GRAPHICAL MODELS FOR HIGH DIMENSIONAL DATA WITH GENOMIC APPLICATIONS

Jenny Yang

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Biostatistics in the Gillings School of Global Public Health.

Chapel Hill
2017

Approved by:

Wei Sun

Stephanie Engel

Quefeng Li

Yufeng Liu

Donglin Zeng

ABSTRACT

Jenny Yang: Graphical Models for High Dimensional Data with Genomic Applications
(Under the direction of Wei Sun)

Many previous studies have demonstrated that gene expression or other types of -omic features collected from patients can help disease diagnosis or treatment selection. For example, a few recent studies demonstrated that gene expression data collected from cancer cell lines are highly informative to predict cancer drug sensitivity (Garnett et al. 2012, Barretina et al. 2012, Chen et al. 2016b). This is partly because many cancer drugs are targeted drugs that perturb a particular mutated gene or protein, and thus having that mutation, or observing the consequence of such mutation in gene expression data, is highly informative for drug sensitivity prediction. Such systematic studies of drug sensitivities require giving different drugs in a series of doses to the same cell line, which is obviously not possible for the human studies. More sophisticated methods are needed to estimate potential effects of cancer drugs based on observational data. Since the effect of a targeted cancer drug can be considered as an intervention to the molecular system of cancer cells, a directed graphical model for gene-gene associations is a natural choice to model the molecular system and to study the consequence of such interventions.

In this dissertation, we develop new statistical methods to estimate DAGs using high dimensional -omic data under two scenarios: i) with a model-free approach and ii) single cell RNA-seq data (scRNAseq). In the 1st chapter, we will give a brief introduction to graphical models, the various statistical characterizations of graphical models and the most current approaches to estimate graph structures. Then, we will

review the scRNAseq data and current approaches to analyze scRNAseq data. Next, in Chapter 2, we propose a model-free method to estimate graphical models in two steps. The first step uses a model-free variable selection method based on the principles of sufficient dimension reduction. Then, the second step uses a non-parametric conditional independence testing method which utilizes embeddings of the conditional spaces into reproducing kernel Hilbert spaces. We will review some theoretical background in order to establish the asymptotic graphical model estimation consistency of this two-step approach. We examine its performance in simulations and TCGA breast cancer data, where we find significant improvements from current methods that require strong model assumptions. In Chapter 3, we propose a graphical model algorithm to analyze scRNAseq data. Similar to the previous algorithm, we create a two-step estimation method which utilizes a joint penalized zero-inflation model. We assess its performance and drawbacks in simulations. Then, we examined its utility when applied after clustering to a sample of 68k peripheral blood mononuclear cells with multiple subpopulations.

To Mama.

For Chris and Luc.

ACKNOWLEDGMENTS

After a good three and a half years of this labor of love, the inumerable times when I thought this would never be complete, it is finally time for me to look back on my journey and recognize everyone who has held me up on their shoulders to get me here.

First, a special thanks and deep gratitude to my advisor, Dr. Wei Sun. He was rare in his support of me and my work, in his patience for my struggles, and in his generosity with his time and funding. I would also like to thank my committee for their patience and insight into my work. Especially, Drs. Donglin Zeng and Stephanie Engel who were instrumental in guiding my transition from coursework to dissertation.

Secondly, I'd like give a shout-out to UNC's ITS computing. In my entire time at UNC, they haven't let me down once with prompt replies to my panicked emails.

Finally, I would like to thank my invaluable support network. My friends that kept me sane by commiserating and supporting me. Who believed in me, although all evidence might be pointing to the contrary. Especially Briana Stephenson and Emily Butler, who My parents, who sacrificed their entire way of life to bring me to the U.S. and keep me here. My father, who always emphasized the importance of academics and was my role model in obtaining a PhD. My mother, who inspires me everyday with her work ethic. My brothers for always being there with an encouraging word. Especially Philip, who never hesitated to travel back home when I needed him to take care of something because I needed to work or travel.

Last but not least, my husband and son. What can I say about them? Without them, all of this would be meaningless. While my son Lucien made my path to graduation considerably more difficult, he has given me renewed focus and ambition. I don't

know what I did to deserve my husband, Christian. He has given up so much time and time again to keep me going and give me the best support anyone could hope for. He has listened to my exasperated tears and my exhausted rants, he has shouldered the burdens that I have heaped on him when I couldn't hold up my end of the household anymore. He has accepted all of this and done all of this with little complaint. Through all of this, he has had his own career to take care of which he has excelled at. My dissertation is a love letter to these two gems in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: LITERATURE REVIEW

1.1 Graphical Models

Directed acyclic graphs (DAGs) are directed graphical models describing the conditional independence amongst a number of random variables [Pearl (2009)]. The "acyclic" part of the name refers to the constraint that there is no directed cycle (or loop) in the graph. This constraint is necessary for causal inference [Pearl (2009), Spirtes et al. (2000)]. Consider a set of p random variables, $\mathbf{X} := \{X_1, ..., X_p\}$ with true DAG structure of $\mathcal{G}^{(0)} := (\mathbf{V}, \mathbf{E}^{(0)})$ – where $\mathbf{V} := \{X_1, ..., X_p\}$ is the set of vertices corresponding to $\mathbf{X}$ and $\mathbf{E}^{(0)}$ is the set of directed edges. The skeleton of a DAG is defined as an undirected graph, obtained by removing the directions of all the edges in a DAG. A v-structure is a structure of $X_1 \to X_2 \leftarrow X_3$, where X_1 and X_3 are not directly connected. The DAGs that share the same skeleton and the same set of v-structures form a Markov equivalence class, and all the DAGs within the same Markov Equivalence class encode the same set of conditional independence relations of the p random variables.

For the purposes of this proposal, we focus on the problem of estimating a DAG from high dimensional observational data. Without randomized interventions, a popular assumption used in graphical model estimation is that of ordering. This is where we assume that for vertices $X_1, ..., X_p$, a vertex with a smaller subscript will always be of an earlier generation. However, such knowledge of natural ordering is often not available, especially in high dimensional settings. Without any additional information, one cannot estimate the individual DAG from observational data but can estimate the

most likely Markov Equivalence class.

Without knowing edge directions it is impossible to separate the parents of a node from its children, making estimation based on independence tests conditional on X_i^{pa} impossible. An alternative is to define the edge between X_i and X_j by $E_{i,j} = I\{X_i \perp X_j | X_{-(i,j)}\}$, which allows us to do conditional independence testing based on observational data. Following this edge definition, we can recover the moral graph, which is constructed by connecting the co-parents of v-structures of a DAG skeleton. Then, one way to estimate the skeleton of a DAG is to use a two-step process; first, estimating the moral graph, then removing the connection between co-parents of v-structures. In addition, even with observational data, we can orient a limited number of edges in the skeleton corresponding to v-structures. Such a partially directed graph can be used to make useful causal inference [Maathuis et al. (2009)]. This is the basic procedure we will propose.

1.1.1 Undirected Graphical Model Estimation

The most intensely studied area for estimating these moral graphs are concentrated on Gaussian Graphical Models (GGM) or the extended family of nonparanormal models as dubbed by [Liu et al. (2009)].

Gaussian Graphical Models

GGMâĂŹs are graphical models with the underlying assumption that $\mathbf{X} \sim N_p(\mu, \Sigma)$. The edge set, $\mathbf{E}^{(0)}$, is usually estimated by the non-zero entries of the precision matrix Σ^{-1} where $E_{i,j} = I(\Sigma^{-1}_{i,j} \neq 0)$.

For $p < n$ problems, one approach to estimate Gaussian Graphical Models is to use a greedy stepwise forward-selection or backward-selection. [Drton and Perlman (2004)]

introduced a method to produce and utilize simultaneous p-values corresponding to partial correlations allowing for model selection in a single step, which they improved upon a few years later [Drton and Perlman (2007)]. For $p \gg n$ problems, machine learning techniques such as penalized regression or penalized maximum likelihood estimate have be utilized. For example, [Meinshausen and BÃijhlmann (2006)] proposed the neighborhood selection approach, where the neighborhood of each node is estimated using penalized regression with the L_1 penalty. Combining the results of neighborhood selection of all the p nodes provided the structure of the GGM. While this neighborhood selection method gives consistent estimates of sparse high dimensional graphical models, its estimation of precision matrix is not consistent. Several penalized likelihood algorithms have been developed to directly estimate the precision matrix by placing an L_1 penalty on the off-diagonal entries of the precision matrix [Yuan and Lin (2006), Friedman et al. (2008)]. These penalized estimation methods use some tuning parameters to tune the strength of the penalty. The optimal tuning parameters are often selected by scoring a series of models estimated from a pre-defined tuning parameter grid. For example, BIC was often used as the scoring function [Lam and Fan (2009)].

[Liu et al. (2009)] developed a graphical model estimation method that can be applied for a class of distributions they termed nonparanormal distribution, which includes any distribution that can be transformed into a multivariate Gaussian distribution. Therefore, their method allows for the edges of the graphical model to be coded by a precision matrix of transformed data [Liu et al. (2009)]. They developed a nonparametric method using additive models to estimate the transformed Gaussian data and used an L_1 penalty in order to extend the method to high dimensional settings. In 2012, they improve upon the method and show that it has the same convergence rates as the state of the art Gaussian Graphical Models [Liu et al. (2012)].

1.1.2 Directed Acyclic Model Estimation

There are two basic approaches that have been developed for estimating DAGs or DAG skeletons: search-and-score based algorithms and conditional independence testing algorithms. Search-and-score based algorithms generate scores (*i.e.* AIC, BIC, log likelihood, Bayesian Dirichlet probability or Kullback-Leibler divergence) for DAGs and selects the one with the best score [Chickering (2003), De Campos and Ji (2011)]. This approach is computationally infeasible for high dimension problems with $p \gg n$, where p is the number of variables and n is sample size. In contrast, the conditional independence testing algorithm is computationally much more efficient and can give consistent estimate of DAG skeletons in high dimensional settings [Ha et al. (2016a)]. Thus, for the purpose of this paper we chose to focus on conditional independence testing algorithms. First, we introduce some definitions.

Definition 1 from [Pearl (2009)] (d-separation)

A path p is said to be d-separated (or blocked) by a set of nodes Z if and only if:

(1) p contains a chain $i \to m \to j$ or a fork $i \leftarrow m \to j$ such that the middle node m is in Z, or

(2) p contains an inverted fork (or collider) $i \to m \leftarrow j$ such that the middle node m is not in Z and such that no descendant of m is in Z.

A set Z is said to be d-separate X from Y if and only if Z blocks every path from a node in X to a node in Y.

Definition 2 from [Spirtes et al. (2000)] (Markov)

The distribution of $\mathbf{V}$ is Markov to $\mathcal{G} := (\mathbf{V}, \mathbf{E})$ if and only if each variable $X_j \in \mathbf{V}$, is conditionally independent of its non-effects (non-descendants) given its direct causes

(parents).

Definition 3 from [Spirtes et al. (2000), Kalisch and BÃijhlmann (2007)] (Faithfulness) A probability distribution $\mathbf{P}$ on $\mathbb{R}^p$ is said to be faithful with respect to a graph $\mathcal{G}$ if conditional dependencies of the distribution can be inferred from so-called d-separation in the graph $\mathcal{G}$ and vice versa. More precisely: consider p random variables $\mathbf{X} = \{X_1, ..., X_p\} \sim \mathbf{P}$. Faithfulness of $\mathbf{P}$ with respect to $\mathcal{G}$ means: for any $i, j \in \mathbf{V}$ with $i \neq j$ and any set $\mathbf{s} \subseteq \mathbf{V}$, X_i and X_j are conditionally independent given $\{X_r; r \in \mathbf{s}\} \Leftrightarrow$ vertex i and vertex j are d-separated by the set $\mathbf{s}$.

Faithfulness was originally defined by [Spirtes et al. (2000)] as when the only conditional independencies in P are those found by the Markov condition on $\mathcal{G}$. It is apparent then that the assumption of faithfulness is required to be able to use conditional independence testing graph estimation. This body of work is largely built on the following central theorem:

The PC-Algorithm and its Relatives

The PC algorithm was named after the first names of its two authors (Peter Spirtes and Clark Glymour). It assesses the existence of each edge in a graphical model by conditional independence testing. The following theorem provides some theoretical justification for such testing.

Theorem 1 from [Spirtes et al. (2000)] If $\mathbf{P}$ is faithful to some directed acyclic graph, then $\mathbf{P}$ is faithful to $\mathcal{G}$ if and only if:

(1) For all vertices, X, Y of $\mathcal{G}, X$ and Y are adjacent if and only if X and Y are dependent conditional on every set of vertices of $\mathcal{G}$ that does not include X or Y; and

(2) For all vertices X, Y, and Z such that X is adjacent to Y and Y is adjacent to Z and X and Z are not adjacent, $X \to Y \leftarrow Z$ is a subgraph of $\mathcal{G}$ if and only if X and Z are dependent conditional on every set containing Y but not X or Z.

SGS algorithm is an early conditional independence testing algorithm for graphical model estimation [Spirtes et al. (2000)]. It is essentially a brute force method that starts with a fully linked undirected graph and then seeks to remove edges by evaluating all possible conditional independence tests. The application of SGS algorithm is limited by its high computational cost. The PC algorithm seeks to improve upon the SGS by testing as few d-separation relationships as possible. It takes the same fully linked undirected graph input and then thins the edges by first using marginal independence testing, then conditional testing based on subsets of adjacent variables of size one, size two, and so on.

Extensive work has been done to improve upon the various limitations of the PC algorithm including its order dependency (PC-stable [Colombo and Maathuis (2014)]) and error accumulation from "unfaithfulness" (Conservative-PC [Lemeire et al. (2010)]), as well as to extend its statistical properties to non-paranormal graphical models [Harris and Drton (2013)]. While the PC algorithm and its derivatives have worst case computation time bounded by $O(p^d)$, their expected computation time is of the order $O(n^2\binom{\bar{p}}{d})$, where d is the degree of vertices in the true DAG and $\binom{\bar{p}}{d}$ is the average of $\binom{p}{d}$ over all vertices.

The Independence Graph (IG) Algorithm is a variation of the PC algorithm, which starts with the estimated moral graph instead of the fully linked undirected graph [Spirtes et al. (2000)]. A moral graph of a DAG is defined by connecting any two parent nodes of a v-structure in the DAG. Recently, [Ha et al. (2016a)] proposed PenPC, which is a combination of moral graph estimation by penalized regression and a modified PC algorithm. More specifically, PenPC uses penalized regression with log penalty

to obtain a more accurate estimate of moral graph. Then the edges within the moral graph is further thinned by a modified PC algorithm, which searches across candidate d-separation sets based on the original input moral graph. PenPC has better performance than the PC algorithm and is computationally more efficient in high dimensional settings. All of these existing methods rely on the assumption that the data follow multivariate Gaussian distribution or the data can be transformed to follow multivariate Gaussian distribution for conditional independence testing. Furthermore, the penalized estimation of moral graph requires the assumption that one variable is associated with other variables through a linear regression model. In this dissertation, we proposed a new method that does not require these potentially restrictive assumptions.

1.1.3 Considering Model-Free Settings

Without the assumption that the data arises from either a multivariate normal distribution or a multivariate Gaussian copula, constructing a graphical model becomes much more complicated. This is because multivariate normality provides a very convenient property which allows graphical structure to commute under convolution. Under the model-free setting we do not place distributional assumptions (e.g. Gaussian) or relationship assumptions (e.g. homoscedastic linearity) on the data. Therefore, the covariance of the data usually does not provide the graphical structure [Loh and Wainwright (2013)]. Instead, we consider that under the assumption of a positive and continuous density for y, the local Markov property infers global and pairwise Markov properties. Hence, we use node-wise conditional independence inference with neighborhood selection in order to obtain the global Markov random field. Once we obtain the global Markov random field, it becomes a question of removing v-structures using conditional independence testing in order to obtain the final undirected skeleton.

Testing conditional independence of continuous variables with a large conditioning

space is a particularly challenging problem in the model-free setting. We examined the body of available methods which fall into the following categories: i) distance based with explicit estimation of conditional densities [Su and White (2007)], ii) discretizing the conditioning set and performing the independence testing within each bin [Huang (2010)], iii) testing for independence against some set of transformed conditioning space [Song et al. (2009)], and iv) tests which examine the embeddings of probability distributions into reproducing kernel Hilbert spaces (RKHS) [Zhang et al. (2012)].

Tests of type i) are difficult to utilize over a wide variety of conditioning spaces as explicit estimation of the conditional densities becomes very complex, especially as the conditioning space increases. For example, [Su and White (2007)] proposes a test which measures the distance between the conditional characteristic functions and requires explicit estimation involving multiple applications of the Nadaraya-Watson leave-one-out kernel regression technique. Tests of type ii) cannot be used effectively in cases of low sample size as discretizing the conditioning space means even lower sample size for each unconditioned test per bin. Tests of type iii) are actually weaker than testing strictly for nonparametric conditional independence. For example, [Song et al. (2009)] proposes a method which tests for whether there exists some function h and parameter θ_0 such that the variables are independent given a single index function $\lambda_{\theta_0}(Z) = h(Z^\top \theta_0)$ of Z. This is a weaker condition than conditional independence. For example, if X and Y depend on two different subsets of Z which have overlap, then even for $X \perp Y|Z$ we cannot find a $\lambda_{\theta_0}(Z)$ for which X and Y are conditionally independent.

