, if any, gold standards in real data.

A.3.3 Feature selection

As pointed out by Kim et al. (2021) and many others, in single-cell RNA-seq data, stable gene expression is a key indicator of cell identity: genes marking a cell type should be (1) expressed and (2) stable in its expression level within this cell type, relative to other cell types. Regarding genes as features, cells as samples, the detection of such gene markers can be formulated as a special feature selection problem in a mixture model. Specifically, important features emerge only in subsets of samples, and the difference between important and unimportant features lies in the variance but not necessarily the mean. In other words, the shape of the data distribution matters and the more centered features are rendered more important.

Problem description. To make the problem statement more concrete, we consider the following data generative model, which is a specification of the general mixture

model described in Section A.3.1. Specifically, we consider a simple setting of only one signal group and independent features, that is

$$\boldsymbol{y} = (1-b)\boldsymbol{x}^{(0)} + b\boldsymbol{x}^{(1)} \in \mathbb{R}^p, \tag{A.13}$$

where the random variables

$$b \sim \text{Bernoulli}(\alpha), \quad \boldsymbol{x}^{(0)} \sim \otimes_p k_{0,1}, \quad \boldsymbol{x}^{(1)} \sim \otimes_{i\in S} k_{\mu,r} \otimes_{i\notin S} k_{0,1},$$

with $k_{\mu,r}(x) := \frac{1}{r}k\left(\frac{x-\mu}{r}\right)$, and k is a kernel smoothing function supported on $[-1,1]$, and S is the set of dimensions that we deem as a signal.

Remark 4. Note that under this model, the entries of $\boldsymbol{x}^{(0)}$ i.i.d follow $k_{0,1}$, and the first d entries of $\boldsymbol{x}^{(1)}$ i.i.d. follow $k_{\mu,r}$, and the rest $p-d$ entries i.i.d. follow $k_{0,1}$. Also note that, for random variable $\epsilon \sim k_{0,1}$, and $x \sim k_{\mu,r}$, we have ϵ and x are respectively σ_ϵ^2-sub-Gaussian and σ_x^2-sub-Gaussian random variables where,

$$\mathbb{E}[\epsilon] = 0, \quad \sigma_\epsilon^2 := \mathbb{E}\left[\epsilon^2\right] = \int_{[-1,1]} u^2 k_{0,1}(u)du; \tag{A.14}$$

$$\mathbb{E}[x] = \mu, \quad \sigma_x^2 := \mathbb{E}\left[x^2\right] = \int_{[-1,1]} (ru+\mu)^2 k_{0,1}(u)du = r^2\sigma_\epsilon^2 + \mu^2. \tag{A.15}$$

We are interested in the exact recovery of S in the high-dimensional setting, specifically when given only $n < p$ observations. For cleaner theoretical analysis, we analyze the ad-hoc version of all methods considered, that is, all the cutoffs (if there are any) are chosen as if the true size of S is known.

Tentative methods. Canonical methods to deal with signal recovery in this sparse, high dimensional setting are often based on global summary statistics, like the first-moment method based on empirical mean over all the samples or the second-moment method based on the empirical covariance matrix. We consider the following two methods to represent the global first moment and second-moment methods.

(1) First moment method: sparse Mean (sMean)

Consider using the empirical mean of all the features, and select feature dimensions with the top d empirical means as signals.

(2) Second moment method: sparse PCA (sPCA)

Consider using the leading eigenvector $\widehat{u}_1$ of the empirical covariance matrix of all the features, and select feature dimensions with the top d absolute values as signals.

As opposed to those *global* type of methods, we propose the following *local* type of method based on our PLoD transformation: We select the feature dimensions with

the top d PLoD scores as the signals, where the PLoD scores for each dimension are defined as

$$Z_i := \sum_{j=1}^{p} \sum_{k=1}^{n} \mathbf{1}_{\widehat{T}_{ijk} > t}, \quad \text{for all } i = 1, \ldots, p, \tag{A.16}$$

where t is some pre-specified threshold and $\widehat{T}_{ijk}$ is the (i, j, k)-th entry of the tensor from PLoD transformtion of the data. In this section we take $\widehat{T}_{ijk}$ to be the estimated gap between the joint and marginal product density of the ith and jth feature at the k-th sample, which works well for this task. Hence we call the resulted feature selection method as *Local Density Gap* (LDG) method.

Theoretical results. We identified sufficient conditions for sMean, sPCA, LDG to have exact recovery of S concerning the simple non-parametric model (A.13), under the mild assumption that the underlying marginal kernel density k is in Holder class $H(\beta, L)$ on $\mathbb{R}$. We found that, for global type of methods, sparse mean and sparse PCA, the sufficient condition requires the signal-to-noise ratio:

$$\text{SNR} := \frac{\mu}{\sigma_\epsilon} \gtrsim \frac{1}{\alpha} \sqrt{\frac{\log p}{n}}$$

for exact recovery of S. The detailed theorems are in Appendix A.5.

These results reveal the following scenario where sparse mean and sparse PCA fail to have exact recovery, but LDG still can: consider $\beta = 1$, the sufficient conditions for LDG to recover signals exactly become

$$1 \ll \frac{\alpha}{r^2} \lesssim \left(\frac{n}{\log p} \right)^{\frac{3}{2}}.$$

Note that, for $\alpha = \left(\frac{\log p}{n} \right)^x$ with $x > \frac{1}{2}$, it is impossible for sparse PCA to recover signals (if taken SNR as fixed); however, LDG can still recover signals perfectly when $\left(\frac{\log p}{n} \right)^{\frac{2x+3}{4}} \lesssim r \ll \left(\frac{\log p}{n} \right)^{\frac{x}{2}}$.

Synthetic experiments. We conducted the following simulations to verify the theoretical findings. We consider generating data following model (A.13), with $n = 100, p = 200, d = 20, \alpha = 0.1$, and study the influence of r, μ on the performance. In order to gain more consistent results for different density, we replace μ with $\text{SNR} := \frac{\mu}{\sigma_\epsilon^2}$. We demonstrate results in Figure A.4 for one simple case of marginal density, specifically, the compact non-smooth density, Boxcar: $k = \frac{1}{2}\mathbf{I}\{|x| \leq 1\}$. We can see that LDG has the highest power in most cases, regardless of SNR level. We observe similar success for other densities like Epanechnikov, Gaussian, Negative Binomial: we refer readers to Appendix A.5.1 for details.

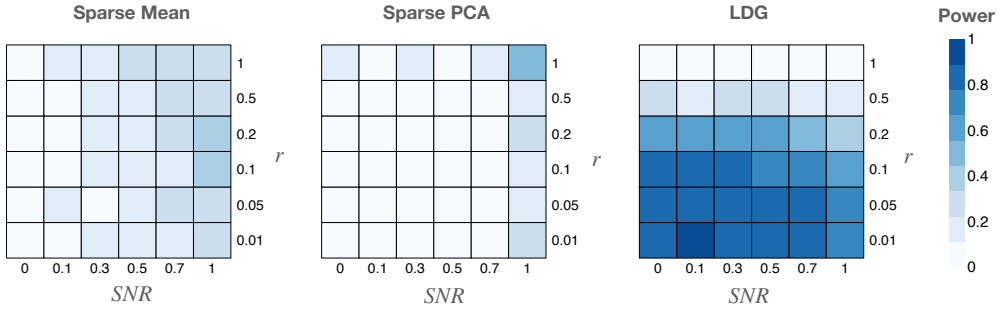


Figure A.4: The empirical power for recovering signal dimensions using simulated boxcar kernel densities in model (A.13) with $n = 100, p = 200, d = 20, \alpha = 0.1$, $\mu = \mathsf{SNR}\sigma_\epsilon$. The power is estimated via averaging over 10 trials.

Future work.

- We wish to improve performance of feature selection when the signal lies on a low-dimensional manifold (e.g. important features form complex dependence relationship).
- We wish to find real data application of this particular type of feature definition: that is the shape define the signal but not the location.