Tests of type iv) have been found to be equivalent to tests based on energy distances or distance covariances [Sejdinovic et al. (2013), SzÃľkely and Rizzo (2012)]. They use a characterization of conditional independence by covariance operators in

the RKHS from [Fukumizu et al. (2004)]. For some random vector (X, Y) on domain $\mathcal{X} \times \mathcal{Y}$, let $\mathcal{H}_\mathcal{X}$ and $\mathcal{H}_\mathcal{Y}$ be the RKHS on $\mathcal{X}$ and $\mathcal{Y}$, with kernel functions $k_\mathcal{X}$ and $k_\mathcal{Y}$, respectively. The cross-covariance operator from $\mathcal{H}_\mathcal{X}$ to $\mathcal{H}_\mathcal{Y}$ is defined by the relation: $\langle g, \sum_{YX} f \rangle = \mathbb{E}_{XY}[f(X)g(Y)] - \mathbb{E}_X[f(X)]\mathbb{E}_Y[g(Y)]$ for all $f \in \mathcal{H}_\mathcal{X}$ and $g \in \mathcal{H}_\mathcal{Y}$. The conditional cross covariance operator of (X, Y) given Z is then defined as $\Sigma_{YX|Z} = \Sigma_{YX} - \Sigma_{YZ}\Sigma_{ZZ}^{-1}\Sigma_{ZX}$. We can view this conditional cross covariance operator as the partial covariance between any two functions belonging to $\mathcal{H}_\mathcal{X}$ and $\mathcal{H}_\mathcal{Y}$ given any function belonging to $\mathcal{H}_\mathcal{Z}$. [Fukumizu et al. (2004)] shows that this cross-covariance operator is related to the conditional independence if characteristic kernels are used, based on the following lemma:

Lemma 1 from [Fukumizu et al. (2007)]

Denote $\ddot{X} \equiv (X, Z)$, $k_{\ddot{\mathcal{X}}} \equiv k_\mathcal{X} k_\mathcal{Z}$, and $\mathcal{H}_{\ddot{\mathcal{X}}}$ the RKHS corresponding to $k_{\ddot{\mathcal{X}}}$. Let the space of square integrable functions of X be denoted L_X^2, and similarly define L_Y^2 and L_Z^2. Assume $\mathcal{H}_\mathcal{X} \subset L_X^2$, $\mathcal{H}_\mathcal{Y} \subset L_Y^2$, $\mathcal{H}_\mathcal{Z} \subset L_Z^2$ and that $k_{\ddot{\mathcal{X}}} k_{\ddot{\mathcal{Y}}}$ is a characteristic kernel on $(\mathcal{X} \times \mathcal{Y}) \times \mathcal{Z}$. The characteristic kernel ensures that the statistical features of the data distribution are preserved by the kernel embedding in the RKHS space. Further, assume that $\mathcal{H}_\mathcal{Z} + \mathbb{R}$ is dense in $L^2(P_Z)$. Then

$$\Sigma_{\ddot{X}Y|Z} = 0 \Leftrightarrow X \perp Y | Z \tag{1.1}$$

1.2 Single-Cell RNA-Seq Data

A typical human cell has 6 billion base pairs of DNA and 600 million bases of mRNA [Eberwine et al. (2014)] and all diseases show some level of heterogeneity between individual cells in their pathology. This is especially apparent in cancer, a set of heterogeneous diseases stemming from the accumulation of somatic mutations. Within the same type of cancer (e.g., breast cancer or colon cancer), there is a considerable

degree of inter-patient heterogeneity, both in terms of molecular features (e.g., somatic mutations, gene expression) and a patient's response to treatments [Garnett et al. (2012), Barretina et al. (2012)]. Even within the tumor tissue of one patient, the tumor cells may have different somatic mutations, which is referred to as intra-tumor heterogeneity [Yap et al. (2012), Patel et al. (2014)].

RNA sequencing has traditionally been done on bulk tissue samples and thus a large number of cells at once (bulk RNA-seq). Analyses of this data would operate on the measure of mean values of signals from individual cells, overlooking internal interactions and heterogeneity within the cell population [Kolodziejczyk et al. (2015)]. However, advances in Next Generation Sequencing (NGS) now allow deep sequencing of a single cell. This paves the way to a variety of applications including understanding why some cells may be drug responsive but others are not and determining molecular states which may be specific to a disease for drug targeting; as well as new research techniques, such as cell level perturbations to dynamically probe cell function [Eberwine et al. (2014)]. Obtaining estimates for homogeneous subclusters of cells within a cancer promise a path towards precision medicine: medical practice tailored to the unique profile of a patient's cancer. Graphical models, such as the directed acyclic graph (DAG), are among the most promising solutions.

1.2.1 Data Generation

The workflow for obtaining scRNA-seq data involves four general steps. First, individual cells need to be isolated. This can be done using either microfluidics like Fluidigm C1 [Pollen et al. (2014)], or microtiter plates like Smart-seq2 [Picelli et al. (2014)], or droplet-based technologies like inDrop [Klein et al. (2015)]. Then, the cell needs to be lysed to allow access to the nucleic acids. There are currently no techniques to sequence mRNA itself, so cDNA needs to be obtained for sequencing using reverse

transcription. Finally, the cDNA needs to be amplified for sequencing using PCR [Tang et al. (2009), Stegle et al. (2015)]. Sequencing itself is done by applying NGS, such as the Illumina sequencing solutions.

After obtaining the raw sequence data, bulk RNA-seq techniques can be applied for sequence alignment in order to summarize the sequences and finally obtain the total read counts (TReC) of each mRNA sequence [Love et al. (2015), Chen et al. (2016b)]. Due to the small amount of starting material, a large amount of amplification needs to be done. Since amplification efficiency can vary between cells simply due to technical variability, it is recommended to use extrinsic spike-ins or unique molecular identifiers (UMIs) to facilitate normalization across cells [Stegle et al. (2015)]. Extrinsic spike-ins are tagged RNA mixes with known sequences and quantities which are added to the cell prior to amplification, in order to produce normalization factors to adjust for amplification bias. Alternatively, UMIs operate like barcodes and allow read counts enumeration that are independent of amplification bias. In the ideal setting, full coverage UMIs would provide data most free of technical variability, however there are many non-UMI-based protocols currently utilized.

1.2.2 Statistical Analysis Challenges

There are a number of differences between scRNA-seq data and bulk RNA-seq data primarily attributable to the low amount of starting RNA in a single cell and the inherent heterogeneity between individual cells. First, the amount of noise in scRNA-seq data is greater, requiring a larger sample size for adequate power. This is not especially problematic, as it is typically easier to separate a single tissue sample into a larger number of individual cell samples than it is to harvest a large number of tissue samples. Secondly, there is an increase to the incident of zero counts. While detection limits have been suspected to play a part, evidence has shown that the limit of detection

is essentially zero and the empirical distribution of data is different from what would be expected from censoring. Instead, the current suspected culprit for zero-inflation ranges from a variety of sources including drop out from amplification, inherent heterogeneity of cell expression, and technical factors [Kolodziejczyk et al. (2015), McDavid et al. (2016)]. Heterogeneity of cell size means that some cells may just have a larger amount of total nucleic acids. This is often normalized for using the ratio of RNA reads to spike-ins reads, termed the endogenous RNA size factor. A more general technique uses relative expression counts instead of raw counts [Stegle et al. (2015)].

The heterogneity between individual cells is not confined to gene expression, but also to cell health and function. The individual cells may be in varying states of degradation or even apoptosis. In order to control for these confounding factors, bulk RNA-seq quality control tools are often used to check the proportion of reads which map back to the genome, the proportion of reads of RNA versus spike-ins, and the use of PCA to cluster good quality cells from bad quality cells [Stegle et al. (2015)].

There are additional challenges more specific to graphical model estimation using scRNA-seq data. It has been shown that the robustness of networks derived from scRNA-seq data may be dependent upon composition of cell types [Mahata et al. (2014)]. scRNA-seq data also may reveal less meaningful associations between genes. For example, if two cells are in different phases of cell cycle, a large number of genes may be associated due to the cell-cycle effect. Therefore it is crucial to remove the effects of such confounding factors.

1.2.3 Current Approaches for Analyses

Current approaches to analyses scRNA-seq data largely fall under two umbrellas: grouping or clustering cells by type or state and construction of gene regulatory networks.

Clustering techniques used for bulk RNA-seq data, which were used to classify tissues into separate types, can be applied to scRNA-seq data. These techniques fall under either dimension reduction or hierarchical clustering techniques. They characterize a sample by its composition of cell types, defined by the expression profile of each cell cluster. Often, this is done in order to find a set of marker genes for easier identification of these clusters in future samples. The identification of these marker genes are the subject of much study, and are usually found by either comparing differential expression of genes between clusters or identifying highly variable genes (HVGs). For scRNA-seq data, these techniques typically use either the Poisson or Negative Binomial distributions in order to model the TReC, which allows confounding covariates to be adjusted for and even account for spatial correlations using the well solved generalized linear model solutions [Anders and Huber (2010), Hardcastle and Kelly (2010), Robinson et al. (2010)]. However, in the context of scRNA-seq data, it is difficult to tease apart true differing clusters or variability due to confounding factors such as cell cycle. Further, the idea of cell type within the same tissue is currently a poorly defined biological concept, and it is unknown whether transcriptional differences represent true subpopulations [Patel et al. (2014)].

Establishing gene-regulatory networks are often done in bulk RNA-seq data by grouping genes which are considered to be 'co-regulated'. This used to be done using correlations between genes or by using clustering techniques mentioned previously. A more sophisticated technique fits regulatory networks into the framework of a GGM by estimating the gene-gene precision matrix. Bayesian approaches have been used to combine prior biological knowledge with gene expression data [Werhli and Husmeier (2007)]. Clustering approaches were used to separate out clusters of genes first, and then estimated individual DAGs within each cluster [Yavari et al. (2008)]. The SGS

algorithm has been incorporated into the estimation to reduce computation time [Yang et al. (2011)]. For even faster computation, the graphical lasso, a penalized Gaussian covariance estimation method has been developed and used in a large number of network reconstructions [Friedman et al. (2008)]. Within the context of scRNA-seq data, we can take advantage of the variability of gene expression between cells as a natural perturbation to infer gene-regulatory networks [Padovan-Merhar and Raj (2013), Segal et al. (2003), PeâĂŹer et al. (2001)].

Currently, the only graphical model proposed specifically for scRNA-seq data has come from [McDavid et al. (2016)]. They propose a penalized multivariate Hurdle model based on the exponential family for neighborhood selection. Consider variable y with its element-wise non-zero indicator variable v_y. To fit this data into the multivariate Hurdle model framework, they let $f(y)$ have Normal density of mean ξ and precision τ^2. Then, they excise zero from the support of $f(y)$ and give it a density of $f_0(y) = \exp\{v_y[1/2\log(\tau^2/(2\pi)) + \log(p/(1-p)) - \xi^2\tau^2/2] + y\xi\tau^2 - \mathbf{y}^2\tau^2/2 + \log(1-p)\}$. This allows the full support of y to be modeled under the exponential family while modeling the extraeneous zero expression. They simplify the form of the multivariate Hurdle model and give the following likelihood:

$$\log f(\mathbf{y};\theta) = \mathbf{v_y}^\top \mathbf{G}\mathbf{v_y} + \mathbf{v_y}^\top \mathbf{H}\mathbf{y} - \frac{1}{2}\mathbf{y}^\top \mathbf{K}\mathbf{y} - C(\mathbf{G},\mathbf{H},\mathbf{K}) \tag{1.2}$$

where $\mathbf{G}, \mathbf{H}$ and $\mathbf{K}$ are interaction parameter matrices and $C(\mathbf{G},\mathbf{H},\mathbf{K})$ is a constant. Then each part of the likelihood pertains to the modeling of a specific contribution, where $\mathbf{G}$ parameterizes the interaction of the binary process, $\mathbf{K}$ parameterizes the interaction of the continuous process, and $\mathbf{H}$ parameterizes the interaction between binary and continuous process.

For variable selection, they impose an L_1 group penalty of the form $\lambda \sum_a \sqrt{\theta_a^\top \mathbf{H}_{aa}\theta_a}$ where θ_a are the parameters of interest across $\mathbf{G}, \mathbf{H}$, and $\mathbf{K}$ corresponding to gene a.

Typically this type of group penalty has been seen with $\mathbf{H}_{aa} = I$. However, this type of group penalty, which causes all parameters in θ_a to vanish simultaneously, assumes similar effect size and scale-equivariance across the parameters. This is distinctly not true in this case due to the different underlying model distributions for the indicator variable $\mathbf{v_y}$ (binomial) and continuous variable $\mathbf{y}$ (Gaussian). The typical solution of scaling the design matrix is not an option in this case due to the different distribution assumption. Hence, the authors propose rescaling the estimated coefficients within the penalty with the Fisher information under the null model of $\theta_a = 0$ as $\mathbf{H}_{aa}$ and show that it is equivalent to a score test of the null hypothesis that $\theta = 0$ versus the alternative of $\theta \neq 0$.

1.2.4 Summary

Forming regulatory networks with scRNA-seq allows for insights which are hidden in bulk RNA-seq data. For example, if two independent transcription factors activate a set of genes, in bulk RNA-seq data, this would look like they were co-expressed or they regulate each other [Stegle et al. (2015)]. Currently, there has only been one model developed specifically for scRNA-seq data and methods used for bulk RNA-seq data have largely depended upon undirected gaussian graphical models. While computationally efficient, they are not suitable for count data, and they do not take into consideration aspects of scRNA-seq data such as zero-inflation, relying instead on quality control protocols. The hurdle model by [McDavid et al. (2016)] assumes all forms of zero expression comes from the same data-generating process and assumes the non-zero gene expression data is Gaussian. Our proposal aims to address these issues by developing a graphical model algorithm which uses a joint penalized zero-inflated negative binomial model for neighborhood selection.

CHAPTER 2: MODEL-FREE ESTIMATION

Recall that we denote a DAG skeleton by $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V} = \{X_1, ..., X_p\}$ and $\mathbf{E}$ are the vertex set and the edge set, respectively. A moral graph $\mathcal{G}_{\mathcal{M}} = (\mathbf{V}, \mathbf{E}_{\mathcal{M}})$ can be derived from a skeleton $\mathcal{G}$ by connecting the co-parents in all the v-structures in $\mathcal{G}$. Our goal is to develop a model free and nonparametric method to estimate DAG skeleton by a two step process. First, we estimate the moral graph using a penalized model-free regression. Then, we further thin the edges of the estimated DAG skeleton by removing false connections due to v-structures using the modified PC-algorithm with nonparametric conditional independence testing. The overall structure of the algorithm is presented below.

2.1 Algorithm

2.1.1 Step 1: Estimation of the Moral Graph

We employ a neighborhood selection approach to estimate the moral graph by estimating the neighbors of each vertex in the moral graph separately, and then consolidating the results. That is, we establish an edge $X_i - X_j$ if either X_j is selected as a neighbor of X_i or X_i is selected as a neighbor of X_j. All the neighbors of a vertex X_k in the moral graph form its Markov Blanket. For neighborhood selection of X_k, we adopt a model free variable selection method (multivariate group-wise adaptive penalization or **mGAP**) for X_k against all the other variables, denoted by X_{-k} [Sun and Li (2012)]. mGAP uses a transformation that expands the one dimensional response variable into several dimensions, such as a spline transformation or a transformation resembling

Sliced Inverse Regression (SIR) [Li (1991)]. Then, it performs variable selection and parameter estimation by minimizing a weighted multivariate L_2 loss function with an adaptive group lasso penalty. Specifically, for response vertex X_k, it uses the following objective function for some transformation function $r(\cdot)$ with h-dimension output:

$$\text{argmin}_{\mathbf{B},\kappa,\omega}\left\{\sum_{s=1}^{h}\frac{\|r(X_k)_s-\mathbf{X}_{-k}B_s\|^2}{\omega_s}+\lambda\sum_{j=1}^{p}\frac{\|\mathbf{b}_j\|+\tau}{\kappa_j}+\lambda\sum_{j=1}^{p}\log(\kappa_j)+n\sum_{s=1}^{h}\log(\omega_s)\right\}, \tag{2.1}$$

where $r(\cdot)_s$ is the s^{th} dimension of $r(\cdot)$, B_s is the s-th column of coefficient matrix B, which includes the $p-1$ regression coefficients for the regression with $r(X_k)_s$ as response variable, λ and τ are tuning parameters, and $\|\mathbf{b}_j\| = \sqrt{\Sigma_{s=1}^{h}B_{j,s}^2}$ is the L_2 norm for the h coefficients of the j-th covariate. ω_s's and κ_j's are the weights for the objective function and the penalty term, and the last two terms in equation (2.1) add constraints on the sizes of ω_s's and κ_j's. Then, the set of vertices associated with X_k are $\{X_j\ ;\ \|\hat{\mathbf{b}}_j\|>0\}$. Estimation is done using a coordinate descent algorithm.