A.4 More synthetic experiments for subpopulation detection

We simulate the correlation matrix $\Sigma \in \mathbb{R}^{d\times d}$ for the signal components to follow the following block structure, as it is always the case with gene interaction: given sets $S^1, S^2, \ldots, S^b \subseteq [d]$, and $i \neq j$,

$$\begin{aligned} \Sigma_{ij} = \Sigma_{ji} = \rho > 0, \quad \text{if } \{i,j\} \subset S^m; \text{ for each } m \in [b]; \\ \Sigma_{ij} = \Sigma_{ji} = 0 \quad \text{otherwise.} \end{aligned} \tag{A.17}$$

Still we simulate two cluster, and denote the corresponding signal sets as $S_1^1, \ldots S_1^{b_1}$ and $S_2^1, \ldots S_2^{b_2}$.

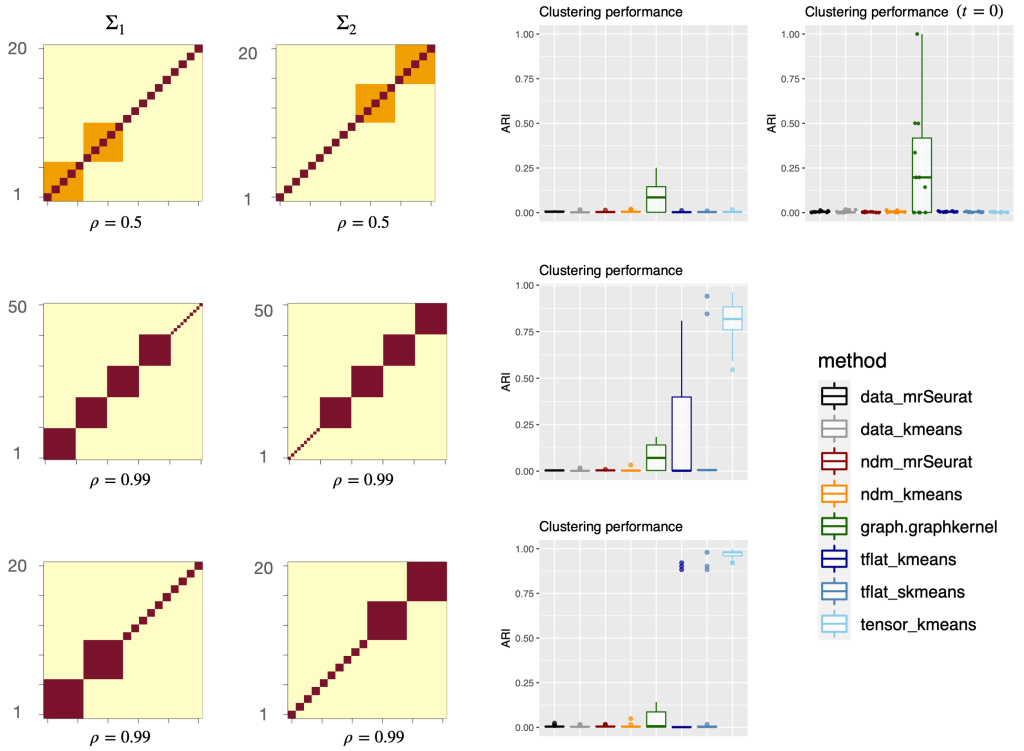


Figure A.5: The clustering performance over the 10 independent trials. For different row, the set up for the correlation matrix is different. Specifically, `data_` indicates clustering on original data matrix; and `ndm_` indicates clustering on the NDM of the tensor from the PLoD transformation; `tflat_` indicates clustering on the vectorized tensor; `graph_` indicates clustering using the graph view of sample specific matrix; `tensor_` indicates clustering on the sample loadings from tensor SVD.

A.5 Theoretical results on feature selection

For succinct, we only present technical results but omit the technical proofs.

Global type of methods.

Proposition 3. Consider n i.i.d observations $Y_1, \ldots, Y_n$ sampled from model (A.13). We have

(a) if $d, p, n \gg 1$, then sMean gives us

$$\min_{i \in S} \widehat{\mu}_i - \max_{j \notin S} \widehat{\mu}_j \gtrsim O\left(\mu - r\sigma_\epsilon \sqrt{\frac{\log d}{\alpha n}} - \sigma_\epsilon \sqrt{\frac{\log p}{(1-\alpha)n}}\right), \tag{A.18}$$

where $\widehat{\mu}_i$ is the i-th entry of $\widehat{\boldsymbol{\mu}} := \frac{1}{n}\sum_{l=1}^{n} Y_l$.

(b) if a eigen-gap condition is satisfied, i.e. $\mu^2 > \frac{1-r^2}{d}\sigma_\epsilon^2$, then sPCA gives us

$$||\boldsymbol{u}_1 - \widehat{\boldsymbol{u}}_1||_2^2 \lesssim O\left(\frac{\sqrt{\frac{\log p}{n}}}{\alpha(\frac{\mu^2}{\sigma_\epsilon^2} - \frac{1-r^2}{d})}\right), \tag{A.19}$$

where $\boldsymbol{u}_1$ and $\widehat{\boldsymbol{u}}_1$ is the leading eigenvector of the true and empirical covariance matrix over all the features respectively.

Proposition 3 indicates that the case is hard for sparse PCA, if signal-to-noise ratio μ/σ_ϵ is low; signal proportion α is low; signal radius r is small; and signal dimensions d is small. What is worth noting is that, the influence of r can be ignored, as long as the signal to noise ratio is bigger than one. However, if the signal to noise ratio is very small, then bigger r gonna makes things easier.

Our method. In the following we analysis the application of LDG statistics on signal recovery. We consider the following statistics based on bivariate density:

Definition A.2. For two random variables Y_i, Y_j in $\mathbb{R}$, we call the gap between their joint density and marginal density product as ***local density gap*** (LDG), that is

$$\Delta_{ij}^{\text{gap}} = g_{ij} - g_i g_j \tag{A.20}$$

where g_{ij}, g_i, g_j are the joint and marginal density of Y_i, Y_j respectively.

It is well known that Y_i and Y_j are independent, iff $\Delta_{ij}^{\text{gap}} \equiv 0$; however such argument does not hold true if Y_i and Y_j are only independent within each mixture component. In fact, Lemma A.3 states the relationship between LDG statistics with and without within-group independence, under general finite mixture bivariate model.

Lemma A.3. For any bi-variate mixture model of Y_i, Y_j with finite mixtures, we have

$$\begin{aligned}\Delta_{ij}^{\text{gap}}(y_i, y_j) =& \sum_m \alpha_m (c_{ij}^m(y_i, y_j) - \alpha_m) g_i^m(y_i) g_j^m(y_j) \\ &- \sum_{m_1 \neq m_2} \alpha_{m_1}\alpha_{m_2} g_i^{m_1}(y_i) g_j^{m_2}(y_j),\end{aligned} \tag{A.21}$$

where $c_m^{ij}(y_i, y_j) := \frac{g_{ij}^m(y_i, y_j)}{g_i(y_i) g_j(y_j)}$.

We consider quantifying the conditions that leads to well separation of the above statistics in the following cases. Without loss of generality, we consider $i < j$.

Denote $D_i^m = \{y : |y - \mu_i^m| \leq r\}$ for all $i \in S$ and $m \in G_i$, and $D = \{y : |y| \leq t\}$. Particularly, denote $D_i^0 = D \setminus \cup_{m \in G_i} D_i^m$, which is the region of pure noise for dimension i. We would like to find appropriate conditions on $\alpha_m, r, t, c_{ij}^m, \mu_i^m$ such that the above statistics can highlight region D_i^m for $i \in S_m$ inside truncated feasible region D.

Condition 1. For the above kernel mixture model formulation, we call it has **strong separability** if

(a) $t > \max_{i\in S, m\in G_i} |\mu_i^m| + 2r$,

(b) $r \ll 1$;

(c) $c_{ij}^m \in [\underline{C}, \overline{C}]$, for all i, j and $m \in G_i \cap G_j$, where constants

$$C_0^2 \Big/ C_1^2 \max_{i,j\in S} A_{ij} < \underline{C} \le 1 \le \overline{C}. \tag{A.22}$$

Lemma A.4. Denote $C_0 = K(0), C_1 = K(1)$. Considering the kernel mixture model with strong separability defined in Condition 1, we have that, for any $i \in S$, and $m \neq m' \in G_i$, for all $y \in D$, $f_i^m(y) \in [0, \frac{C_0}{r}]$, $h_0(y) \in [\frac{C_1}{t}, \frac{C_0}{t}]$, and

$$\begin{cases} f_i^m(y) \in [\frac{C_1}{r}, \frac{C_0}{r}], & \text{if } y \in D_i^m; \\ f_i^m(y) \equiv 0, & \text{if } y \notin D_i^m; \end{cases} \tag{A.23}$$

Lemma A.4 is easy to verify from the properties of K. From $C_0 \ge C_1 \ge 0$; Lemma A.4 describes a nice separation among marginal signal density and noise. In the following, we use Lemma A.4 constantly to obtain additional separation conditions.

Theorem A.5. Considering a kernel mixture model with strong separability. If $\frac{\min_{m\neq 0}\alpha_m}{r^2} \gg 1$, and $\frac{\max_{m\neq 0}\alpha_m}{\min_{m\neq 0}\alpha_m} < \infty$, then the following arguments holds true:

(a) if $\{i, j\} \subseteq S$, $y_i \notin D_i^0$ and $y_j \notin D_j^0$, then we have

$$\begin{cases} \Delta_{ij}^{\text{gap}}(y_i, y_j) \gtrsim \alpha_{m^\star}\Big(\underline{C} - \frac{C_0^2}{C_1^2}A_{ij}\Big)\frac{C_1^2}{r^2}, \\ \qquad \text{if } \exists m^\star \in G_i \cap G_j \text{ s.t. } (y_i, y_j) \in D_i^{m^\star} \otimes D_j^{m^\star}; \\ \Delta_{ij}^{\text{gap}}(y_i, y_j) \lesssim -\alpha_{m_1^\star}\alpha_{m_2^\star}\frac{C_1^2}{r^2}, \\ \qquad \text{otherwise, implies } \exists m_1^\star \neq m_2^\star \text{ s.t. } (y_i, y_j) \in D_i^{m_1^\star} \otimes D_j^{m_2^\star}. \end{cases} \tag{A.24}$$

(b) if $\{i, j\} \subseteq S$, and $(y_i, y_j) \in D_i^m \otimes D_j^0$, with $m \in G_i$, $\Big($or $(y_i, y_j) \in D_i^0 \otimes D_j^m$, with $m \in G_j$ $\Big)$, then

$$\Delta_{ij}^{\text{gap}}(y_i, y_j) \lesssim A_{i\setminus j}(1 - A_{i\setminus j})\frac{C_0^2}{rt}. \tag{A.25}$$

$\Big($or $\Delta_{ij}^{\text{gap}}(y_i, y_j) \lesssim A_{j\setminus i}(1 - A_{j\setminus i})\frac{C_0^2}{rt}$ $\Big)$.

(c) if $\{i,j\} \subseteq S$, and $(y_i, y_j) \in D_i^0 \otimes D_j^0$, then

$$\Delta_{ij}^{\text{gap}}(y_i, y_j) \leq A_{ij^c}(1 - A_{ij^c})\frac{C_0^2}{t^2}. \tag{A.26}$$

(d) if $\{i,j\} \not\subseteq S$, then for all (y_i, y_j),

$$\Delta_{ij}^{\text{gap}}(y_i, y_j) \equiv 0. \tag{A.27}$$

Theorem A.5 guarantees the nice property of LDG statistics: for $T \asymp \frac{\alpha}{r^2}$,

$$\Delta_{ij}^{\text{gap}} | m \geq T \text{ w.h.p} \Longleftrightarrow i,j \in S_m.$$

In the following we consider the one-dimensional kernel density k belongs to the Holder class $H(\beta, L)$ on $\mathbb{R}$, that is its β-th derivative is bounded by L. And we consider a kernel estimator of k with bandwidth h on $\mathbb{R}$ and $\mathbb{R}^2$, that is

$$\begin{aligned}
\widehat{K}_{h;i}(y_i) &:= \frac{1}{n}\sum_{t=1}^{n}\frac{1}{h}K\left(\frac{Y_{t;i} - y_i}{h}\right), \\
\widehat{K}_{h;ij}(y_i, y_j) &:= \frac{1}{n}\sum_{t=1}^{n}\frac{1}{h^2}K\left(\frac{Y_{t;i} - y_i}{h}\right)K\left(\frac{Y_{t;j} - y_j}{h}\right),
\end{aligned} \tag{A.28}$$

where the kernel estimator function K belongs to the class $G(\beta)$ on $\mathbb{R}$, that is K has support on $[-1,1]$, and $\int K = 1$, $\int |K|^p < \infty$ for any $p \geq 1$, $\int |t|^\beta K(t)dt < \infty$ and $\int t^s K(t)dt = 0$ for $s \leq \beta$.

Then the corresponding ad-hoc version of LDG method for signal recovery is the following.

Definition A.6. (LDG signal recovery) We select the feature dimensions with the top d LDG scores as the signals, where the LDG scores for each dimension is

$$Z_i := \sum_{j=1}^{p}\sum_{l=1}^{n}\mathbf{1}_{\widehat{\Delta}_{ij;l}^{\text{gap}} > T}, \quad \text{for all } i = 1, \ldots, p; \tag{A.29}$$

in which the LDG statistics is estimated as

$$\widehat{\Delta}_{ij;l} := \widehat{K}_{h;i}(Y_{l;i}) - \widehat{K}_{h;ij}(Y_{l;i}, Y_{l;j}), \tag{A.30}$$

and T is some pre-specified threshold.

One key step to quantify the accurateness boils down to control the estimation error in the kernel density estimation. Therefore we first present the following results with regard general kernel density estimation error under model (A.13).

Proposition 4. Consider 1-dimensional kernel smoothing function $K \in G(\beta)$, and 1-dimensional kernel density $k \in H(\beta, L)$. Denote $K_h(\cdot) = \frac{1}{h}K(\frac{\cdot}{h})$, $\mathbf{K}_h = \otimes^d K_h$, $\widehat{\boldsymbol{K}}_h(\cdot) = \frac{1}{n}\sum_{i=1}^n \boldsymbol{K}_h(X_i - \cdot)$; $k_{\alpha,\mu,r}(\cdot) = (1-\alpha)k(\cdot) + \alpha\frac{1}{r}k\left(\frac{\cdot-\mu}{r}\right)$, and $\mathbf{k}_{\alpha,\mu,r} = \otimes^d k_{\alpha,\mu,r}(\cdot)$. Then for any $\delta > 0$, we have

$$\Pr\left\{\sup_{\boldsymbol{x}\in\mathbb{R}^d} |\widehat{\boldsymbol{K}}_h(\boldsymbol{x}) - \boldsymbol{k}_{\alpha,\mu,r}(\boldsymbol{x})| > \sqrt{\frac{C\log(1/\delta)(1-\alpha+\frac{\alpha}{r})^d}{nh^d}} + c\left(1-\alpha+\frac{\alpha}{r^{\beta+1}}\right)^d h^{d\beta}\right\} < \delta, \tag{A.31}$$

where C and c are positive constants which do not depend on h, α, μ, r. Particularly, choosing adaptively

$$h = \left(\frac{C\log\frac{1}{\delta}(1-\alpha+\frac{\alpha}{r})^d}{c^2 n(1-\alpha+\frac{\alpha}{r^{\beta+1}})^{2d}}\right)^{\frac{1}{(2\beta+1)d}}, \tag{A.32}$$

we have

$$\Pr\left\{\sup_{\boldsymbol{x}\in\mathbb{R}^d} |\widehat{\mathbf{K}}_h(\boldsymbol{x}) - \mathbf{k}_{\mu,r}(\boldsymbol{x})| > 2c\left(\frac{C\log\frac{1}{\delta}}{c^2 n}\right)^{\frac{\beta}{2\beta+1}}\left(1-\alpha+\frac{\alpha}{r^{\beta+1}}\right)^{\frac{\beta+1}{2\beta+1}d}\right\} < \delta. \tag{A.33}$$

Now we are ready to introduce the results about using LDG method for signal recovery.