The pseudo-code of the algorithm for estimating the moral graph is then:

Algorithm 1: Moral Graph Estimation

Data: X

Result: $\mathcal{G}_{\mathcal{M}} = (\mathbf{V}, \mathbf{E})$

for $j = 1$ **to** p **do**

- Let $y = X_k$ and $X = X_{-k}$.
- Scale y and X to standard deviation of 1 and center to 0.
- $y_t \leftarrow r(y)$.
- Obtain tuning parameter grid λ, τ from Algorithm 4 in *Appendix A*.
- Select variables among X associated with y by mGAP, denoted by $\mathcal{E}$.
- Create edges $E_{k,j}$ and $E_{j,k}$, where $X_j \in \mathcal{E}$. Here $E_{k,j} \equiv E_{j,k}$ for the undirected graph.

2.1.2 Step 2: Estimation of the Skeleton

The modified-PC algorithm attempts to further thin the estimated moral graph to obtain an estimate for the skeleton by first testing for marginal independence then conditional independence while conditioning on all possible candidate d-separation sets. Testing is done only on connected pairs of the estimated moral graph. For a nonparametric conditional independence test, we use a kernel conditional independence test (KCI-test) proposed by [Zhang et al. (2012)]. First, we present the characterization of conditional independence utilized by Zhang:

Notation and Set up

Consider vertex X_j with Markov Blanket $\mathbf{X}_j^{MB} := \text{adj}(\mathcal{G}_{\mathcal{M}}, X_j)$, where $\text{adj}(\mathcal{G}_{\mathcal{M}}, X_j)$ denotes the neighbors of X_j in moral graph $\mathcal{G}_{\mathcal{M}}$. Then consider the conditional independence test of X_j and $X_k \in \mathbf{X}_j^{MB}$, conditioned on a subset of the variables of $\mathbf{X}_j^{MB}$, denoted by $X_{\mathbf{s}}$. Let $\ddot{X} := (X_j, X_{\mathbf{s}})$, $Y := X_k$, and $Z := X_{\mathbf{s}}$. We denote the kernel functions of the three variables by $k_{\ddot{X}}$, k_Y, and k_Z, and the corresponding reproducing kernel Hilbert spaces (RKHS) by $\mathcal{H}_{\ddot{X}}$, $\mathcal{H}_Y$, and $\mathcal{H}_Z$, respectively. A kernel function characterizes the similarity of any two samples with respect to one variable (set). For example, suppose there are n samples of Z, a Gaussian kernel defines the similarity between the u-th and the v-th sample as $k_Z(\mathbf{z}_u, \mathbf{z}_v) = \exp\{-\|\mathbf{z}_u - \mathbf{z}_v\|^2/(2\sigma^2)\}$, where σ^2 is the prespecified kernel width. A kernel matrix for n samples is an $n \times n$ matrix, with the $(u, v)^{\text{th}}$ entry being defined by the kernel function on the u-th and the v-th observations. We denote the kernel matrices for these three variables as $\mathbf{K}_{\ddot{X}}$, $\mathbf{K}_Y$, and $\mathbf{K}_Z$, respectively. The centralized kernel matrix for $\ddot{X}$ is defined by $\tilde{\mathbf{K}}_{\ddot{X}} := (\mathbf{I} - \frac{1}{n}JJ^\intercal)\mathbf{K}_{\ddot{X}}(\mathbf{I} - \frac{1}{n}JJ^\intercal)$, where $\mathbf{I}$ is the identity matrix and J is a vector of 1's. Analogously, we can define the centralized kernel matrices for Y and Z: $\tilde{\mathbf{K}}_Y, \tilde{\mathbf{K}}_Z$.

Lemma 1 from [Daudin (1980)]

Denote the space of all square integrable functions of X (i.e., any $f(x)$ such that $\int_{-\infty}^{\infty} |f(x)|^2 dx < \infty$) by L_X^2. Following the notation from [Zhang et al. (2012)], consider the following constrained L^2 spaces:

$$\begin{aligned} \mathcal{E}_{\ddot{X}Z} &:= \{\tilde{f} \mid \tilde{f} = f(\ddot{X}) - \mathbb{E}[f|Z], f \in L^2_{\ddot{X}Z}\}, \\ \mathcal{E}'_{YZ} &:= \{\tilde{g}' \mid \tilde{g}' = g'(Y) - \mathbb{E}(g'|Z), g' \in L^2_Y\}. \end{aligned}$$

Then, the following conclusions are equivalent:

(1) $X \perp Y|Z$.

(2) $\mathbb{E}[\tilde{f}\tilde{g}'] = 0 \ \forall \ \tilde{f} \in \mathcal{E}_{\ddot{X}Z}$ and $\tilde{g}' \in \mathcal{E}'_{YZ}$.

Here we use notation g' instead of g to be consistent with the notation used by [Zhang et al. (2012)]. Let $h^*_f(Z) \equiv \mathbb{E}[f|Z]$, where $h^*_f(Z) \in L^2_Z$ is the regression function of $f(\ddot{X})$ on Z. $h^*_f(Z)$ can then be estimated with kernel ridge regression: $h^*_f(Z) = \tilde{\mathbf{K}}_Z(\tilde{\mathbf{K}}_Z + \epsilon I)^{-1} f(\ddot{X})$, with some regularization parameter ϵ. Let $h^*_{g'}(Z) \equiv E(g'|Z)$. Then $h^*_{g'}(Z) \in L^2_Z$ can be estimated similarly: $h^*_{g'}(Z) = \tilde{\mathbf{K}}_Z(\tilde{\mathbf{K}}_Z + \epsilon I)^{-1} g'(Z)$. Let $\mathbf{R}_Z = I - \tilde{\mathbf{K}}_Z(\tilde{\mathbf{K}}_Z + \epsilon I)^{-1} = \epsilon(\tilde{\mathbf{K}}_Z + \epsilon I)^{-1}$. Then the centralized kernel matrices corresponding to $\tilde{f}(\ddot{X})$ and $\tilde{g}'$ are $\tilde{\mathbf{K}}_{\ddot{X}|Z} = \mathbf{R}_Z \tilde{\mathbf{K}}_{\ddot{X}} \mathbf{R}_Z$ and $\tilde{\mathbf{K}}_{Y|Z} = \mathbf{R}_Z \tilde{\mathbf{K}}_Y \mathbf{R}_Z$, respectively. Let $\tilde{\mathbf{K}}_{\ddot{X}|Z} = \boldsymbol{\psi}_{\ddot{\mathbf{x}}|\mathbf{z}} \boldsymbol{\psi}^{\intercal}_{\ddot{\mathbf{x}}|\mathbf{z}}$ and $\tilde{\mathbf{K}}_{Y|Z} = \boldsymbol{\phi}_{\mathbf{y}|\mathbf{z}} \boldsymbol{\phi}^{\intercal}_{\mathbf{y}|\mathbf{z}}$. More specifically, $\boldsymbol{\psi}_{\ddot{\mathbf{x}}|\mathbf{z}}$ (and simlarly $\boldsymbol{\phi}_{\mathbf{y}|\mathbf{z}}$) can be derived by eigen-value decomposition of $\tilde{\mathbf{K}}_{\ddot{X}|Z} = V \Lambda V^{\intercal}$ such as $\boldsymbol{\psi}_{\ddot{\mathbf{x}}|\mathbf{z}} = V \Lambda^{1/2}$.

Lemma 2 from Proposition 5 of [Zhang et al. (2012)]

Under the null hypothesis H_0: X and Y are conditionally independent given Z, we have that the statistic:

$$T_{CI} := \frac{1}{n} Tr(\tilde{\mathbf{K}}_{\ddot{X}|Z} \tilde{\mathbf{K}}_{Y|Z}) \tag{2.2}$$

has the same asymptotic distribution as:

$$\check{T}_{CI} := \frac{1}{n}\sum_{k=1}^{n^2} \lambda_k \cdot z_k^2, \tag{2.3}$$

where λ_k are eigenvalues of $\check{\mathbf{w}}\check{\mathbf{w}}^\intercal$ and $\check{\mathbf{w}} = [\check{\mathbf{w}}_1, ..., \check{\mathbf{w}}_n]$, with the vector $\check{\mathbf{w}}_t$ obtained by stacking $\check{\mathbf{M}}_t = \psi^\intercal_{\ddot{\mathbf{x}}|\mathbf{z},t}\phi_{\mathbf{y}|\mathbf{z},t}$, where $\psi_{\ddot{\mathbf{x}}|\mathbf{z},t}$ and $\phi_{\mathbf{y}|\mathbf{z},t}$ denote the t-th row of the corresponding matrics.

The authors use a Monte Carlo simulation to approximate the null distribution. Instead, we chose to use an exact method, [Imhof (1961)]. This is further expanded upon in Section 3.4.4.

To evaluate the conditional independence of X_i and X_j, we do not need to search across all possible conditional sets. Instead, we will follow the procedure used by PenPC [Ha et al. (2016b)] to select conditional sets. For completeness, we briefly describe our procedure below, starting by defining a few notations. Let $\mathcal{G}$ be an arbitrary Markov graph.

- $A_{\mathcal{G},i,j} = \text{adj}(\mathcal{G}, X_i) \bigcup \text{adj}(\mathcal{G}, X_j) \backslash \{X_i, X_j\}$, i.e., the Markov Blanket of X_i and X_j. Recall that $\text{adj}(\mathcal{G}, X_j)$ denotes the neighbors of X_j in graph $\mathcal{G}$.

- $B_{\mathcal{G},i,j} = \text{adj}(\mathcal{G}, X_i) \bigcap \text{adj}(\mathcal{G}, X_j) \backslash \{X_i, X_j\}$, which includes all potential common children of X_i and X_j.

- $C_{\mathcal{G},i,j} = A_{\mathcal{G},i,j} \bigcap (B_{\mathcal{G},i,j} \bigcup Con_{\mathcal{G}}(B_{\mathcal{G},i,j}))$, where $Con_{\mathcal{G}}(B_{\mathcal{G},i,j})$ denotes the vertices that are connected to $B_{\mathcal{G},i,j}$. $C_{\mathcal{G},i,j}$ includes any possible common descendants of X_i and X_j within the Markov Blanket of X_i and X_j.

- $\mathbf{\Pi}_{i,j} = \{D_{\mathcal{G},i,j} : A_{\mathcal{G},i,j} \backslash \tilde{C}_{\mathcal{G},i,j}, \tilde{C}_{\mathcal{G},i,j} \subseteq C_{\mathcal{G},i,j}\}$. At least one of the set in $\mathbf{\Pi}_{i,j}$ includes all common parents of X_i and X_j, but excludes any common descendants. In other words, if there is any d-separation set of X_i and X_j, it will be included in $\mathbf{\Pi}_{i,j}$.

Before applying conditional independence testing, we perform marginal independence testing for all connected nodes using Hoeffding's test of independence. Hoeffding's test is a non-parametric test for bivariate independence. Denote the two variables of interest by X and Y. Let $F_{X,Y}$ be the joint distribution of X and Y, and let F_X and F_Y be their marginal distribution functions. The motivation behind Hoeffding's test starts with the notion that if and only if X and Y are independent, then $D(x,y) = F_{X,Y}(x,y) - F_X(x)F_Y(y) = 0$. Then, across the full sample the quantity is summarized as $\int D^2(x,y)dF(x,y)$ which is estimated with the statistic $D_n = \frac{Q-2(n-2)R+(n-2)(n-3)S}{n(n-1)(n-2)(n-3)(n-4)}$; where $Q = \Sigma_{i=1}^{n}(R_i-1)(R_i-2)(S_i-1)(S_i-2)$, $R = \Sigma_{i=1}^{n}(R_i-2)(S_i-2)c_i$, and $S = \Sigma_{i=1}^{n}(c_i-1)c_i$. R_i and S_i are the ranks of X_i and Y_i respectively, and c_i is is the number of bivariate observations smaller than both X_i and Y_i. [Hoeffding (1948), Wilding and Mudholkar (2008)]

Based on the aforementioned approach, the modified-PC algorithm estimates the skeleton by performing conditional independence tests for each pair of connected variables X_i and X_j, conditioning any subset of $\mathbf{\Pi}_{i,j}$. Full pseudo-code of the algorithm is then:

Algorithm 2: Modified-PC Algorithm

Data: $\mathcal{G}_{\mathcal{M}} = (\mathbf{V}, \mathbf{E})$, $\mathbf{X}$

Result: $\mathcal{G}$

foreach $(\mathbf{E}_{i,j}) \in \mathbf{E}$ **do**

  if $X_i \perp X_j$ **then** ; `/* Hoeffding's test of independence. */`

    Remove $E_{i,j}$ and $E_{j,i}$.

$l = -1$

repeat

  $l = l + 1$

  $\tilde{\mathcal{G}} = \mathcal{G}$

  foreach $(i, j); |D_{\tilde{\mathcal{G}},i,j}| \geq l$ **do**

    foreach $[\Gamma; \Gamma \subseteq D_{\tilde{\mathcal{G}},i,j}, |\Gamma| = l]$ **do**

      if $X_i \perp X_j | X_\Gamma$ **then** ; `/* KCI-test */`

        Remove $E_{i,j}$ and $E_{j,i}$.

        Exit while loop and move to next $E_{i,j}$ in for each loop.

until $\max_{i,j} |D_{\tilde{\mathcal{G}},i,j}| < l$;

2.2 Theoretical Properties

Condition A (Causal Sufficiency) $\forall X_j \in \mathbf{V}$, the set of all causes (or parents) of X_j are also a subset of $\mathbf{V}$.

Condition B (Faithfulness) Let the distribution of $\mathbf{V}$ be faithful to the associated graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$.

Condition A requires that all relevant parts of the model information be contained in $\mathbf{V}$. Condition B is a core component of identifiability for DAG skeleton estimation

using conditional independence methods. These two conditions are needed for identifiability of the problem. We argue that condition A is reasonable in our application on gene expression studies because we analyze genome-wide gene expression data, which gives a comprehensive molecular portrait of the underlying molecular system. The faithfulness assumption is commonly used and we refer to [Zhang and Spirtes (2016)] for more details.

Step 1 - mGAP Variable Selection

Condition 1.1

$\log(p) = O(n^{\alpha})$ and $d_0 = O(n^{\nu})$ where $0 \leq \alpha < 1$, $0 \leq \nu < \frac{1}{2}$. This condition puts bounds on the dimensionality (p) as well as the number of causal variables (d_0). The latter is effectively a condition of sparsity.

Condition 1.2

Denote d_n as half of the smallest effect size for the regression coefficient matrix in the model free penalized regression, given the regression coefficient is non-zero. Then $d_n \equiv O(n^{-\gamma_0}(\log n)^{\frac{1}{2}})$ for some $\gamma_0 \in (\nu, \frac{1}{2})$.

In a general problem with arbitrary penalty, there are additional conditions imposed on the penalty function. These conditions are satisfied by the penalty function that we use for mGAP. We defer the details to the proof for Theorem 2 in the Appendix.

For the consistency of Step 2 - Conditional Independence Testing, we need conditions A and B, as well as the following condition on minimum effect size:

Condition 2.1

If $X_i \not\perp X_j | X_{\mathbf{s}}$, we assume a lower bound on the expectation of the test-statistic: $\inf_{i,j|\mathbf{s}} E(T_{CI}) \geq c_n$, where $c_n = O(n^{\frac{1}{2}-d})$ and $0 < d < 1/2$.

Theorem 2. Given conditions 1.1 to 1.3, with probability at least $P_{\texttt{converge}} = 1 - 2\left[d_0 n^{-1} + (p - d_0)\exp\left(-n^{\alpha}\log n\right)\right]$, there exists an estimator to minimize the penalized objective function $\hat{\mathbf{B}} = (\hat{B}_1, ..., \hat{B}_h)$. For each $\hat{B}_s$, with $s = 1, ..., h$, we can partition the coefficients into two parts, $\hat{B}_s = (\hat{B}_{1,s}^{\intercal}, \hat{B}_{2,s}^{\intercal})^{\intercal}$, where the true values of $\hat{B}_{1,s}$ are non-zero and the true values of $\hat{B}_{2,s}$ are 0.

(1) Sparsity: $P(\hat{B}_{2,s} = 0) \to 1 \; \forall \, s$

(2) L_∞ loss: $\|\hat{B}_{1,s} - B_{1(0),s}\|_\infty = o_p(n^{-\gamma_0}\sqrt{\log n}) \; \forall \, s$.

Corollary 1. Joint variables found by Step 1 of the algorithm contains the true Markov blanket of a vertex with probability of at least $P = 1 - 2[d_0 n^{-1} + (p - d_0)\exp\left(-n^{\alpha}\log n\right)]$.

This follows directly from the results of Theorem 2.

Lemma 4.

(1) Recall that $\mathcal{G}$ denotes the skeleton of a DAG, and $\mathcal{G}_{\mathcal{M}}$ denotes corresponding moral graph. The set of edges $\mathbf{E}_{\mathcal{M}}$ of $\mathcal{G}_{\mathcal{M}}$ includes all the edges $\mathbf{E}$ of $\mathcal{G}$ plus the edges between co-parents of v-structures. If $X_i - X_j \in \mathbf{E}_{\mathcal{M}}$ but $X_i - X_j \notin \mathbf{E}$, then there exists a subset of $A_{\mathcal{G},i,j}$ which d-separates vertices X_i and X_j in $\mathcal{G}$.

(2) $\mathbf{\Pi}_{i,j} = \{A_{\mathcal{G},i,j} \backslash D_{\mathcal{G},i,j}, D_{\mathcal{G},i,j} \subseteq C_{\mathcal{G},i,j}\}$ contains at least one such subset.