Theorem A.7. Consider the simple mixture model (A.13) with the underlying marginal kernel density $k \in H(\beta, L)$. If we use the 1-dimensional kernel smoothing function $K \in G(\beta)$ in density estimation, and assume

$$r^2 \vee \frac{1}{d}(\frac{p}{n} \vee 1) \ll \alpha \ll 1; \quad \frac{\alpha}{r^{(\beta^2+1)(2\beta+2)-2(2\beta+1)}} \lesssim \left(\frac{n}{\log p}\right)^{\frac{(2\beta+1)\beta}{\beta+1}}; \quad T \asymp \frac{\alpha}{r^2}. \tag{A.34}$$

Then for any $\delta > 0$, with probability at least $1-\delta$, we have

$$\sum_{j=1}^p \sum_{l=1}^n \mathbf{1}_{\widehat{\Delta}_{ij;l}^{\text{gap}} > T} \,\Big|\, i \in S \;\gtrsim\; \sum_{j=1}^p \sum_{l=1}^n \mathbf{1}_{\widehat{\Delta}_{ij;l}^{\text{gap}} > T} \,\Big|\, i \notin S. \tag{A.35}$$

The high level sketch of the proof follow the following four steps:

Step 1. Use Theorem A.5 to obtain asymptotic bounds on population level statistics Δ_{ij}^{gap}.

Step 2. Use Proposition 4 to obtain high probability bound on empirical level statistics $\widehat{\Delta}_{ij}^{\text{gap}}$.

Step 3. Use a graph dependency based variant of Bernstein inequality (Theorem 2.4. Janson (2004)) to obtain high probability bound on $\sum_{l=1}^{n} \mathbf{1}_{\widehat{\Delta}_{ij;l}^{\text{gap}}}$.

Step 4. Use a simple union bound over all feature dimensions to obtain high probability bound on the final accumulated degrees for each dimension $\sum_{j=1}^{p}\sum_{l=1}^{n} \mathbf{1}_{\widehat{\Delta}_{ij;l}^{\text{gap}}}$.

A.5.1 More synthetic experiments for feature selection

We consider another three marginal densities as follows:

(i) Compact smooth density, Epanechnikov : $k = \frac{3}{4}(1-x^2)\mathbf{I}\{|x| \le 1\}$;

(ii) Non-compact smooth density, Gaussian : $k = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$

(iii) Non-compact non-smooth density , single cell gene expression Tian et al. (2021):

$$
\begin{aligned}
Y_{ij} &\sim \text{Poisson}(\lambda_{ij}); \qquad\qquad (A.36)\\
\lambda_{ij} &= \text{Gamma}(1/\phi_{ij}, \lambda'_{ij}\phi_{ij}), \quad \phi_{ij} \sim \left(\phi + \frac{1}{\sqrt{\lambda'_{ij}}}\right)^2 \frac{1}{\text{Chiq}(df)}\\
\lambda'_{ij} &= L_j \frac{\lambda'_i}{\sum_i \lambda'_i}, \quad \lambda'_i \sim \text{Gamma}(\alpha, \beta); \quad L_j \sim \text{logNorm}(\mu_L, \sigma_L)
\end{aligned}
$$

In order to gain more consistent results for different density, we replace μ with $\text{snr} := \frac{\mu}{\sigma_\epsilon^2}$ for case (i-ii). For case (iii), as the data distribution is much more complex, hence the signal-to-noise ratio and signal radius is hard to quantitatively defined. Particularly we vary the mean parameter μ_L of library size L_j to control the overall mean and difference between the mean of noise group and cell group, and we vary the shape α (and rate β, under constrains $\alpha = \beta$ such that the mean is not influenced) parameters in the distribution of gene means λ'_g to control the difference between the variance of noise group and cell group. In fact we set and $\mu_L = 4$ and $\alpha = \beta = 0.1$ for noise group; $\mu_L = 4 + \mu$ with $\mu > 0$, and $\alpha = \beta = \frac{1}{10r}$ for the signal group, such that the ratio between signal variance and noise ratio is r.

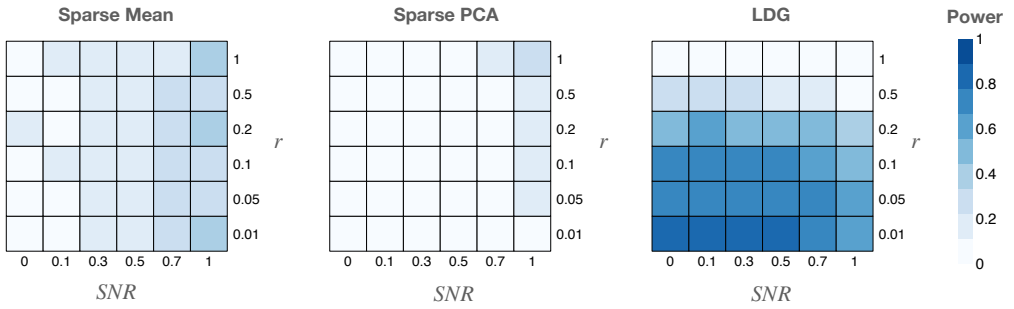


Figure A.6: The empirical power for recovering signal dimensions using simulated Epanechnikov densities with $n = 100, p = 200, d = 20, \alpha = 0.1$, $\mu = \text{SNR}\sigma_\epsilon$. The power is estimated via averaging over 10 trials.

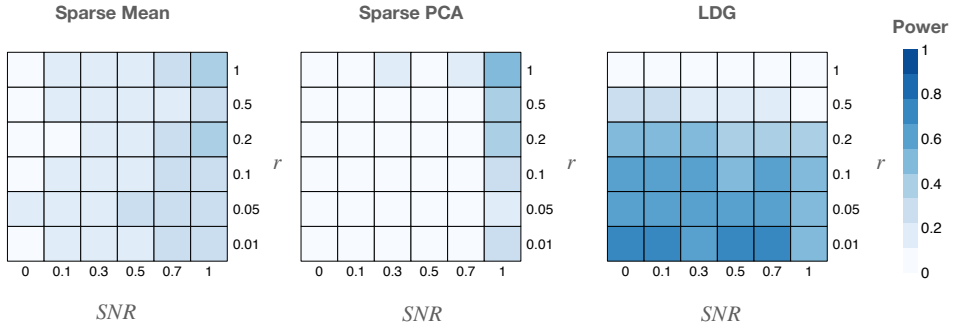


Figure A.7: The empirical power for recovering signal dimensions using simulated Gaussian densities with $n = 100, p = 200, d = 20, \alpha = 0.1$, $\mu = \text{SNR}\sigma_\epsilon$. The power is estimated via averaging over 10 trials.