Lemma 4 follows from Conditions A and B, as well as the way that $\mathbf{\Pi}_{i,j}$ is constructed. The following Lemma 4 is a technical condition needed to prove the consistency of conditional independence testing.

Lemma 8. For any $\gamma > 0$, $\sup_{i,j,\mathbf{s}\in\mathbf{\Pi}_{i,j}} \mathbf{P}\left[|T_{CI} - \mathbb{E}(T_{CI})| > \gamma\right] \leq \exp(-2n\gamma^2/R^4)$, where R^2 is the largest possible element of $\tilde{\mathbf{K}}_{X_{i,\mathbf{s}}|X_\mathbf{s}}\tilde{\mathbf{K}}_{X_j|X_\mathbf{s}}$.

Proof can be found in the Appendix.

Theorem 4. Assume a perfect estimation of $\mathcal{G}_\mathcal{M}$ in Step 1 as well as conditions (1.1) and (2.3). Let the estimate from Step 2 be $\hat{\mathcal{G}}^{\alpha_n}_{\text{skel},n}$, where α_n is the significance level used in the conditional independence testing for Step 2. Then, there exists an $\alpha_n \to_{n\to\infty} 0$ such that:

$$\mathbf{P}[\hat{\mathcal{G}}_{\text{skel},n}(\alpha_n) = \mathcal{G}_{\text{skel},n}] = 1 - O\left(\exp(n^\alpha - C(n^{2(1-d)})\right) \to_{n\to\infty} 1$$

for some constant $C > 0$.

Corollary 2. With sample estimation of $\mathcal{G}_\mathcal{M}$ and conditional independence test statistic convergence of $O(n)$ with p-value threshold α_n, the combined error rate of Step 1 and 2 converged to 0 as $n \to \infty$.

2.3 Implementation Considerations

2.3.1 mGAP

There are two tuning parameters for mGAP estimation: λ and τ. We need to establish the grid to search for (λ, τ). The grid is selected so that it covers the situation from no penalization (all variables selected) to full penalization (no variables selected). Larger λ and smaller τ leads to a stronger penalty and smaller λ and larger τ leads to

a weaker penalty. Following [Sun and Li (2012)], we set the range of τ as $10^{-6} \leq \tau \leq 1$, and then set the range of λ to be $0 \leq \lambda \leq \max\left(\lambda_1^{(\max)}, ..., \lambda_p^{(\max)}\right)$, where $\lambda_j^{(\max)} = \left\| \left[\frac{(2\tau(X_j^\intercal X_j)^{-1} X_j^\intercal T_1)}{\|T_1\|^2/(n\|X_j\|^2)}, ..., \frac{(2\tau(X_j^\intercal X_j)^{-1} X_j^\intercal T_h)}{\|T_h\|^2/(n\|X_j\|^2)} \right]^\intercal \right\|^2$, and T_s denotes the s-th transformation of the response variable. In all the simulations and real data studies of this dissertation, we employ the cubic spline with one inner knot, which has been shown to have comparable or better performance than other transformations [Sun and Li (2012)].

Depending on the degree of penalty from the tuning parameters, a range of models from size 0 to size $p-1$ is selected. To be able to find the best fitting model, we use a pre-defined scoring function. [Sun and Li (2012)] use the traditional BIC. However, it has been shown that traditional BIC tend to select models with a large number of false positives in high dimensional settings. We found this to be the case in our simulation studies and thus we explore two forms of the extended BIC in Section 2.4. Algorithm pseudo-code can be found in Supplementary Materials Algorithm 1.

2.3.2 Estimation of Null Distribution for KCI-test

In the original formulation of KCI-test, [Zhang et al. (2012)] use a Monte Carlo simulation to find the null distribution of the test statistic, which is distributed as the weighted summation of central χ_1^2 random variables. To improve accuracy as well as computational efficiency, we choose to use Imhof's exact method [Imhof (1961)] to quantify null distribution. We compared Imhof's exact method with the Monte Carlo by simulating two sets of 500 variable pairs (X, Y). In the first set, X and Y are correlated but are conditionally independent given one-dimensional variable Z; in the second set, X and Y are correlated and conditionally dependent given one-dimensional variable Z. For each pair, we estimated the p-value for KCI-test using Imhof's exact method with an error bound of 2.22×10^{-16} and Monte Carlo then terminating at 100, 1000, 5000, 10000 iterations. In addition, we benchmarked the performance of each

estimate 75 times.

For the correlated pairs, we found that on average the median runtime for a single test was 30.3ms when using Imhof's exact method, which fell between the median runtimes of the Monte Carlo method between 1000 iterations (12.6ms) and 5000 iterations (58.8ms). Similarly, for independent pairs, the average median runtime of Imhof's exact method was 17.9ms, which fell between the median runtimes of the Monte Carlo method between 1000 iterations (11.4ms) and 5000 iterations (56.3ms). Overall, Imhof's exact method had much smaller ranges for runtimes (*e.g.* between 30.0ms and 32.1ms for correlated pairs) while Monte Carlo's ranges were much wider (*e.g.* between 58.1ms and 188.5ms for correlated pairs at 5000 iterations). Figure F2 illustrates the difference in $-\log_{10}$(P-Values) between the two methods at each iteration range. While conditionally independent pairs show very little difference between the two methods, conditionally dependent pairs show that large deviations may be found when we're dealing with the tail-end of the distribution even at 5000 iterations of the Monte Carlo. Therefore, we choose to use Imhof's method because it has advantages in terms of both accuracy and computational efficiency than the Monte Carlo method.

2.4 Simulation

The simulations aimed to evaluate the performance of our model free algorithm versus *PenPC*, which assumes multivariate Gaussian distribution. *PenPC* was applied using the R package `PenPC`. Similarly to [Kalisch and BÃijhlmann (2007)] and [Ha et al. (2016b)], we simulated the base graph structure using the Erdős and Rényi (ER) model where we connect the vertices randomly with equal probability [ErdËİos and RÂťenyi (1959)]. Specifically, the probability that any two vertices are connected is $p_E = d_0/p$. Then, all graphs with p vertices and d_0 edges have probability of $p_E^{d_0}(1-p_E)^{\binom{p}{2}-d_0}$ to be generated. For these particular simulations, we used $n = 100$, $p = 100$, and $d_0 = 1$.

Table 2.1: Simulation Results using (2.4) with Effect Size 1

$h(x)$	Method	$\mathcal{G}_{\mathcal{M}}$ Estimate		$\mathcal{G}$ Estimate			
		FDR	FRR	FDR	Type 1 Error	Power	H Dist
x	PenPC	0.3264	0.0001	0.1184	0.0014	0.9880	14.78
	ModelFree	0.1131	0.0002	0.0415	0.0004	0.9706	7.38
	PenPC GG	0.0026	0.0002	0.0010	0.0000	0.9851	1.68
	ModelFree GG	0.0180	0.0004	0.0064	0.0001	0.9618	4.68
x^2	PenPC	0.7599	0.0154	0.6611	0.0050	0.2425	126.02
	ModelFree	0.3934	0.0047	0.1616	0.0008	0.3813	71.34
	PenPC GG	0.4999	0.0166	0.4677	0.0019	0.2042	99.66
	ModelFree GG	0.3423	0.0054	0.1087	0.0005	0.3961	67.22
x^3	PenPC	0.5476	0.0068	0.3870	0.0042	0.6286	78.96
	ModelFree	0.3891	0.0033	0.0452	0.0003	0.6259	41.88
	PenPC GG	0.3478	0.0073	0.2830	0.0026	0.6153	64.68
	ModelFree GG	0.3556	0.0034	0.0306	0.0002	0.6251	41.02
e^x	PenPC	0.5133	0.0049	0.3229	0.0036	0.7219	63.94
	ModelFree	0.3948	0.0015	0.0564	0.0005	0.8227	23.50
	PenPC GG	0.2673	0.0063	0.2102	0.0019	0.6760	51.92
	ModelFree GG	0.3518	0.0016	0.0436	0.0004	0.8223	22.30

FDR = False Discovery Rate; FRR = False Recovery Rate; H Dist = Hamming's Distance.
GG = Indicates the use of the Gaussian Graphical Extended BIC.
Note: Step 1 FDR is calculated against the moral graph and Final FDR is calculated against the true graph.

Once the graph structure is generated, the underlying data structure is assumed to be one of two linear structural equation models:

$$\mathbf{X} = \mathbf{B}^{\intercal} h(\mathbf{X}) + \mathbf{e}. \tag{2.4}$$

$$\mathbf{X} = h(\mathbf{B}^{\intercal}\mathbf{X} + \mathbf{e}). \tag{2.5}$$

where, $\mathbf{e} \sim N(0, \sigma^2 I_{n\times n})$. Without loss of generality, let $\mathbf{B}$ be an upper triangular matrix which implicitly enforces an ordering to the variables such that all parents have a smaller index than their children. Finally, let $h(\cdot)$ be a transformation function applied over each column of $\mathbf{X}$.

Table 2.2: Simulation Results using (2.5) with Effect Size 1

$h(x)$	Method	$\mathcal{G}_{\mathcal{M}}$ Estimate		$\mathcal{G}$ Estimate			
		FDR	FRR	FDR	Type 1 Error	Power	H Dist
x	PenPC	0.3264	0.0001	0.1184	0.0014	0.9880	14.78
	ModelFree	0.1131	0.0002	0.0415	0.0004	0.9706	7.38
	PenPC GG	0.0026	0.0002	0.0010	0.0000	0.9851	1.68
	ModelFree GG	0.0180	0.0004	0.0064	0.0001	0.9618	4.68
x^2	PenPC	0.7653	0.0157	0.6629	0.0049	0.2352	125.86
	ModelFree	0.4561	0.0043	0.0631	0.0003	0.4246	62.26
	PenPC GG	0.5023	0.0166	0.4632	0.0019	0.2011	99.46
	ModelFree GG	0.3627	0.0056	0.0344	0.0002	0.4322	60.18
x^3	PenPC	0.5585	0.0069	0.3966	0.0043	0.6198	80.82
	ModelFree	0.7030	0.0030	0.0295	0.0002	0.6196	41.56
	PenPC GG	0.3618	0.0075	0.2907	0.0026	0.6039	66.18
	ModelFree GG	0.5729	0.0036	0.0175	0.0001	0.6070	42.10
e^x	PenPC	0.5290	0.0060	0.3357	0.0036	0.6766	68.50
	ModelFree	0.6575	0.0009	0.0472	0.0004	0.8181	23.14
	PenPC GG	0.2720	0.0070	0.2186	0.0019	0.6359	55.92
	ModelFree GG	0.5451	0.0013	0.0288	0.0003	0.8032	22.98

FDR = False Discovery Rate; FRR = False Recovery Rate; H Dist = Hamming's Distance
GG = Indicates the use of the Gaussian Graphical Extended BIC, rather than regular BIC.
Note that Step 1 FDR is calculated against the moral graph and Final FDR is calculated against the true graph.

Results for the simulation with 100 iterations and conditional independence testing threshold $\alpha = 0.01$ can be found in *Table 2.1* through *Table 2.4*. Transformation $h(x) = x^2$ is a particular challenging case since it is not a monotone transformation. We included it to show that Model Free still performs adequately in this case, with substantial improvements in comparison to *PenPC*. Hamming's Distance show a 30% to 60% improvement when using Model Free over *PenPC*. Overall, if tuning parameters are selected by Gaussian Graphical Extended BIC [Foygel and Drton (2010)], both *PenPC* and Model Free performed better than their counterparts if we consider false negatives and false positives equally, trading a substantially lower false discovery rate (FDR) for a slight decrease in power. In *Table 2.2*, we can see that the change in

Table 2.3: Simulation Results using (2.4) and Effect Size 0.5

$h(x)$	Method	$\mathcal{G}_{\mathcal{M}}$ Estimate		$\mathcal{G}$ Estimate			
		FDR	FRR	FDR	Type 1 Error	Power	H Dist
x	PenPC	0.4308	0.0008	0.2360	0.0029	0.9015	38.62
	ModelFree	0.1324	0.0066	0.0684	0.0005	0.6425	41.50
	PenPC GG	0.0022	0.0125	0.0019	0.0000	0.4059	60.96
	ModelFree GG	0.0285	0.0116	0.0188	0.0001	0.4381	58.50
x^2	PenPC	0.7791	0.0166	0.7048	0.0048	0.1865	129.63
	ModelFree	0.2878	0.0099	0.1938	0.0008	0.2970	80.00
	PenPC GG	0.2524	0.0191	0.2287	0.0003	0.0903	96.53
	ModelFree GG	0.1801	0.0111	0.1050	0.0004	0.3112	74.98
x^3	PenPC	0.5109	0.0045	0.3129	0.0035	0.7346	61.80
	ModelFree	0.3373	0.0030	0.0602	0.0004	0.6273	42.84
	PenPC GG	0.2545	0.0062	0.2031	0.0018	0.6800	50.92
	ModelFree GG	0.3039	0.0034	0.0400	0.0003	0.6199	42.10
e^x	PenPC	0.4918	0.0034	0.2837	0.0032	0.7759	54.60
	ModelFree	0.3012	0.0024	0.0954	0.0008	0.7263	36.04
	PenPC GG	0.1438	0.0067	0.1134	0.0009	0.6607	43.84
	ModelFree GG	0.2410	0.0034	0.0667	0.0005	0.6963	36.44

FDR = False Discovery Rate; FRR = False Recovery Rate; H Dist = Hamming's Distance
GG = Indicates the use of the Gaussian Graphical Extended BIC, rather than regular BIC.
Note that Step 1 FDR is calculated against the moral graph and Final FDR is calculated against the true graph.

underlying data structure does not substantially effect the performance of Model-Free.

While Model Free performs better than *PenPC* in all cases in *Tables 2.1* and *2.2*, we find that Model Free's primary drawback is a large performance drop with low effect sizes as can be seen in *Tables 2.3* and *2.4*. While for non-linear cases, Model Free still does substantially better, by at least 20% in terms of Hamming's Distance, in the linear case PenPC has better performance.

Table 2.4: Simulation Results using (2.5) and Effect Size 0.5

$h(x)$	Method	$\mathcal{G}_{\mathcal{M}}$ Estimate		$\mathcal{G}$ Estimate			
		FDR	FRR	FDR	Type 1 Error	Power	H Dist
x	PenPC	0.4308	0.0008	0.2360	0.0029	0.9015	38.62
	ModelFree	0.1324	0.0066	0.0684	0.0005	0.6425	41.50
	PenPC GG	0.0022	0.0125	0.0019	0.0000	0.4059	60.96
	ModelFree GG	0.0285	0.0116	0.0188	0.0001	0.4381	58.50
x^2	PenPC	0.7671	0.0163	0.6843	0.0047	0.2007	126.85
	ModelFree	0.2413	0.0076	0.1018	0.0004	0.3253	73.19
	PenPC GG	0.2847	0.0182	0.2594	0.0005	0.1260	94.08
	ModelFree GG	0.1450	0.0113	0.0396	0.0001	0.2943	73.76
x^3	PenPC	0.5115	0.0047	0.3215	0.0036	0.7252	63.66
	ModelFree	0.6713	0.0027	0.0469	0.0003	0.6243	42.20
	PenPC GG	0.2623	0.0063	0.2125	0.0019	0.6721	52.64
	ModelFree GG	0.5029	0.0038	0.0265	0.0002	0.5998	43.26
e^x	PenPC	0.4929	0.0045	0.3038	0.0033	0.7227	61.14
	ModelFree	0.5797	0.0022	0.0745	0.0006	0.7192	34.91
	PenPC GG	0.1775	0.0075	0.1443	0.0011	0.6178	50.48
	ModelFree GG	0.4714	0.0046	0.0520	0.0004	0.6341	41.27

FDR = False Discovery Rate; FRR = False Recovery Rate; H Dist = Hamming's Distance

GG = Indicates the use of the Gaussian Graphical Extended BIC, rather than regular BIC.

Note that Step 1 FDR is calculated against the moral graph and Final FDR is calculated against the true graph.

2.5 Application to TCGA Data

2.5.1 Data Source

We obtained gene expression data from The Cancer Genome Atlas (TCGA) database for 551 breast cancer patients and 18,827 genes. The RNA-seq bam files were downloaded from the TCGA data portal. We counted the number of RNA-seq reads per gene and selected the 18,827 genes with at least 20 reads in 25% of samples. Then the read count data were log-transformed after read depth correction. To identify appropriate sets of genes to create graphical models from, we grouped genes based on pathway annotation obtained from the Pathway Commons 2 API (cPath2). A total

of 130 gene groups were found, ranging from membership numbers of 1 gene to 2,208 genes. We chose to run the analysis on three clinically interesting groups consisting of $p = 64$ to 148 genes with a single random sample of $n = 135$ patients and cross-validate against the remaining $n = 416$ patients. The selected gene groups belonged to pathways: (1) T-Cell Receptor Signaling Pathway (TCR), (2) B-Cell Receptor Signaling Pathway (BCR), and (3) Angiogenesis Signaling Pathway (Angiogenesis).