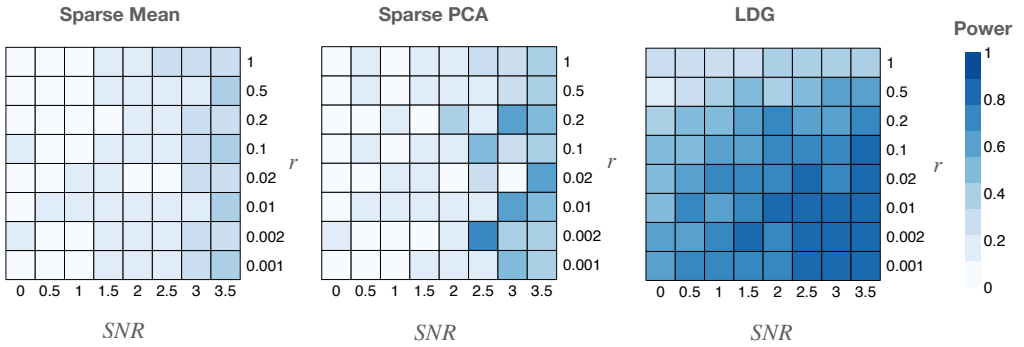


Figure A.8: The empirical power for recovering signal dimensions using simulated single cell gene expression densities in model (A.13) with $n = 500, p = 1000, d = 100, \alpha = 0.1$. The power is estimated via averaging over 10 trials.

B

Regional partial gene network estimation

Recall that in PNS, for gene i, we estimate its edges with other genes via running lasso regression. For simplicity of demonstration, let's drop the penalization for now, and just consider the following linear model. Previously we estimate the gene network for the whole cortex using the following model:

$$\text{(whole cortex model)} \quad x_i = \boldsymbol{x}_{\mathcal{N}(i)}^\top \boldsymbol{\beta},\ \boldsymbol{\beta} \in \mathbb{R}^{p\times 1} \tag{B.1}$$

Now take the regional differences into consideration, we will instead assume

$$\text{(regional model)} \quad x_i = \boldsymbol{x}_{\mathcal{N}(i)}^\top \boldsymbol{\beta}^{(r)},\ \boldsymbol{\beta}^{(r)} \in \mathbb{R}^{p\times 1},\ \text{ for region } r \tag{B.2}$$

where $\mathcal{N}(i)$ is the selected partial neighbourhood for gene i, and p is the size of $\mathcal{N}(i)$.

Key assumption. Assuming that neighbour gene j has $\beta_j^{(r)} \equiv \beta_j$ for the most of the genes $\mathcal{N}(i)$. That is, the regional effect is sparse. Under our assumption, it makes sense to bring together samples from different regions to estimate the network, such that they can borrow strength from each other when estimating parameters for the majority common part. However, simply stacking every sample together will likely bury the sparse regional effect.

Baseline: separate PNS. If we have enough samples, the best way to estimate the regional network is to run PNS within each region separately. We call this method as the baseline.

B.1 THE NECESSITY OF JOINT ESTIMATION

Network estimation is unstable given small sample size.. We first using simulation to show how unstable the network estimation can be if the sample size is too small. We consider the similar simulaiton steps in the literature:

- Generate a random graph $G^{init} \in \mathbb{R}^{p\times p}$ as the initialization.

- Get the "upper triangular" adjacency matrix of this graph and replace any non null coefficient by a random realization of a uniform variable (e.g. $U(0.8, 1)$, but any interval is possible), which then allows to define an upper triangular weight matrix, W.
- Compute the following matrix $M := (I + W)^\top (I + W)$, where is the identity matrix, defining a new graph G slightly different from the initial graph G^{init}, but above all defining a sparse positive definite matrix M.
- Normalize this matrix to get a partial correlation matrix Π.
- Generate the dataset from the multivariate Gaussian distribution $X \sim N(\boldsymbol{\mu}, \Pi^{-1})$, with $\mu_i \sim N(0, 1)$ for each dimension i.

We consider $p = 3000$, $n \in \{28, 28*4, \ldots, 28*20\}$. (Since the smallest region has only 28 samples). In PNS, we set the genes with the top 700 degrees as important genes to recover edges upon.

We measure the instability by the following experiments: given n samples, we generate $b = 20$ subsets of samples by excluding a random $1/28$ of the total samples. For example, for $n = 28$, we will end up with b subsamples $X^1, \ldots X^b$, where each two are only differ by at most 2 samples. Then we run PNS given each of the subsamples and get the estimated partial correlation network $\widehat{\Pi}^1, \ldots \widehat{\Pi}^b$. Then we compute the instability by

$$\text{Instability} := \frac{1}{p^2} \sum_{ij} \overline{\widehat{\Pi}}_{ij}; \quad \text{where } \overline{\widehat{\Pi}} := \frac{1}{b} \sum_{k=1}^{b} \widehat{\Pi}^k. \tag{B.3}$$

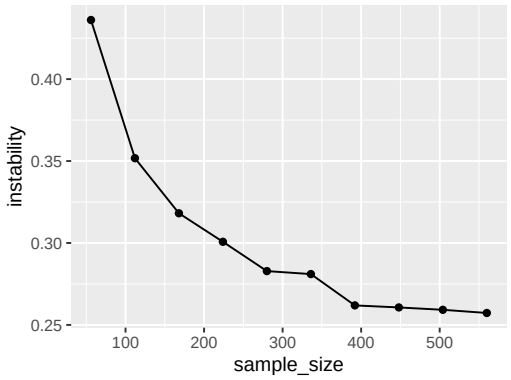


Figure B.1: The instability of edge estimation versus sample size.

Network estimation has low power with separate PNS. Then we consider the setting of simulating multiple regions. To mimic the real data setting we have, I just simulate three regions. First we start with simulating three partial correlation networks:

- Generate a random graph $G^{init} \in \mathbb{R}^{p\times p}$ as the initialization.
- Generate a region specific random subgraph $S_r^{init} \in \mathbb{R}^{d\times d}$ for each region r, with higher degree per node.
- Get the regional random graph r via selecting d nodes $\mathcal{D}_r$ out of all p nodes, and modify the edge for those d nodes as $G_r^{init}(i,j) = \mathbf{I}\{G^{init}(i,j) = 1 \text{ or } S_r^{init}(i,j) = 1\}$ for $i,j \in \mathcal{D}_r$. So now we have regional graph that have edges only differ in some subset of nodes.
- Get the partial correlation matrix Π_r for each region as before.
- Generate the dataset for region r from the multivariate Gaussian distribution $X \sim N(\boldsymbol{\mu}, \Pi_r^{-1})$, with $\mu_i \sim N(0,1)$ for each dimension i.

We consider $p = 300$, $d = 30$, and select important genes as the union of genes with the top 50 degrees in each regional graph. (This gives us 100 important genes). We simulate $n_1 = 28, n_2 = 132, n_3 = 94$ samples for regions $1, 2, 3$, which exactly mimics the number of samples of our real dataset. Then, given the nonzero entries of Π_1, Π_2, Π_3 as truth, we can compute the power and false discovery rate of edge recovery.

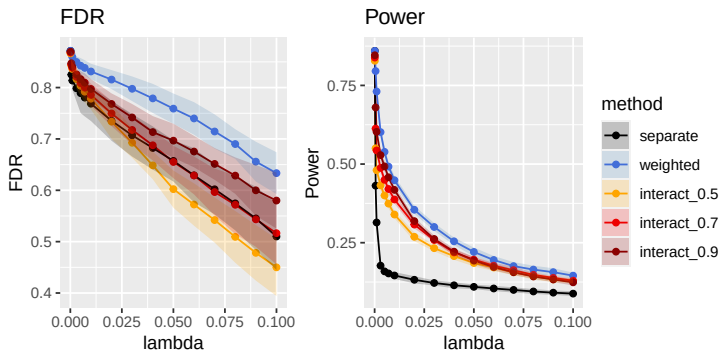


Figure B.2: The power and FDR of edge recovery versus penalty λ using different methods. We report the maximum FDR and minimum Power over three regions for each method. The results are averaged over 20 independent trials, where the bandwidth indicates the standard deviation.