2.5.2 Analysis Results

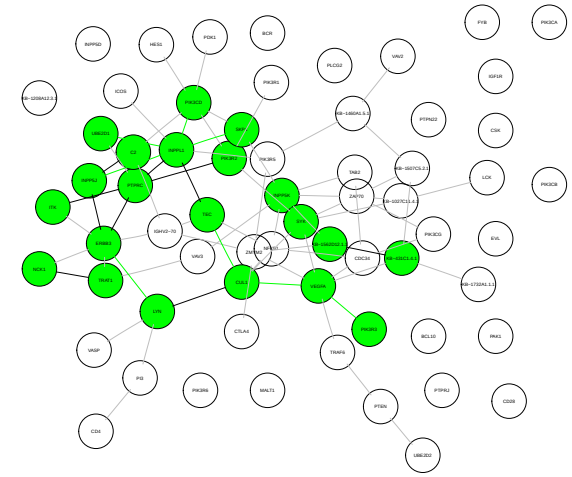
Table 2.5: Results for TCGA Breast Cancer Data

Gene Group	Method (Sample)	# of Edges	Connected Vertices	Total Vertices	% of Excl. Edges
TCR	PenPC	49	50	64	76%
	PenPC (CV)	109	59	64	61%
	Model Free	19	21	64	37%
	Model Free (CV)	64	46	64	34%
BCR	PenPC	44	42	61	59%
	PenPC (CV)	108	59	61	60%
	Model Free	25	33	61	28%
	Model Free (CV)	57	44	61	25%
Angiogenesis	PenPC	122	109	139	79%
	PenPC (CV)	329	136	139	77%
	Model Free	36	47	139	28%
	Model Free (CV)	132	93	139	42%

Excl. Edges refer to edges found with one method but not the other.

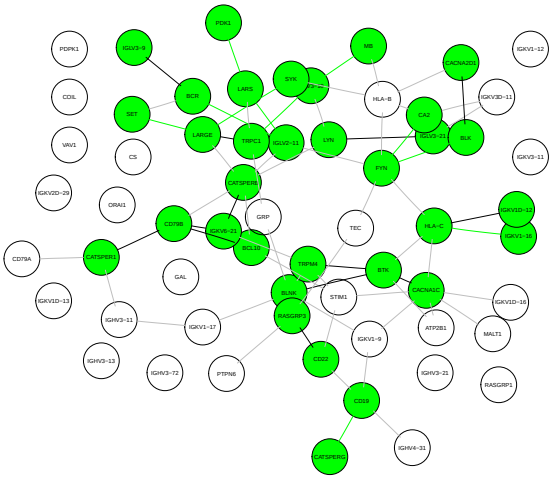
Figure 2.1: Estimated Graphs for TCGA Breast Cancer Data.

(a) TCR



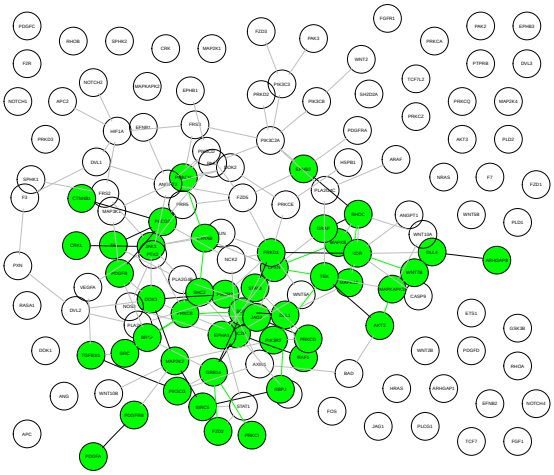
Nodes Connected by Original Sample Estimate
Edges found by Both Samples
Edges only found by Cross Validation Sample
Edges only found by Original Sample

(b) BCR



Nodes Connected by Original Sample Estimate
Edges found by Both Samples
Edges only found by Cross Validation Sample
Edges only found by Original Sample

(c) Angiogenesis



Nodes Connected by Original Sample Estimate
Edges found by Both Samples
Edges only found by Cross Validation Sample
Edges only found by Original Sample

Table 2.6: Mean Results for Mardia's Test of Multivariate Normality for Residuals of Exclusive Pairs

	Percentile	Skew Stat	Skew P	Kurt. Stat	Kurt. P	Max Mah. Dis
PenPC	25%	0.226	0.260	9.134	0.099	15.176
	50%	0.620	5.91e-03	10.412	4.59e-04	20.525
	75%	1.241	8.29e-06	12.510	6.40e-11	30.381
Model	25%	0.473	0.031	9.655	0.016	17.582
Free	50%	1.071	7.628e-05	10.682	1.09e-04	25.164
	75%	1.979	6.746e-09	13.233	6.505e-14	37.294

Max Mah. Dis refers to maximum Mahalanobis Distance.
Kurt. refers to kurtosis.

Table 2.7: Similarity and Differences between Estimates from Original and Cross-Validated Sample

	Method	# Edges in Original Sample	Overlapping Edges	P-Value	Pos Discrep Total	Attr to Sample Space	Attr to Sample Size	Neg Discrep Total	Attr to Sample Space	Attr to Sample Size
TCR	PenPC	49	27	4.41e-23	22	18%	77%	82	20%	2%
	Model Free	19	10	3.75e-11	9	44%	44%	54	30%	31%
BCR	PenPC	44	22	8.47e-16	22	23%	77%	86	28%	2%
	Model Free	25	13	2.44e-14	12	42%	42%	44	55%	14%
Angio	PenPC	122	64	8.15e-61	58	3%	83%	265	16%	0%
	Model Free	36	22	2.12e-32	14	14%	43%	110	39%	16%

Neg Discrep refers to edges not found in the original sample but found in the cross-validation sample.
Pos Discrep refers to edges found in the original sample but not found in the cross-validation sample.

Both *PenPC* and our Model Free approach were applied to the three different gene sets using both sets of samples (termed: original and cross-validation). The tuning parameters were selected by Gaussian Graphical extended BIC. Basic summary of results can be found in *Table 2.5*. Similar to our simulation results, we see that *PenPC* finds more edges than Model Free and accordingly has a larger set of connected vertices. While 25-40% of the edges identified by Model Free are not found by *PenPC*, around 60-80% of the edges identified by *PenPC* are not found by Model Free approach. These trends are consistent between samples. With a much larger sample size, we expect an increase in power and true discovery rate, which agrees with the increased number of edges found in the cross-validation sample for all gene groups across both methods.

Figure 2.1 gives a visualization of the estimated graphs for each gene group as well as the agreement between the original and cross-validation sample. We found the amount of overlap to be statistically significant for all three gene groups across both methods (Table 2.7) against the null that the graphs estimated are independent. To compare the discrepancies between the two sample estimates, we looked at positive discrepancies (when the original sample found an edge that was not found in the cross-validation sample) and negative discrepancies (when the original sample did not find an edge that was found in the cross-validation sample). Within these categories, we split the discrepancies that can be attributed to sample space and sample size. Discrepancies which occurred when all other results were in agreement were attributed to the difference in "sample size $\times$ method interaction" (for example, PenPC found an edge in the original sample but Model Free did not find it in either sample and PenPC did not find it in the cross-validation sample). Discrepancies which occurred between the original sample and the cross-validation sample for both methods were attributed to the difference sample space (aka. what was feasible to find within that sample).

For both PenPC and Model Free methods, the vast majority of positive discrepancies

can be attributed to either sample size × method interaction or sample space. Sample size × method interaction is more often the reason for PenPC while two reasons split more evenly for Model Free method. This indicates that the Model Free method is more volatile across different samples. It also implies that the high false discovery rate of PenPC can be mitigated with sample size. For negative discrepancies, PenPC can attribute less than 30% to sample space, with nearly none attributed to sample size × method interaction. Model Free attributes over 60% for either reasons and again is similarly split.

The discrepancies between the two methods can largely be attributed to differences in their model assumptions, namely linearity and multivariate normality. Most PenPC exclusive edges have relatively smaller effect sizes (left panel of Figure F1a), which suggest that PenPC has higher power for smaller effect sizes when linearity assumption is correct. In contrast, Model Free-exclusive edges have more uniform distribution of effect sizes (right panel of Figure F1a), suggesting that PenPC miss those edges because of non-linear relationships rather than effect sizes. We see that this trend is even more pronounced in the cross-validation sample based on Figure F1b. This is congruent with our cross-validation comparison.

For multivariate normality, we looked at whether the residuals of each pair followed bivariate normal distribution given the other variables that had been selected by mGAP. This provides us a proxy for whether the original data followed a multivariate normal distribution. To test for bivariate normality, we used Mardia's Tests for multivariate normality which tests for deviation of skew and kurtosis, as well as observations with large Mahalanobis distance from the expected distribution [Mardia (1970)]. Full results can be seen in *Table 2.6*. As expected, on average PenPC-exclusive pairs have smaller deviation statistics in all three sets. We see similar trends for the cross-validation sample (*Table F1*) with much smaller differences, which is congruent with the increase

in sample size.

2.6 Discussion

The use of -omic data to guide disease prognosis, prevention, and treatment is becoming a popular approach of precision medicine. The complexity and sheer number of variables within these data sets have set some new statistical challenges. Low false discovery rate is especially important to obtain meaningful results with a large number of variables. In comparison to *PenPC*, we have shown that Model Free obtains much lower false discovery rates at a relatively small cost to power when the relation among variables is non-linear. As shown in our real data analysis, Model Free was able to better capture associations that do not satisfy multivariate Gaussian assumption and/or have non-linear relations. Further, Model Free's lower false discovery rate is reflected in the increased parsimony of the selected models lending to easier interpretability.

For application in denser graphical models, we may insert a dimension reduction step between model free variable selection and conditional independence testing. This would be particularly useful if the selected Markov Blanket is too large to be handled effectively with KCI-test. We have explored the use of dimension reduction with sliced inverse regression (SIR) [Li (1991)] and sliced average variance estimation (SAVE) [Cook (2000)]. In *Table 2.8*, we compare the results obtained with or without dimension reduction to 1-dimension, regardless of whether a suitable 1-dimension central subspace has been found. In addition to KCI-test, we also include a conditional independence test by [Song et al. (2009)] which requires that the condition set has a dimension of one.

As expected, with dimension reduction, there is an increase in FDR which is sometimes accompanied by a small increase in power. Hamming's Distance shows that the overall effect is negative, but also minor in most cases. In addition, the choice of

Table 2.8: Dimension Reduction for Model-Free GG Simulation Results using (2.4)

$h(x)$	CI	DR	$\mathcal{G}_{\mathcal{M}}$ Estimate		$\mathcal{G}$ Final Estimate			
	Method	Method	FDR	FRR	FDR	Type I Err	Power	H Dist
x	Zhang	None	0.018	0.000	0.012	0.000	0.978	3.58
	Zhang	SIR	0.018	0.000	0.011	0.000	0.869	14.86
	Zhang	SAVE	0.018	0.000	0.012	0.000	0.906	11.04
	Song	SIR	0.018	0.000	0.010	0.000	0.888	12.90
	Song	SAVE	0.018	0.000	0.010	0.000	0.929	8.72
x^2	Zhang	None	0.342	0.005	0.163	0.001	0.547	57.92
	Zhang	SIR	0.342	0.005	0.251	0.002	0.626	60.46
	Zhang	SAVE	0.342	0.005	0.234	0.002	0.684	54.84
	Song	SIR	0.342	0.005	0.356	0.002	0.344	87.12
	Song	SAVE	0.342	0.005	0.349	0.002	0.368	85.86
x^3	Zhang	None	0.356	0.003	0.072	0.001	0.682	38.68
	Zhang	SIR	0.356	0.003	0.092	0.001	0.636	44.70
	Zhang	SAVE	0.356	0.003	0.178	0.002	0.680	48.88
	Song	SIR	0.356	0.003	0.123	0.001	0.694	42.26
	Song	SAVE	0.356	0.003	0.214	0.002	0.765	46.42
$\exp(x)$	Zhang	None	0.352	0.002	0.109	0.001	0.869	24.63
	Zhang	SIR	0.3518	0.002	0.141	0.001	0.755	38.50
	Zhang	SAVE	0.3518	0.002	0.199	0.002	0.823	39.90
	Song	SIR	0.3518	0.002	0.125	0.001	0.832	30.22
	Song	SAVE	0.3518	0.002	0.192	0.002	0.893	33.56

CI Method = Conditional Independence test used for Step 2
DR Method = Dimension Reduction method
FDR = False Discovery Rate; FRR = False Recovery Rate
H Dist = Hamming's Distance

dimension reduction method and conditional independence testing method does not make much difference outside of the quadratic case. SAVE performs better than SIR with the quadratic transformation, which is known from earlier studies [Cook (2000)]. Overall, KCI-test has better performance than [Song et al. (2009)]'s method. Although current results fail to show the advantage of this dimension reduction step, it remains an opportunity for future research.

In this dissertation, we focused on the difference between *PenPC* and our model-free based method. In practice, the results of these two methods complement each

other. *PenPC* can identify weaker effects when multivariate Gaussian and linearity assumptions are satisfied. Model free estimation has higher power when there is non-linear relations. Therefore, we recommend the use of both methods in data analysis and to combine their results for more accurate graphical model estimation in real world data.

CHAPTER 3: GRAPHICAL MODEL FOR SCRNA-SEQ DATA

Recall that we denote a DAG skeleton by $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V} = \{Y_1, ..., Y_p\}$ is the the vertex set, and $\mathbf{E}$ is the edge set. A moral graph $\mathcal{G}_\mathcal{M} = (\mathbf{V}, \mathbf{E}_\mathcal{M})$ can be derived from a skeleton $\mathcal{G}$ by connecting the co-parents in all the v-structures in $\mathcal{G}$. The data consists of total read count (TReC) from p genes across n single cells. Similar to the work in the previous chapter, we aim to estimate a DAG skeleton for these p genes in two steps. First construct a moral graph by neighborhood selection. Next remove false edges in the moral graph by a modified PC-algorithm. The specific approaches in both steps are designed to analyze single cell RNA-seq data, which are discrete counts with inflation of zeros.

3.1 Overall Algorithm

Negative binomial distribution has been widely used to model read count data from bulk tissue RNA sequencing. As mentioned in Chapter 1.2, a major difference between bulk RNA-seq and single cell RNA-seq data is that the latter often have many zeros. Therefore, we choose to model such single cell RNA-seq data by zero-inflated negative binomial distribution (ZINB), which is a mixture distribution with one component being 0, and the other component being a negative binomial distribution. For the first step of our algorithm, the neighborhood selection step, we allow both the zero inflation and the negative binomial components to be associated with other genes, and perform jointly penalized ZINB regression. For the second step of our algorithm, we use a likelihood ratio test to assess conditional dependence. We first present the overall algorithm and

give the details of each step in the following sections.

Denote the p genes by $\mathbf{Y} = \{Y_1, ..., Y_p\}$. Recall that

- $A_{\mathcal{G},i,j} = \text{adj}(\mathcal{G}, Y_i) \bigcup \text{adj}(\mathcal{G}, Y_j) \backslash \{Y_i, Y_j\}$, i.e., the Markov Blanket of Y_i and Y_j. Recall that $\text{adj}(\mathcal{G}, Y_j)$ denotes the neighbors of Y_j in graph $\mathcal{G}$.

- $B_{\mathcal{G},i,j} = \text{adj}(\mathcal{G}, Y_i) \bigcap \text{adj}(\mathcal{G}, Y_j) \backslash \{Y_i, Y_j\}$, which include all potential common children of Y_i and Y_j.

- $C_{\mathcal{G},i,j} = A_{\mathcal{G},i,j} \bigcap (B_{\mathcal{G},i,j} \bigcup Con_{\mathcal{G}}(B_{\mathcal{G},i,j}))$, where $Con_{\mathcal{G}}(B_{\mathcal{G},i,j})$ denotes the vertices that are connected to $B_{\mathcal{G},i,j}$. $C_{\mathcal{G},i,j}$ includes any possible common descendants of Y_i and Y_j within the Markov Blanket of Y_i and Y_j.

- $\mathbf{\Pi}_{i,j} = \{A_{\mathcal{G},i,j} \backslash \tilde{C}_{\mathcal{G},i,j}, \tilde{C}_{\mathcal{G},i,j} \subseteq C_{\mathcal{G},i,j}\}$. At least one of the set in $\mathbf{\Pi}_{i,j}$ includes all common parents of Y_i and Y_j, but excludes any common descendants. In other words, if there is any d-separation set of Y_i and Y_j, it will be included in $\mathbf{\Pi}_{i,j}$.

Algorithm 3: Moral Graph Estimation

Data: $\mathbf{Y}$

Result: $\mathcal{G}_{\mathcal{M}} = (\mathbf{V}, \mathbf{E})$

for $j = 1$ **to** p **do**

Let $y = Y_k$ and $X = Y_{-k}$.

Select variables among X associated with y by jointly penalized zero-inflated negative binomial regression, denoted by $\mathcal{E}$.

For any j such that $Y_j \in \mathcal{E}$, create edges $E_{k,j}$ and $E_{j,k}$. $E_{k,j} \equiv E_{j,k}$ for the undirected graph.

Algorithm 4: Modified-PC Algorithm

Data: $\mathcal{G}_{\mathcal{M}} = (\mathbf{V}, \mathbf{E})$, $\mathbf{X}$

Result: $\mathcal{G}$

foreach $(\mathbf{E}_{i,j}) \in \mathbf{E}$ **do**

  if $X_i \perp X_j$ **then** ; `/* Likelihood Ratio Test */`

    Remove $E_{i,j}$ and $E_{j,i}$.