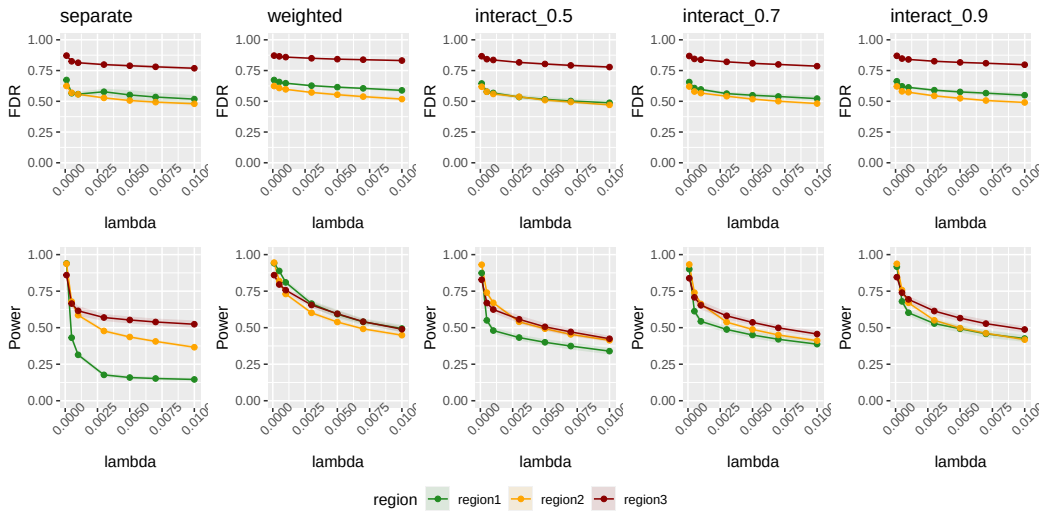


Figure B.3: The power and FDR of edge recovery versus penalty λ for each region using different methods. We report the average FDR and Power over 20 independent trials, where the bandwidth indicates the standard deviation.

B.2 FDR control in large-scale graphical models

A better screening. Identifying large-scale conditional dependence structures through graphical models is a challenging yet practical problem. Under ultra-high dimensional settings, a screening procedure is generally suggested before variable selection to reduce computational costs. However, most existing screening methods examine the marginal correlations, thus not suitable to discover the conditional dependence in graphical models. To overcome this issue, Wang and Leng (2015),Zheng et al. (2020) propose a new procedure called graphical uniform joint screening (GUS) for edge identification in graphical models. Instead of screening out edges node-wisely, GUS utilizes a uniform threshold for all statistics indicating the significance of different edges to adapt to various kinds of graphical structures. They demonstrate that GUS enjoys the sure screening property and even the screening consistency by preserving the rankings of the significant edges. Furthermore, a scalable implementation of GUS is developed for big data applications.

Network estimation with FDR control. Recently, Li and Maathuis (2021) and Zhou et al. (2022) propose procedures called the high-dimensional graphical knockoff filter to control the overall FDR for large-scale graph recovery. The proposed procedure enjoys not only theoretical guarantees and high power but also the robustness of FDR control even when the population precision matrices of predictors are replaced by consistent estimates. Furthermore, a scalable implementation approach is developed such that all knockoff variables can be generated through one single

estimation of the overall graphical structure.
However, after trying out the above two techniques on our simulation and data, the performance is poor: though FDR is lower (but still not controled), the power is only about 0.01.

B.3 Joint estimation methods

In this section, we propose three directions for joint estimation of region-specific partial gene network and provide detailed methodologies for the practical ones among them (the first two directions).

B.3.1 Approach 1: weighted PNS

First, we propose an intuitive weighted ensemble method, where weights comes from prior knowledge about how likely a sample is from each region:

$$\widehat{\boldsymbol{\beta}}^{(r)} := \arg\min_{\boldsymbol{\beta}} \sum_{k=1}^{n} \mathbb{P}(k \in \text{Region } r)\big(X_{ki} - \boldsymbol{X}_{k,\mathcal{N}(i)}\boldsymbol{\beta}\big)^2 + \lambda||\boldsymbol{\beta}||_1 \tag{B.4}$$

We can estimate $\mathbb{P}(k \in \text{Region } r)$ from the data itself.

Unsupervised estimation of weights. First I tried to use soft-clustering to gain weights but unfortunately, the data do not form very distinct clusters (see Figure A.5). Also, the Parietal lobule (the three blue-colored regions) seems to be scattered around the whole place. Therefore, we remove this lobule from our consideration now, also we merge the regions inside the same lobule (such that we can have a more stable estimation).

Supervised estimation of weights. In the following, we focus on the BA17, Frontal, Temporal lobules(regions). We now use supervised learning to gain weights: specifically, I fit a shallow classification tree such that the estimated probability distribution over classes is not too spiky. Then, for each class, I use the estimated probability of sample i being a member of this class as this sample's weight. See Figure B.5 for calculated weights. I removed one region as its samples are highly overlapping with other regions from tSNE plot.

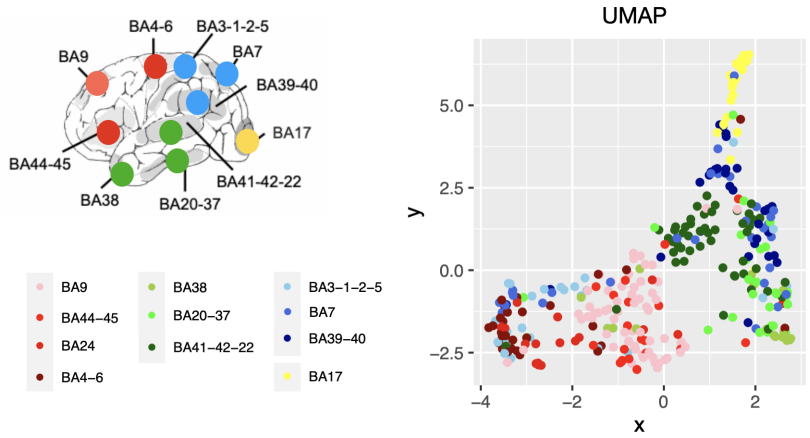


Figure B.4: The embedding of the expression data using all the region DE genes. We use all the DE genes with p-value < 0.01 (~ 800 genes). It seems like many regions are mixed together. And not all the regions from the same cortical lobule are placed together.

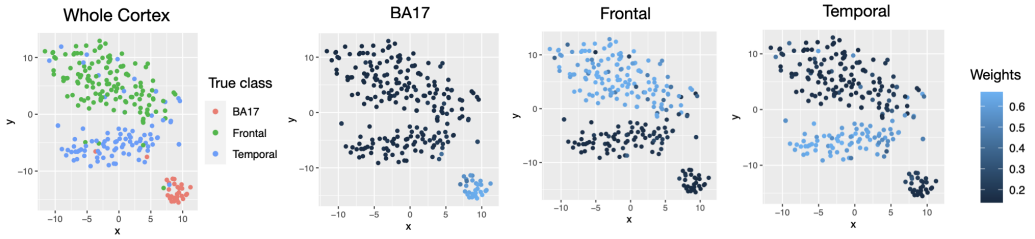


Figure B.5: The tSNE embedding of the DE genes with samples colored by true class and by class-specific weights learned by classification.

B.3.2 Approach 2: interactive PNS

Another more statistical way of introducing this regional variation is incoporating the region as a categorical feature, and considering the interaction between it and each of the neighbour genes:

$$x_i = \boldsymbol{x}_{\mathcal{N}(i)}^\top \boldsymbol{\beta} + \sum_{r=1}^{R} \gamma_r \mathbb{I}_{c=C_r} + \sum_{j \in \mathcal{N}(i)} \sum_{r=1}^{R} \alpha_{(j-1)R+r} x_j \mathbb{I}_{c=C_r}, \tag{B.5}$$
$$\boldsymbol{\beta} \in \mathbb{R}^{p \times 1},\ \boldsymbol{\gamma} \in \mathbb{R}^{R \times 1},\ \boldsymbol{\alpha} \in \mathbb{R}^{pR \times 1},$$

where c is a categorical feature taking values in $\{C_0, \ldots, C_R\}$ representing which region a sample is from, and p is the size of neighbourhood.

Weighted lasso. In practice, we observe that the interaction effects (i.e. the α coefficient) are often overwhelmed by the main effects (i.e. β coefficient), if using the same lasso penalty. To emphasize more the interaction effect, we use different levels of lasso penalty on the main effect and the interaction effect.

$$(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}}, \widehat{\boldsymbol{\alpha}}) := \arg\min_{\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\alpha}} \sum_{k=1}^{n} (X_{ki} - \boldsymbol{X}_{k,\mathcal{N}(i)}\boldsymbol{\beta} - \boldsymbol{D}_k\boldsymbol{\gamma} - \boldsymbol{T}_k\boldsymbol{\alpha})^2 + \lambda_1||\boldsymbol{\beta}||_1 + \lambda_1||\boldsymbol{\gamma}||_1 + \lambda_2||\boldsymbol{\alpha}||_1, \tag{B.6}$$

where $\boldsymbol{D}, \boldsymbol{T}$ is the matrix representation of the categorical feature and the interaction term respectively.
Then for region C_r, the region-wise partial correlation $\boldsymbol{\beta^{(r)}}$ can be calculated as

$$\beta^{(r)} = \begin{cases} \widehat{\boldsymbol{\beta}}, & \text{if } r = 0; \\ \widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\alpha}}^{(r)}, & \text{if } r > 0, \quad \text{where } \widehat{\boldsymbol{\alpha}}^{(r)} := \big(\widehat{\alpha}_r, \widehat{\alpha}_{r+R}, \ldots, \widehat{\alpha}_{r+(p-1)R}\big). \end{cases} \tag{B.7}$$

Related work. In fact, this approach coincides with an existing work Cheng et al. (2017), where they extend Gaussian graphical models to mixed Gaussian graphical models, with categorical features incorporated into the joint density. Specifically, consider

$$p(\boldsymbol{x}, c) \propto \exp\Big\{ \sum_{r=1}^{R} \eta_r \mathbb{I}_{c=C_r} + \boldsymbol{x}^\top(\alpha_0 + \sum_{r=1}^{R} \alpha_r \mathbb{I}_{c=C_r}) - \frac{1}{2}\boldsymbol{x}^\top(\Phi^0 + \sum_{r=1}^{R} \Phi^r \mathbb{I}_{c=C_r})\boldsymbol{x} \Big\}, \tag{B.8}$$

where $diag(\Phi^r) = 0$ for all r. Then the conditional distribution of x_i given $\boldsymbol{x}_{-i}, c$ is given by

$$x_i \sim \frac{1}{\Phi^0_{i,i}} \Big(\sum_{r=1}^{R} \eta_r \mathbb{I}_{c=C_r} - \sum_{j\neq i} (\Phi^0_{ij} + \sum_{r=1}^{R} \Phi^r_{ij} \mathbb{I}_{c=C_r}) x_j + e_i \Big) \tag{B.9}$$

$$:= \frac{1}{\Phi^0_{i,i}} \Big(\sum_{r=1}^{R} \eta_r \mathbb{I}_{c=C_r} - \sum_{j\neq i} \beta_{ij} x_j - \sum_{j\neq i} \sum_{r=1}^{R} \alpha_{jr} x_j \mathbb{I}_{c=C_r} + e_i \Big) \tag{B.10}$$

where $e_i \sim N(0,1)$, and $K := \sum_{r=0}^{R} \Phi^r$.

For the loss function, in addition to mean squared error, they also consider following overlapping group lasso penalization

$$\text{penalty} := \sum_{r=1}^{R} ||(\eta_r, \boldsymbol{\alpha}_{\cdot,r})||_2 + \sum_{j\neq i} ||(\beta_{ij}, \boldsymbol{\alpha}_{i,\cdot})||_2. \tag{B.11}$$

Note that this penalty is overlapping group lasso, which is hard to optimize, therefore they propose to use an upper bound of this penalty instead (using $||\boldsymbol{b}||_2 \leq ||\boldsymbol{b}||_1$ for any vector $\boldsymbol{b}$ to get the upper bound):

$$\text{penalty}' := \sum_{r=1}^{R} |\eta_r| + 2\sum_{r=1}^{R}\sum_{j\neq i} |\boldsymbol{\alpha}_{jr}| + \sum_{j\neq i} |\beta_{ij}|, \tag{B.12}$$

which essentially gives them a weighted lasso.
We are essentially doing the same thing, just that we use different penalization:

$$\text{our penalty} := \sum_{r=1}^{R} |\eta_r| + \text{ratio}\sum_{r=1}^{R}\sum_{j\neq i} |\boldsymbol{\alpha}_{jr}| + \sum_{j\neq i} |\beta_{ij}|, \tag{B.13}$$

i.e. the penalty weight is a tuning parameter in our scenario.

B.3.3 Approach 3: Bayesian modeling

There is a group of methods using Bayesian modeling (e.g. Li et al. (2019)). One of the most popular approaches for Bayesian inference with Gaussian graphical models is the G-Wishart prior. The G-Wishart prior estimates the precision matrices with exact zeros in the off-diagonal elements and enjoys the conjugacy with the Gaussian likelihood. However, posterior inference under the G-Wishart prior can be computationally burdensome and has to rely on stochastic search algorithms over the large model space, consisting of all possible graphs. In recent years, several classes of shrinkage priors have been proposed for estimating large precision matrices, including the graphical lasso prior, the continuous spike- and-slab prior, and the graphical horseshoe prior. This line of work draws direct connections between penalized likelihood schemes and, as their names suggest, the posterior modes in a Bayesian setting. Unlike the G-Wishart prior, these shrinkage priors do not take point mass at zero for the off-diagonal elements in the precision matrix, and thus usually lead to efficient block sampling algorithms with improved scalability. However, fully Bayesian procedures still need to rely on stochastic search to achieve model selection, making it less appealing for many problems. To address this issue, deterministic algorithms have been proposed to perform fast posterior exploration and mode searching in Gaussian graphical models.

We did not dive deep into this direction as this line of methods could be too complex and thus not suitable for high-dimensional problems.

B.4 Hyperparameter selection

Criteria based on network structure. For a single network, we can just use the R^2 of the network fitting to the power law to choose the hyperparameter.

Criteria based on regional effect. Since we are estimating multiple networks, and we hope to see meaningful differences between them. Therefore, in addition to the power law criteria, we also introduce the following criteria based on the permutation test of meaningful differences. Considering we already defined some good metrics of network differences, we denote those L metrics using a single vector $\boldsymbol{u} \in \mathbb{R}^{L\times 1}$.

Then we permute the indices of the true categorical feature. We use this permuted categorical feature to estimate the regional network as usual, and then compute the network differences measure. We do such permutation for m times and get a list of network differences estimation $\{\widetilde{\boldsymbol{u}}^1, \ldots, \widetilde{\boldsymbol{u}}^M\}$.

Then we compute the standardized true network differences for each metric:

$$z_l := \frac{u_l - \frac{1}{M}\sum_{m=1}^{M} \widetilde{u}_l^m}{std(\{\widetilde{u}_l^1, \ldots, \widetilde{u}_l^M\})}, \quad , l = 1 \ldots, L \tag{B.14}$$

which is basically treating $\{\widetilde{\boldsymbol{u}}^1, \ldots, \widetilde{\boldsymbol{u}}^M\}$ as an estimation of the network differences due to randomness/instability.

Criteria based on stability. Sedgewick et al. (2016) used a very similar idea like our interaction model, and they propose to use the following stability measure to select hyperparameters.