  $l = -1$ **repeat**

    $l = l + 1$

    $\tilde{\mathcal{G}} = \mathcal{G}$, **foreach** $(i, j); |C_{\tilde{\mathcal{G}},i,j}| \geq l$ **do**

      foreach $[\Gamma; \Gamma \subseteq C_{\tilde{\mathcal{G}},i,j}, |\Gamma| = l]$ **do**

        $\kappa = A_{\tilde{\mathcal{G}},i,j} \backslash \Gamma$ **if** $X_i \perp X_j | X_\kappa$ **then** ; `/* Likelihood Ratio Test */`

          Remove $E_{i,j}$ and $E_{j,i}$.

          Exit while loop and move to next $E_{i,j}$ in for each loop.

  until $\max_{i,j} |C_{\tilde{\mathcal{G}},i,j}| < l$;

We refer to our algorithm for DAG skeleton estimation as scZINB, which stands for single cell and Zero Inflated Negative Binomial distribution.

3.1.1 Step 1: Neighborhood selection

We apply our neighborhood selection method for each variable (i.e., a node in the graph) separately. To select the neighborhood of the j-th variable, we model the observed count data of the j-th variable by ZINB distribution, and use log-transformed data of all the other variables as covariates. Since our neighborhood selection method is a general variable selection method and it can be applied to other settings, we use more generic notations in the following discussions. Let the observations of the (count) response variable be $\mathbf{y} = (y_1, ..., y_n)^T$ and the (continuous) covariate data be $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_p)$, where $\mathbf{x}_j = (x_{j1}, ..., x_{jn})^T$, and $\mathbf{x}_1 = (1, ..., 1)^T$ is the intercept.

Let X_i be the i-th row of matrix $\mathbf{X}$. z_i is an indicator that equals to 1 if y_i arises from the zeo-inflated component of the ZINB distribution, and 0 otherwise. Our ZINB log-likelihood function is then:

$$\ell = \sum_{i=1}^{n} \left\{ \log[\pi_i I(y_i = 0) + (1 - \pi_i) f_{\text{NB}}(y_i; \mu_i, \phi)] \right\}, \tag{3.1}$$

where $f_{\text{NB}}(y_i; \mu_i, \phi) = \frac{\Gamma(y_i + 1/\phi)}{y_i! \Gamma(1/\phi)} \left(\frac{1}{1+\phi\mu_i} \right)^{1/\phi} \left(\frac{\phi\mu_i}{1+\phi\mu_i} \right)^{y_i}$ denotes a negative binomial distribution with mean μ_i and over-dispersion parameter ϕ, and π_i is the probability that an observation arises from the zeo-inflated component. We associated both μ_i and π_i with the covariates by

$$\mu_i = \exp(X_i \boldsymbol{\beta}), \text{ and } \pi_i = \frac{\exp(X_i \boldsymbol{\gamma})}{1 + \exp(X_i \boldsymbol{\gamma})}. \tag{3.2}$$

The complete data log likelihood for the i-th observation is

$$\ell(y_i) = \begin{cases} \log(\pi_i) & \text{if } z_i = 1 \\ \log[(1 - \pi_i) f_{\text{NB}}(y_i; \mu_i, \phi)] & \text{if } z_i = 0 \end{cases}$$

which can be written as

$$\begin{aligned} \ell(y_i) &= z_i \log(\pi_i) + (1 - z_i) \log[(1 - \pi_i) f_{\text{NB}}(y_i; \mu_i, \phi)] \\ &= z_i \log \left[\frac{\exp(X_i \boldsymbol{\gamma})}{1 + \exp(X_i \boldsymbol{\gamma})} \right] - (1 - z_i) \log[1 + \exp(X_i \boldsymbol{\gamma})] + (1 - z_i) \log[f_{\text{NB}}(y_i; \mu_i, \phi)] \\ &= z_i X_i \boldsymbol{\gamma} - \log[1 + \exp(X_i \boldsymbol{\gamma})] + (1 - z_i) \log[f_{\text{NB}}(y_i; \mu_i, \phi)]. \end{aligned}$$

For our application of studying gene-gene co-expression, the dimension of the covariates p can be larger or much larger than sample size n, and thus we further impose a penalty function for the regression coefficients. We adopt the log penalty for its

desirable performance in high-dimensional settings (Chen et al. 2016a).

$$\ell = \sum_{i=1}^{n} \{\log[\pi_i I(y_i = 0) + (1 - \pi_i) f_{\text{NB}}(y_i; \mu_i, \phi)]\} + \sum_{k=2}^{p} \lambda \log(|\beta_k| + |\gamma_k| + \tau), \quad (3.3)$$

where λ and τ are two tuning parameters of the log penalty. β_1 and γ_1 are not penalized because they are coefficients for intercepts.

In the above equation, the form of the penalty β_k and γ_k implicitly assumes they are of the same scale, while it is not true because they are regression coefficients for different types of model (negative binomial versus logistic regression) with different types of response variables. Following the strategy used by McDavid et al. (2016), we weight β_k and γ_k in our penalty function by the inverse of their variance, hence the weighted versions of β_k and γ_k have the same variance. More specifically, let the weights for β_k and γ_k be $w_k^{(\beta)}$ and $w_k^{(\gamma)}$, then the penalty term becomes $\sum_{k=2}^{p} \lambda \log(w_k^{(\beta)}|\beta_k| + w_k^{(\gamma)}|\gamma_k| + \tau)$, where

$$w_k^{(\beta)} = -\frac{\partial^2 \ell}{\partial \beta_k^2}\bigg|_{\mu=\mu_0}, \quad w_k^{(\gamma)} = -\frac{\partial^2 \ell}{\partial \gamma_k^2}\bigg|_{p=p_0},$$

and $\mu_0 = \exp(0) = 1, p_0 = \frac{\exp(0)}{1+\exp(0)} = 0.5$. The derivation of the two 2nd derivatives can be found in Appendix C.

Since this is a typical mixture distribution problem, we use an EM algorithm to obtain maximum likelihood estimate (MLE) of the parameters. We describe the E-step and M-step of our algorithm in the following sections.

Expectation

Since the complete data likelihood is a linear function of z_i, the expectation given current estimates of parameters (i.e., the Q function) can be simply computed by

plugging in z_i with its expectation.

$$\begin{aligned}\hat{z}_i = \mathbb{E}[z_i|y_i] = P(z_i = 1|y_i) &= \frac{P(y_i|z_i = 1)P(z_i = 1)}{P(y_i|z_i = 1)P(z_i = 1) + P(y_i|z_i = 0)P(z_i = 0)} \\ &= \frac{I(y_i = 0)\pi_i}{I(y_i = 0)\pi_i + f_{\text{NB}}(y_i; \mu_i, \phi)(1 - \pi_i)}. \end{aligned} \tag{3.4}$$

Maximization

Instead of maximizing the Q function, we minimize the penalized negative Q function. Using a coordinate descent algorithm, we estimate β_k and γ_k iteratively with the following objective function:

$$\begin{aligned}\mathcal{O}_k &= -\sum_i \{\hat{z}_i X_i \boldsymbol{\gamma} - \log[1 + \exp(X_i \boldsymbol{\gamma})] + (1 - \hat{z}_i) \log[f_{\text{NB}}(y_i; \mu_i, \phi)]\} \\ &\quad + I(k > 1)\lambda \log(w_k^{(\beta)}|\beta_k| + w_k^{(\gamma_k)}|\gamma_k| + \tau) \end{aligned} \tag{3.5}$$

We minimize this objective function iteratively with respect to β_k and γ_k with the following two objective functions:

$$\mathcal{O}_k^{(\beta_k)} = -\sum_i (1 - \hat{z}_i) \log[f_{\text{NB}}(y_i; \mu_i, \phi)] + I(k > 1)\lambda \log(w_k^{(\beta)}|\beta_k| + w_k^{(\gamma_k)}|\gamma_k| + \tau) \tag{3.6}$$

$$\begin{aligned}\mathcal{O}_k^{(\gamma_k)} &= -\sum_i \{\hat{z}_i X_i \boldsymbol{\gamma} - \log[1 + \exp(X_i \boldsymbol{\gamma})]\} \\ &\quad + I(k > 1)\lambda \log(w_k^{(\beta)}|\beta_k| + w_k^{(\gamma_k)}|\gamma_k| + \tau) \end{aligned} \tag{3.7}$$

Equation (3.6) is just a penalized weighted negative binomial and equation (3.7) is just a penalized logistic regression. We then employ two approximations for this maximization problem.

First, we apply a local linear approximation (LLA) of the penalty around the current estimate for the parameter to be estimated. For example, if we are currently optimizing equation (3.6) for β_k where the current estimate is $\beta_k^{(t)}$, then we can approximate our

penalty as:

$$\begin{aligned}\lambda \log(w_k^{(\beta)}|\beta_k| + w_k^{(\gamma_k)}|\gamma_k| + \tau) = \lambda \log(w_k^{(\beta)}|\beta_k^{(t)}| + w_k^{(\gamma)}|\gamma_k| + \tau) \\ + \frac{\lambda w_k^{(\beta)}}{w_k^{(\beta)}|\beta_k^{(t)}| + w_k^{(\gamma)}|\gamma_k| + \tau}(|\beta_k| - |\beta_k^{(t)}|).\end{aligned} \tag{3.8}$$

We also apply a quadratic approximation of the objective function. Since equation (3.7) is a well solved GLM, we can use its iteratively reweighted least squares (IRLS) quadratic approximation for the likelihood, with generalized form: -$\Sigma_i v_i(\xi_i - X_i\beta)^2$ where we have, for the logistic model:

$$v_i^{(\gamma)} = \pi_i(1-\pi_i) \tag{3.9}$$

$$\xi_i^{(\gamma)} = X_i\gamma + \frac{\hat{z}_i - \pi_i}{v_i^{(\gamma)}} \tag{3.10}$$

Similarly, if we fix ϕ, we can use the IRLS quadratic approximation for equation (3.6) where

$$v_i^{(\beta)} = (1-\hat{z}_i)\frac{\mu_i^2}{\mu_i + \mu_i^2\phi} \tag{3.11}$$

$$\xi_i^{(\beta)} = X_i\beta + \frac{y_i - \mu_i}{\mu_i} \tag{3.12}$$

Exact derivations can be found in the appendix.

Taking these approximations together and dropping the constants, the final approximated objective functions are:

$$\mathcal{O}_k^{(\beta)} = \Sigma_i v_i^{(\beta)}(\xi_i^{(\beta)} - X_i\beta)^2 + I(k>1)|\beta_k|\frac{\lambda}{w_k^{(\beta)}|\beta_k^{(t)}| + w_k^{(\gamma)}|\gamma_k| + \tau} \tag{3.13}$$

$$\mathcal{O}_k^{(\gamma)} = \Sigma_i v_i^{(\gamma)}(\xi_i^{(\gamma)} - X_i\gamma)^2 + I(k>1)|\gamma_k|\frac{\lambda}{w_k^{(\beta)}|\beta_k| + w_k^{(\gamma)}|\gamma_k^{(t)}| + \tau} \tag{3.14}$$

Setting $\frac{\partial \mathcal{O}_k^{(\beta)}}{\partial \beta_k} = 0$ gives us the following iterative updates:

$$\beta_k = \begin{cases} \nu_{(\beta)}^{-1}\left(\bar{b}_k - \frac{\lambda I(k>1)}{2(w_k^{(\beta)}|\beta_k^{(t)}|+w_k^{(\gamma)}|\gamma_k|+\tau)}\right), & \text{if } \frac{\bar{b}_k}{\nu_{(\beta)}} > \frac{\lambda I(k>1)}{2\nu_{(\beta)}(w_k^{(\beta)}|\beta_k^{(t)}|+w_k^{(\gamma)}|\gamma_k|+\tau)} \\ \nu_{(\beta)}^{-1}\left(\bar{b}_k + \frac{\lambda I(k>1)}{2(w_k^{(\beta)}|\beta_k^{(t)}|+w_k^{(\gamma)}|\gamma_k|+\tau)}\right), & \text{if } \frac{\bar{b}_k}{\nu_{(\beta)}} < -\frac{\lambda I(k>1)}{2\nu_{(\beta)}(w_k^{(\beta)}|\beta_k^{(t)}|+w_k^{(\gamma)}|\gamma_k|+\tau)} \\ 0 & \text{otherwise} \end{cases} \tag{3.15}$$

where $\nu_{(\beta)} = \sum_i v_i^{(\beta)} X_{ik}^2$ and $\bar{b}_k = \sum_i v_i^{(\beta)}(\xi_i^{(\beta)} - X_{i,-k}\beta_{-k})X_{ik}$. Since the objective function for γ_k (equation 3.14) has the same form as the one for β_k (equation 3.13), γ_k can be solved similarly.

3.1.2 Step 2: Testing Conditional Independence

After estimating moral graph using neighborhood selection, we remove false positive connections using a modified PC algorithm, which removes an edge between two variables if they are conditional independent given any subset of other variables. Consider testing for conditional independence of two variables Y_1 and Y_2 with conditioning set Y_κ. Then, if Y_1 is the response variable, we have design matrix $\mathbf{X}_{2,\kappa} = [\mathbf{x}_2, \mathbf{X}_\kappa]$ where $\mathbf{x}_2 = \log(\mathbf{y}_2 + 1)$ and $\mathbf{X}_\kappa = \log(\mathbf{Y}_\kappa + 1)$. The coefficient vectors $\boldsymbol{\beta}_1 = [\beta_{1,2}, \boldsymbol{\beta}_{1,\kappa}^\intercal]^\intercal$ and $\boldsymbol{\gamma}_1 = [\gamma_{1,2}, \boldsymbol{\gamma}_{1,\kappa}^\intercal]^\intercal$. Similarly, if Y_2 is the response, we have design matrix $\mathbf{X}_{1,\kappa}$ and coefficient vectors $\boldsymbol{\beta}_2 = [\beta_{2,1}, \boldsymbol{\beta}_{2,\kappa}^\intercal]^\intercal$ and $\boldsymbol{\gamma}_2 = [\gamma_{2,1}, \boldsymbol{\gamma}_{2,\kappa}^\intercal]^\intercal$. Then, we perform two following likelihood ratio tests: $H_{01}: \ \beta_{1,2} = \gamma_{1,2} = 0$, and $H_{02}: \ \beta_{2,1} = \gamma_{2,1} = 0$. We remove the edge between Y_1 and Y_2 if the p-value of both tests are larger than a pre-specified significance level α.

3.2 Implementation Details

Due to the complex nature of this optimization problem, there were several implementation decisions made in order to improve runtime and accuracy of our algorithm.

Dynamic Tuning Parameters: Despite efforts to define the tuning parameter grid to encompass the full range of model sizes, in rare instances this would fail. In practice, we found it was beneficial to use a dynamic tuning parameter algorithm by appending additional tuning parameter grid searches if found necessary. This means, after running the model on an initial set of tuning parameters, we evaluated the sizes of models derived and either re-ran the model using a finer grid search on a subsection lacking granularity, or we expanded the range of penalty strengths to ensure full coverage.

Initial values of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$: We set the initial values of $\boldsymbol{\beta}$ by a penalized Poisson regression with Lasso penalty using the observed count data (including zero's) as response variable, and set the initial values of $\boldsymbol{\gamma}$ by a penalized logistic regression with Lasso penalty and the response variable is an indicator whether the observed count is 0.

First Step Screening: We found in simulations that marginal screening (to remove covariates that are not even weakly associated with the response variable) prior to penalized regression improved the performance of the variable selection and greatly reduced runtime.

Model Selection: For model selection, we found the best metric for model selection to be a version of extended BIC $-2\ell + k\log(n) + 2\gamma_{\text{BIC}}\log\binom{2p}{k}$, where k is the total number of non-zero coefficients including both β's and γ's. γ_{BIC} is the BIC tuning parameter which we set as $\log_n p$.

3.3 Simulation

3.3.1 Set-up

Similarly to our model-free algorithm set up, we will simulate the base graph structures using the Erdős and Rényi (ER) model where we connect the vertices randomly with equal probability (ErdËİos and RÂťenyi 1959). Specifically, the probability of having an edge for any pair of vertices $(Y_i,\ Y_j)$ with $i < j$ is $p_E = d_0/p$. Then, all graphs with p vertices and d edges have probability of $p_E^d(1-p_E)^{\binom{p}{2}-d}$ to be generated. A single graph is generated for the count process and for the zero-inflated process. For these particular simulations, we will use combinations of $n = 300$, $p = 100$ or 1000, and $d_0 = 1$ or 5.

For data generation, we will start with simulating all of the count data for all samples, $Y^{(NB)}$ and all of the latent binomial data for all samples, $Y^{(B)}$. We do so by first randomly generating multivariate normal data according to the underlying data structure:

$$\mathbf{Y} = \mathbf{B}^\intercal \mathbf{Y} + \mathbf{e} \tag{3.16}$$

by sampling from the distribution $N(0, \Sigma)$ where $\Sigma = (I_{p\times p} - \mathbf{B}^\intercal)^{-1}$ and $\mathbf{B}$ is a upper triangular matrix of coefficients. For the purpose of this simulation, we assume all coefficients are 1. We obtain the multivariate Gaussian copula then by applying the distribution function for $N(0,1)$ over the generated data ($\mathbf{U} = \Phi(\mathbf{Y})$). To transform the multivariate Gaussian copula data into marginally negative binomial and binomial data, we use the generalized inverse distribution function ($F_\cdot^{-1}$) for the negative binomial with $\mu = 2$ and $\theta = 1.5$ and binomial with success probability 0.6. We define $F_{\text{NB}}^{-1}(p) = \inf\{x \in \mathcal{D}_{\text{NB}} : F_{\text{NB}}(x) \geq p\}$, where F_{NB} is the negative binomial cumulative distribution function and $\mathcal{D}_{\text{NB}} \equiv \mathbb{N}$ is the domain of the negative binomial distribution. We define $F_{\text{B}}^{-1}(p)$ similarly. Then $Y_{ji}^{(NB)} = F_{\text{NB}}^{-1}(\mathbf{U})$ and $Y_{ji}^{(B)} = F_{\text{B}}^{-1}(\mathbf{U})$. We derive the final

observed variable Y_j:

$$Y_{ji} = \begin{cases} Y_{ji}^{(NB)} & \text{if } Y_{ji}^{(B)} = 0 \\ 0 & \text{if } Y_{ji}^{(B)} = 1 \end{cases} \tag{3.17}$$

3.3.2 Results

We conducted simulations in four settings with $n = 300$, $p = 100$ or $p = 1000$, and $d_0 = 1$ or 5. In the modified PC-algorithm portion of scZINB, a p-value threshold α is needed to evaluate the conditional independence testing results. We examined at a range of α values. We compared the performance of scZINB against the Hurdle model after transforming the zero-inflated count data to zero-inflated normal: $g(X) = \log(X + 1)$. The current Hurdle model method only supports BIC model selection. Since in scenarios with larger p, this model selection method does poorly, we augmented the results with extended BIC model selection as well as best possible graphical model across all tuning parameters. Here the best model is defined as the one that has smallest number of false positives plus false negatives. Similarly, in the sparse case of $n = 300$ and $p = 1000$ it is clear that the model selection criteria for scZINB is not sufficiently parsimonious. Therefore, we included the best model case for better comparisons.

We found that using $\alpha = 0.001$ consistently gave good results. In denser graphs, we see a marked decrease in the power while the true discovery rate remains consistent high. Compared to BIC or extended BIC Hurdle model selected graphs, scZINB sees similar results for $n = 300$ and $p = 100$ when the graph is sparse, but significantly better results in all other scenarios (Table 3.1).

The better performance of scZINB can be partially attributed to the difference in penalties, where the log penalty tends to have better performance in cases of higher density, as well as to the assumption of normality of the log transformed count data. *Table 3.2* contains comparisons of scZINB when using the Lasso penalty rather than the

log penalty. We see that when $d_0 = 1$ the performance is roughly equivalent, whereas Lasso's performance is much worse than the log penalty when $d_0 = 5$.

3.4 Application to Peripheral Blood Mononuclear Cell Data

3.4.1 Data Sourcing and Processing

For our data analysis, we use a sample of approximately 68k peripheral blood mononuclear cells sequenced by Zheng et al. (2017). We re-ran the raw data through the Cell Ranger 2.0 pipeline which included 1) converting Illumina's sequencing data into FASTQ files and 2) aligning, filtering, and UMI counting the data, using the GRCh38 transcriptome. This gave us a data set of 63,495 cells. There were 21,257 genes detected with 22,132 mean number of reads per cell and a median of 550 genes per cell. We trimmed down this data set by removing all genes with 0 read counts in all the cells. and finally ended up with a data set of 63,495 cells and 20,387 genes.

3.4.2 Clustering

For clustering, we followed the same clustering procedure laid out in Zheng et al. (2017). First, for each cell, we standardized the total UMI count per gene by dividing it with the cellular total UMI count, and then multiplying it by the median cellular total UMI count. Then, following the approach used by Zheng et al. (2017), we derived the top 1000 most variable genes after adjusting for their mean values. Specifically, we sort all genes into 20 bins by their total counts and calculating their normalized dispersion within each bin. Dispersion d_i was first calculated as the ratio of variance to mean. Normalized dispersion was calculated as $|d_i - d_{i,\text{med}}|/d_{i,\text{mad}}$ where $d_{i,\text{med}}$ was the median dispersion within the bin d_i belongs to, and $d_{i,\text{mad}}$ was the median adjusted absolute median deviation within the bin d_i belongs to. All normalized dispersion values were then pooled and the 1,000 genes with largest values were used for clustering.

To prepare for clustering, the data was log transformed after adding 1 to the count value. Then, to perform PCA analysis, we used a singular value decomposition on the standardized data set, such that each gene has mean 0 and variance 1 (Wall et al. 2003). K-means clustering was then applied to the eigenvectors corresponding to the largest 10 eigenvalues obtained from the SVD using the MacQueen (1967) algorithm for $k = 10$ and a maximum of 150 iterations. Results are plotted in *Figure 3.1* using coordinates derived by t-SNE (Maaten and Hinton 2008). Zheng et al. (2017) has also collected single cell RNA-seq data of 11 cell types. Following their approach, we calculated the correlation between the gene expression of each cell (using the 1,000 most variable genes) and the mean gene expression of the purified 11 sub-populations, and assigned a cell to the sub-population with highest correlation. Using this approach, we can assign each cluster based on the majority vote of the cells within the cluster. This gives us the following cell types per cluster: 1) clusters 1, 2, 7 and 8 are T lymphocyte cells, 2) cluster 10 are natural killer (NK) cells, 3) cluster 4 are B lymphocyte cells, 4) cluster 5 are myeloid cells.

Myeloid cells are key players in the metastatic process including detachment from the primary tumor and colonization at new sites (Toh et al. 2012). Mutations in myeloid cells are also the driver of acute myeloid leukemia (AML), the most common type of acute leukemia. They differentiate into macrophages and dendritic cells. Applying the same clustering strategy to cluster 5 alone revealed these substructures with $k = 4$ for k-means, as shown in *Figure 3.2*. Using gene expression heatmaps in *Figure 3.3*, we found that 1) subcluster 1 were CD16-/low monocytes, 2) subcluster 3 were CD16+ monocytes, and 3) subcluster 4 were dendritic cells. Subcluster 2 was small and consisted of a few wayward cytotoxic T cells. Thus, for the purposes of this analysis, we focused on subclusters 1, 3, and 4 of cluster 5. *Table 3.3* contains the exact number of cells within each subpopulation.

Figure 3.1: K-means clustering results compared with classification using subpopulation correlations for full population.

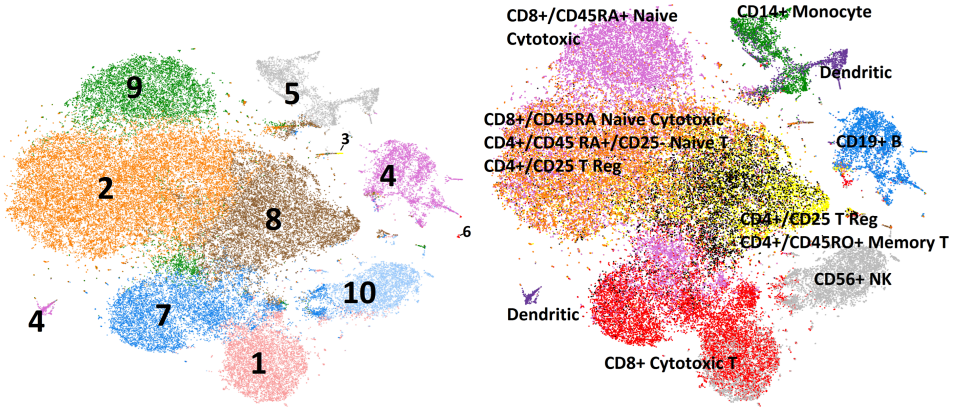


Figure 3.2: K-means clustering results compared with classification using subpopulation correlations for Myeloid Cells (Cluster 5).

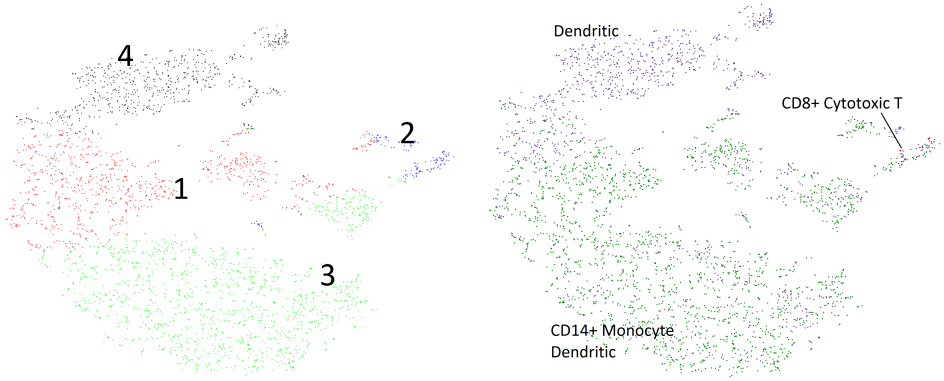
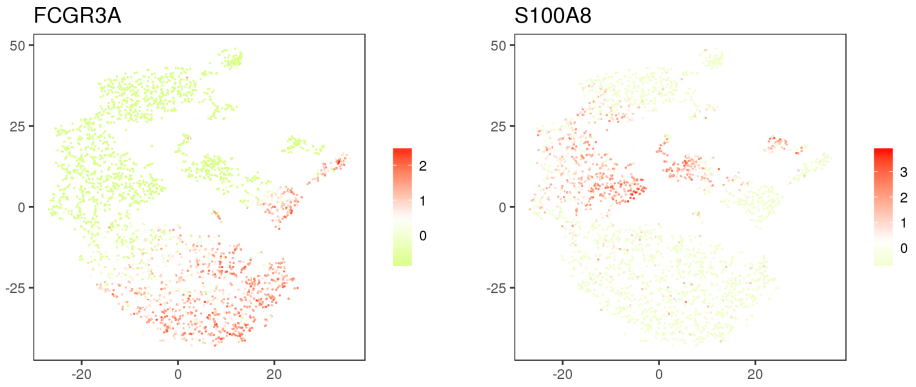


Figure 3.3: Gene expression heatmaps for macrophages.



3.4.3 Results

As expected from our simulations, a selective set of edges were found. Within the dendritic cells subcluster we filtered out genes with more than 80% zeroes. This left us with a sample size of 651 cells and 882 genes. Our final estimate for dendritic cells consisted of 208 genes with connections and 181 edges between genes. The full graph estimate can be found in *Figure 3.4*. For preliminary analysis we examined the pathways with the most number of connected vertices. The estimated graph had only a few connected genes involved with known pathways. We highlighted the genes genes from a few pathways that contribute larger number of connections to the graph.

Within the subcluster of CD16-/low Monocyte cells, we again applied a filter to remove the genes with more than 80% zeroes. This left us with a sample size of 939 cells and 445 genes. Based on our simulation results, we expect higher power and a similar true discovery rate of 80%+. Our final estimate consisted of 91 genes with connections and 75 edges between genes. We see a similar cluster of connected nodes involving S100A8 and S100A9. Interestingly, the classical CD16- monocyte is characterized by a high level of CD14 expression, in this particular graph we see that its expression is associated with S100A8 and S100A9, implying variable expression. The full graph estimate can be found at *Figure 3.5*. Similarly to the dendritic subcluster, for preliminary analysis we looked for the pathways with the most connected number of vertices. The three most represented pathways are highlighted: rRNA Processing in the Nucleus and Cytosol in green (9), Metabolism of Amino Acids and Derivatives as boxes (10), and Influenza Viral RNA Transcription and Replication (8).

These results demonstrate the very different gene regulatory networks that are derived when we separate out single cells to their respective sub populations. In the context of dendritic versus macrophages, the extensive differences between the two make sense as they have very different roles within the body. Macrophages tend to be

singularly focused on the engulfing and digesting of cellular debris, as indicated by the represented pathways, while dendritic cells identify foreign bodies to present as targets to the immune system.

In terms of similarities, we find that out of the 200+ connected genes, only 34 of them are shared between the cell types. These largely include genes focused on generalized immune system functions such as Lyz, which codes a protein that attacks cell walls, and HLA genes, which codes for proteins used by the immune system to identify foreign identities. In both cell types, the most connected vertex was S100A9 which has been found to have large impact with clinical significance for a multitude of diseases including leukemia, HIV, and cystic fibrosis. This highlights the usefulness of such comparisons in order to identify genes with high impact.

3.5 Discussion

In real world use, the high computation load of zero inflation modeling meant that a large amount of parallel computing was necessary for estimation to take place in a reasonable amount of time. Despite the numerous techniques used to improve computation time, it still remains our primary drawback. The main culprit of this is having to do a joint penalized mixed model which requires a long calculate and convergence time.

One of the avenues we explored to minimize this was to use a score test for zero inflation, in order to do a penalized negative binomial regression in the instances where zero inflation was not detected. There are challenges present in this. In Poisson data, there exists a score test which allows testing for zero-inflation without fitting the mixture model (van den Broek 1995) which would be ideal for computation efficiency. However, we found in the case of negative binomial data, the procedure used for Poisson data does not afford the same simplified test. In order to realistically test for zero-inflation, we

Figure 3.4: Full estimated graph of dendritic myeloid cells. Genes within pathways: rRNA Processing in the Nucleus and Cytosol, Metabolism of Amino Acids and Derivatives, and Influenza Viral RNA Transcription and Replication are highlighed. Unconnected singletons were removed. The graph consists of 208 genes and 181 edges between genes.

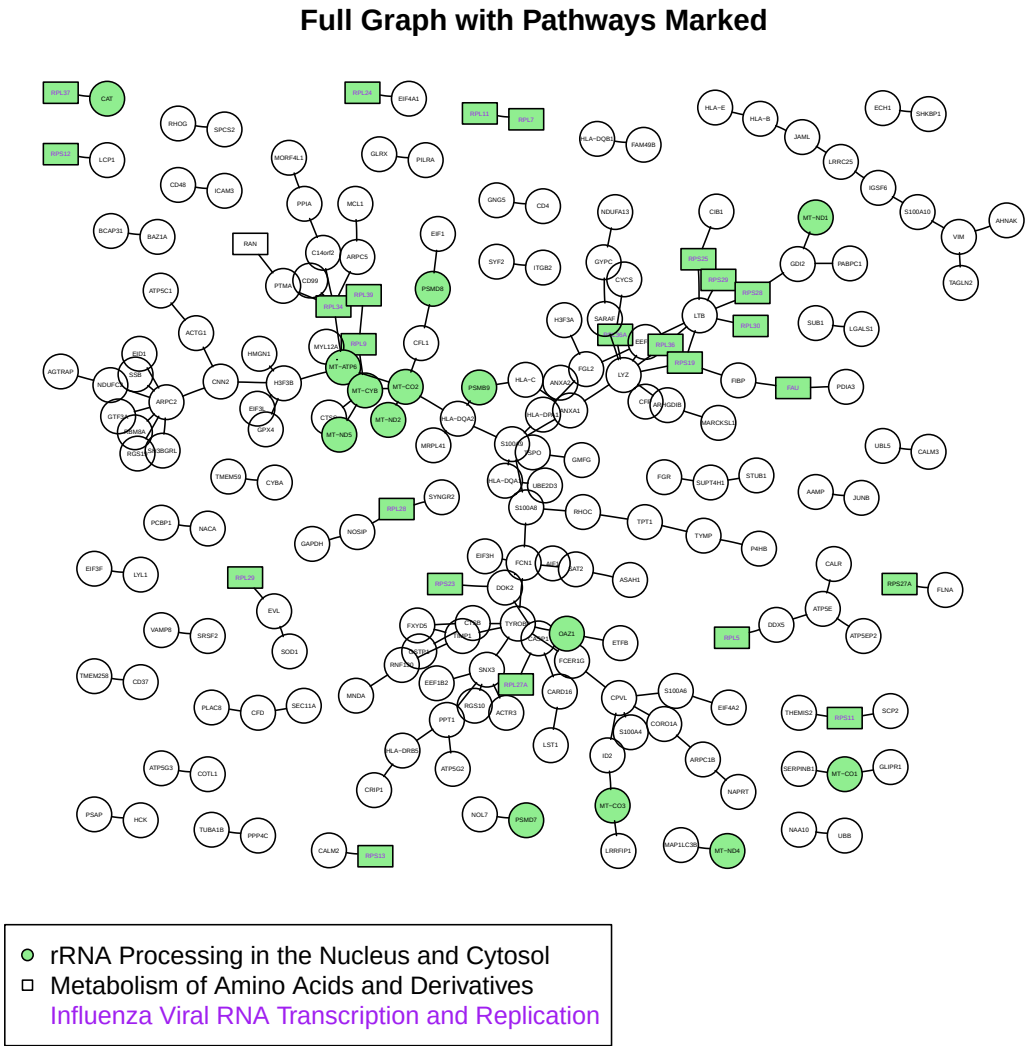
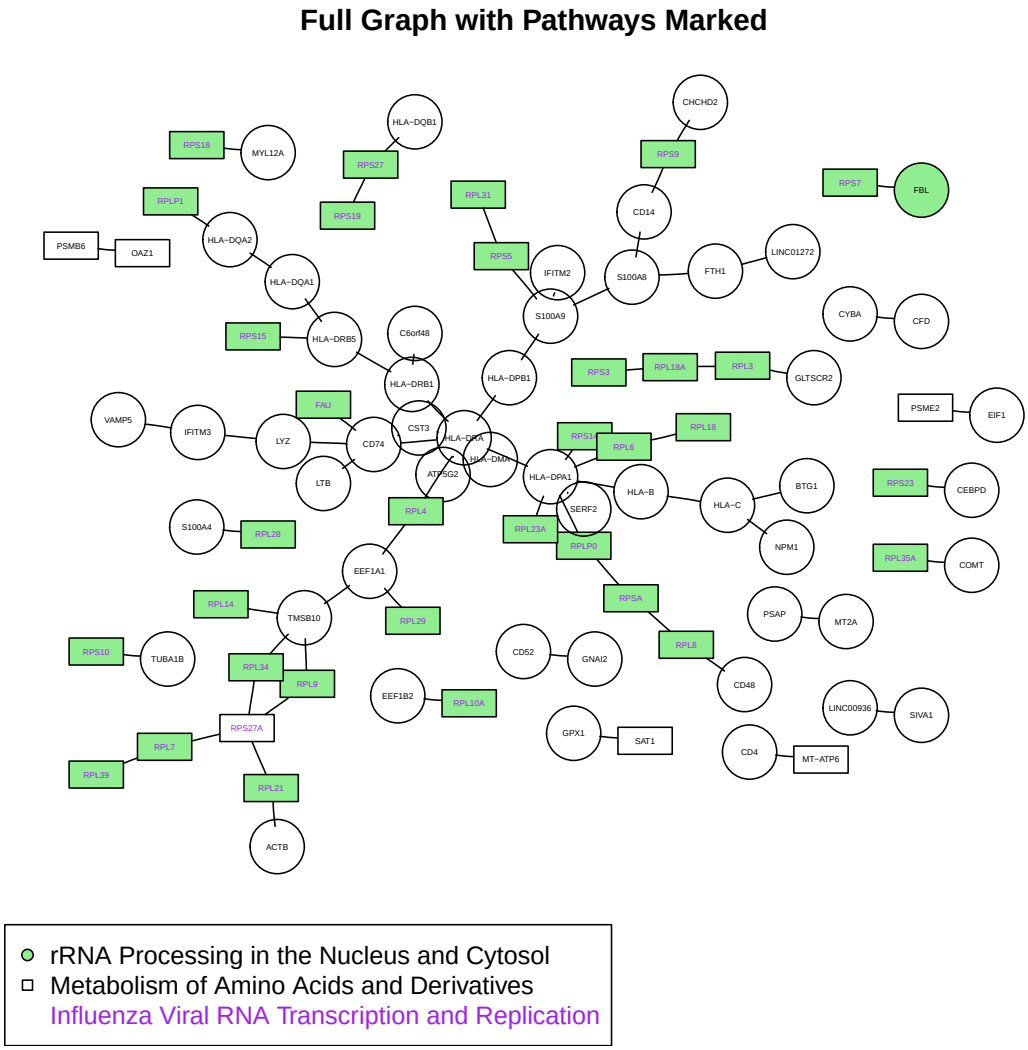