They draw N subsamples of size b without replacement and compute the network using those subsamples and get network $A_1 \ldots A_N$. Then they compute the averaged variance for each edge, treating the edge as a Bernoulli random variable. The final instability metric $S = \texttt{Mean}(2 * (\overline{A}) * (1 - \overline{A}))$, where $\overline{A} = \frac{1}{N}\sum_{i=1}^{N} A_i$ and $*$ means element-wise multiplication. They propose to select the parameter such that the stability is minimized. We can see that, however, the instability measure in our case is not informative enough: it does not vary much for different λ (here when I do the `Mean`, I do it over only the nonzero entries, such that we do not trivially assign more stability to very big λ, since those can be stable just due to that there is no edge left).

Combine all criteria. For each pair of the network, we can calculate such a difference vector. For each network, we can also calculate the power law non-fitness metric, R^2. We stack all those vectors together as a long vector $\boldsymbol{v}$. We require each dimension of $\boldsymbol{v}$ to have range $[0, 1]$, with 0 being the ideal best case, and 1 being the worst case possible. Then we choose hyperparameters $\boldsymbol{h}$ using

$$\boldsymbol{h} := \arg\max ||\boldsymbol{v}(\boldsymbol{h})|| \tag{B.15}$$

Note that we need to do this on a hold-out set, to avoid p-hacking! Not sure whether this will cause us problems since we already have a too-small sample size.

B.5 Results

When applied to the Danel data, we only consider on the lobule level (otherwise the estimation is too unstable). We find that the Parietal lobule (the three blue-colored regions) seems to be mixed with the other three lobules from the embedding of the expression data using all the region DE genes Figure B.4. Therefore, we remove this lobule from our consideration now. See Figure B.5 for calculated weights for each lobule when using the weighted PNS method.

In Figure B.6 and Figure B.7, we provide a comparison of three methods we mentioned before in regional partial network estimation. Specifically, Figure B.6 tests the significance of regional differences using different regional network estimation methods, and we can see that while none of them provide significant results over all differences metrices, interact PNS seems to produce significance in most metrics. Also, visually we can see in Figure B.7 that, interact PNS gives the most interesting contrast of network structure across regions.

The performance of these methods on downstream active and reactive DE cluster identification is left to future work.

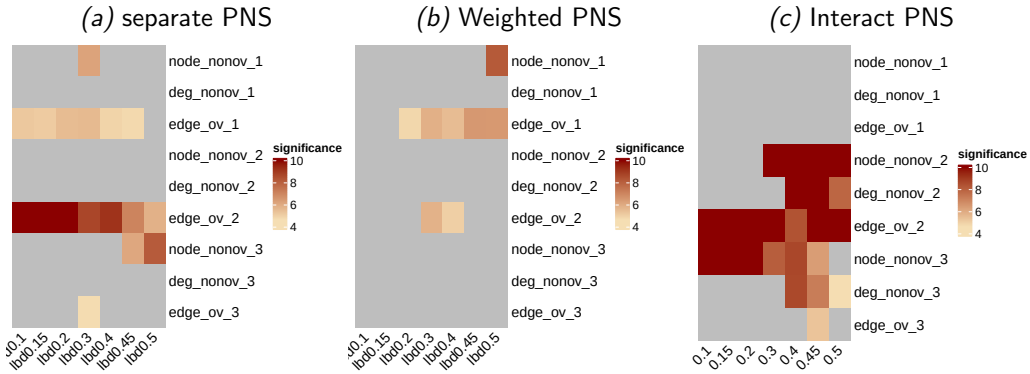


Figure B.6: The significance of regional differences using different regional network estimation methods. Each column is of the same lasso penalty λ, and each row is of the same regional differences measure. Specifically, "node_nonov" is the proportion of nonoverlapping nodes, "deg_nonov" is the ratio between the averaged degree of nonoverlapping nodes and overlapping nodes; and "edge_ov" is the proportion of conflict edges among overlapping nodes. The suffix number 1 means BA17 v.s. Frontal; 2 means BA17 v.s. Temporal; and 3 means Frontal v.s. Temporal.

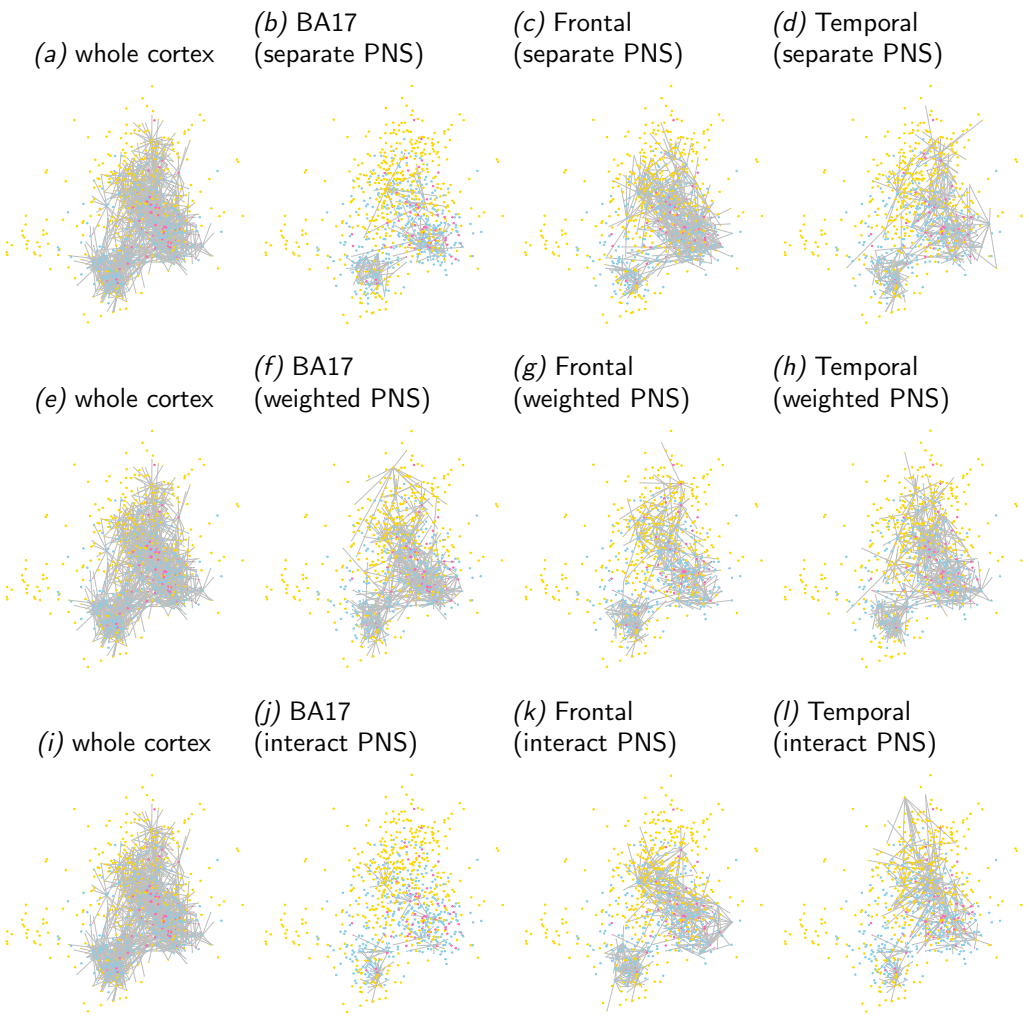


Figure B.7: The visualization of regional networks estimated via different methods, with the hyperparameters chosen via the above-described way. We can see that, the interactive PNS method seems to give the most interesting regional differences. Here we fix the position of each node across all networks for better visualization of edge differences (and therefore the networks seem a bit messy as the layout is not optimized for each network). Each node is colored by the initial hidden states (active: hotpink; reactive: skyblue; others: yellow) estimated from the data before HMRF regularization.