Figure 3.5: Full estimated graph of CD16-/low monocyte myeloid cells. Genes within pathways: rRNA Processing in the Nucleus and Cytosol, Metabolism of Amino Acids and Derivatives, and Influenza Viral RNA Transcription and Replication are highlighted. Unconnected singletons were removed. The graph consists of 231 genes and 224 edges between genes.



instead must use the identity link and test for $\gamma = 0$ rather than $\pi = 0$. Consequently, this makes it difficult to test one-sided hypotheses for π, which means that in cases of zero deflation, such as when the response has no zeros, the test would come back as significant (Jansakul and Hinde 2008). Further, since we use penalization for variable selection, if we wish not to fit the full mixture model first, we must use a pre-specified covariate structure. We tested whether or not testing just the marginal zero-inflation would be sufficient in simulations, and found the power to be too low to pursue further.

A comparison with the Hurdle model also highlighted the issue of scaling the BIC with p. While the scZINB model does not encounter this issue as often as the Hurdle model, the case of sparse $n = 300$ and $p = 1000$ makes it clear that it is still an issue. For the most part, a tighter α level seems to ameliorate most of the ill effects in the variable selection portion, however it does quickly increase the computation time.

Table 3.1: Full Simulation Results

n	p	d_0	Est Type	# Edges/ Step 2 α	Power	scZINB Model Type 1 Error	TDR
300	100	1	$\mathcal{G}_M$	94.8	0.996	0.009	0.545
			$\mathcal{G}$	0.1	0.996	0.008	0.563
			$\mathcal{G}$	0.05	0.996	0.007	0.600
			$\mathcal{G}$	0.001	0.968	0.000	0.957
300	100	5	$\mathcal{G}_M$	277.2	0.911	0.011	0.815
			$\mathcal{G}$	0.1	0.844	0.004	0.927
			$\mathcal{G}$	0.05	0.828	0.002	0.948
			$\mathcal{G}$	0.001	0.643	0.000	0.997
300	1000	1	$\mathcal{G}_M$	2212.5	0.984	0.003	0.219
			$\mathcal{G}$	0.05	0.981	0.003	0.240
			$\mathcal{G}$	0.001	0.933	0.000	0.670
			$\mathcal{G}$	0.0005	0.915	0.000	0.791
			$\mathcal{G}_M$ Best Model	679.8	0.974	0.000	0.705
			$\mathcal{G}$ Best Model	0.0005	0.918	0.000	0.951
300	1000	5	$\mathcal{G}_M$	2638	0.869	0.001	0.808
			$\mathcal{G}$	0.1	0.839	0.001	0.867
			$\mathcal{G}$	0.05	0.828	0.000	0.895
			$\mathcal{G}$	0.001	0.672	0.000	0.992
						Hurdle Model	
300	100	1	BIC	49.1	0.955	0.000	0.993
			extBIC	46.8	0.916	0.000	0.998
			Best Model	50.9	0.980	0.000	0.983
300	100	5	BIC	92.6	0.307	0.004	0.823
			extBIC	70.6	0.238	0.002	0.837
			Best Model	211.4	0.617	0.012	0.727
300	1000	1	BIC	113989.9	1.000	0.227	0.004
			extBIC	113989.9	1.000	0.227	0.004
			Best Model	454.4	0.868	0.000	0.947
300	1000	5	BIC	116063.3	0.998	0.229	0.021
			extBIC	116059.1	0.998	0.229	0.021
			Best Model	2235.8	0.822	0.000	0.916

*Effect size was set at 1 with the exception of no transformation which was set at 0.5.
TDR = True Discovery Rate

Table 3.2: Log Penalty versus Lasso Penalty Simulation Results

n	p	d_0	scZINB with Group Log Penalty			
			Step 2 α	Power	Type 1 Error	TDR
300	100	1	0.001	0.945	0.000	0.987
300	100	5	0.001	0.518	0.005	0.847
			scZINB with Lasso			
300	100	1	0.001	0.938	0.000	0.990
300	100	5	0.001	0.332	0.005	0.765

TDR = True Discovery Rate

Table 3.3: Subpopulation counts for full cell population and myeloid subpopulation.

All Cells	T cells	NK Cells	B Cells	Myeloid Cells
Count	52,450	3,633	3,972	3,234
Proportion	82.87%	5.72%	6.26%	5.09%
Myeloid Cells	CD16- Monocytes	Cytotoxic T	CD16+ Monocytes	Dendritic
Count	939	133	1,511	651
Proportion	29.04%	4.11%	46.72%	20.13%

CHAPTER 4: CONCLUSION

Over the course of this dissertation, we have discussed a generic two-step framework for estimating the skeletons of directed acyclic graphs and applied it to two types of data. We showed the flexibility of this framework to tackle a variety of situations when implemented in a modular manner. Under the generic scenario of non-linear and non-parametric data, we were able to demonstrate its efficacy in a variety of simulations. We saw its ability to better capture non-linear and non-parametric relationships in real world data. We explored variations on our selected methods within this framework, and showed robustness even under dimension reduction of the conditioning space under conditional independence testing. Theoretically, we were able to show asymptotic consistency properties in both steps. Within the narrower context of scRNA-seq data, we showed its efficacy as well as its computational disadvantages.

APPENDIX A: ALGORITHMS FOR MODEL-FREE APPROACH

Algorithm A1: Tuning Parameter Grid Generation

Data: X, y_t (transformation of y with h dimensions), τ_n (the number of τ's), λ_n (the number of λ's)

Result: λ, τ

$\lambda \leftarrow []$

$\tau \leftarrow []$

Choose τ uniformly in log scale. Let $\tau_{\max} = 1$, $\tau_{\min} = 10^{-6}$, and $d_\tau = [\log(\tau_{\max}) - \log(\tau_{\min})]/(\tau_n - 1)$. $\tau_t \leftarrow$ $[\exp(\log(\tau_{\min})), \exp(d_\tau + \log(\tau_{\min})), \exp(2d_\tau + \log(\tau_{\min})), .., \exp(\log(\tau_{\max}))]$

for $it = 1$ **to** τ_n **do**

- $\kappa \leftarrow \tau_t[it]$
- $\lambda.max \leftarrow 0$
- **for** $s = 1$ **to** h **do**
  - $\eta \leftarrow 0$
  - **for** $j = 1$ **to** $p - 1$ **do**
    - $\bar{b} = \left(X_j^\intercal X_j\right)^{-1} X_j^\intercal y_{t,s}$
    - $\omega = \|y_{t,s}\|^2/(n\|X_j\|^2)$
    - $\eta = \eta + \left(2\bar{b}\kappa/\omega\right)^2$
  - $\eta = \sqrt{\eta_t}$
  - **if** $\eta > \lambda_{max}$ **then**
    - $\lambda_{\max} = \eta$
- Let $\lambda_{\min} = 10^{-6}$, and $d_\lambda = [\log(\lambda_{\max}) - \log(\lambda_{\min})]/(\lambda_n - 1)$. $\lambda_t \leftarrow$ $[\exp(\log(\lambda_{\min})), \exp(d_\lambda + \log(\lambda_{\min})), \exp(2d_\lambda + \log(\lambda_{\min})), .., \exp(\log(\lambda_{\max}))]$
- $\tau = [\tau, \textbf{repeat } \kappa \ \lambda_n \textbf{ times}]$
- $\lambda = [\lambda, \lambda_t]$

APPENDIX B: PROOFS FOR THEORETICAL RESULTS OF MGAP

B.1 Weak Oracle Property for Model Free Variable Selection

For graphical model estimation, we treat one of the p variables as the response variable and the other $p-1$ variables as covariates and then apply model free variable selection alone. In this section, we focus on the model free variable selection. To simplify the notation, we denote the response variable as Y, the h spline transformation of Y as $T = (T_1, ..., T_h)$, and assume there are p covariates $X_1, ..., X_p$. Recall that our objective function is:

$$\mathcal{O}(B; \omega, \kappa, \lambda, \tau) = \sum_{s=1}^{h} \frac{\|T_s - \mathbf{X}B_s\|^2}{\omega_s} + \lambda \sum_{j=1}^{p} \frac{\|\mathbf{b}_j\| + \tau}{\kappa_j} + \lambda \sum_{j=1}^{p} \log(\kappa_j) + n \sum_{s=1}^{h} \log(\omega_s). \tag{4.1}$$

with $\|\mathbf{b}_j\| = \sqrt{\Sigma_{s=1}^{h} B_{j,s}^2}$. Without the loss of generality, we assume the covariates have been standardized to have mean 0 and L_2 norm $\|X_j\|^2 = \sqrt{n}$, $\forall j = 1, ..., p$.

We denote the true value of $\mathbf{B}$ by $\mathbf{B}^{(0)}$, hence its j-th row is denoted by $\mathbf{b}_j^{(0)}$, and its s-th column is denoted by $B_s^{(0)}$. We separate the space of the vectors $\{X_j : j \in \{1, 2, \ldots, p\}\}$ into the subspace of the supporting set (S) and its complement (S^c) where $X_j \in S$ if and only if $\|\mathbf{b}_j^{(0)}\| \neq 0$. Denote the size of S by d_0. Let $\mathbf{X}_1$ be an $n \times d_0$ matrix for those $X_j \in S$. Let $\mathbf{X}_2$ be an $n \times (p - d_0)$ matrix for those $X_j \in S^c$. We define $\mathbf{B}_1^{(0)}$ and $\mathbf{B}_2^{(0)}$ analogously for the supporting set and its complement. Then we use $B_{1,s}^{(0)}$ and $B_{2,s}^{(0)}$ to denote the s-th column of $\mathbf{B}_1^{(0)}$ and $\mathbf{B}_2^{(0)}$, respectively.

Condition 1.1 $\log(p) = O(n^\alpha)$ and $d_0 = O(n^\nu)$ with $0 \leq \alpha < 1$, $0 \leq \nu < 1/2$.

Condition 1.2 $d_n \equiv 2^{-1} \min_s \min_{1 \leq j \leq d_0} \{|b_{j,s}^{(0)}|\} = O(n^{-\gamma_0} (\log n)^{1/2})$ for some $\gamma_0 \in (\nu, 1/2)$.

Condition 1.3 $p'_s(d_n) = o(d_n)$. Where p'_s is the penalty gradient for transformation s. For the purposes of this proof, we are keeping the penalty general.

Condition 2.1 $p'_s(0+) > \sigma^{-\frac{1}{2}}\eta(1-K)^{-1}$ for $K \in (0,1)$ where $\eta = O(n^{\frac{\alpha-1}{2}}\sqrt{\log n})$, and $n^{\frac{1}{2}+\nu}\sqrt{\log n} = o(p'_s(0+))$.

Condition 2.2 $\frac{\|\mathbf{X}_2^\intercal \mathbf{X}_1\|_\infty}{\|\mathbf{X}_1^\intercal \mathbf{X}_1\|_\infty} \leq \min\left\{KO_p(n^{-1})\frac{p'_s(0+)}{p'_s(d_n)}, O_p(n^\nu)\right\}$.

Condition 3 $\lambda\kappa_0 = o(\tau_0)$, where $\kappa_0 = \max_{\delta\in\mathcal{N}_0,s}\kappa(p'_s,\delta)$, $\tau_0 = \min_{\delta\in\mathcal{N}_0}\lambda_{\min}[\Delta^2\mathcal{L}(\delta)]$, $\mathcal{L}(\delta) = n\sum_{s=1}^h \log\|T_s - \mathbf{X}\delta\|^2$, and $\mathcal{N}_0 = \{\delta \in \mathbf{R}^{d_0} : \|\delta - B^{(0)}_{1,s}\|_\infty \leq d_n\}$. $\lambda_{\min}[\Delta^2\mathcal{L}(\delta)]$ is the smallest eigenvalue across all of the second derivative matrices of $\mathcal{L}(\delta)$.

The log penalty that we use for this dissertation satisfies condition 1.3, 2.1, 2.2, and 3. Though to keep our results more general for any penalty functions that may satisfy these conditions, we state them explicitly. Recall that the coefficient matrix $\mathbf{B}$ is an $h\times p$ matrix. The j-th row of $\mathbf{B}$, denoted by $\mathbf{b}_j$, includes the h regression coefficients for the j-th covariate; and the s-th column of $\mathbf{B}$, denoted by B_s, includes the p regression coefficients for T_s, the s-th transformation of y.

Recall that the coefficient matrix $\mathbf{B}$ is an $h \times p$ matrix. The j-th row of $\mathbf{B}$, denoted by $\mathbf{b}_j$, includes the h regression coefficients for the j-th covariate; and the s-th column of $\mathbf{B}$, denoted by B_s, includes the p regression coefficients for T_s, the s-th transformation of y.

Lemma 6. Estimating the $\mathbf{B} = \{B_1, ..., B_h\}$ which minimizes the mGAP objective function is equivalent to:

$$\text{argmin}_{\mathbf{B}}\left\{n\sum_{s=1}^h \log\|T_s - \mathbf{X}B_s\|^2 + \lambda\sum_{j=1}^p \log(\|b_j\| + \tau)\right\} \tag{4.2}$$

Proof of Lemma 6

The mGAP objective function is:

$$\text{argmin}_{\mathbf{B},\omega,\kappa}\left\{\sum_{s=1}^{h}\frac{\|T_s-\mathbf{X}B_s\|^2}{\omega_s}+\lambda\sum_{j=1}^{p}\frac{\|b_j\|+\tau}{\kappa_j}+\lambda\sum_{j=1}^{p}\log(\kappa_j)+n\sum_{s=1}^{h}\log(\omega_s)\right\} \tag{4.3}$$

with tuning parameters λ and τ. First, we note that if we optimize over κ and ω first while holding $\mathbf{B}=(B_1,...,B_h)$ constant, the solutions are: $\hat{\kappa}_j=\|b_j\|+\tau$ and $\hat{\omega}_s=\frac{\|T_s-\mathbf{X}B_s\|^2}{n}$. Substituting $\hat{\kappa}_j$ and $\hat{\omega}_s$ into equation (4.3) gives us:

$$nh+\lambda p+\lambda\sum_{j=1}^{p}\log(\|b_j\|+\tau)+n\sum_{s=1}^{h}\log\|T_s-\mathbf{X}B_s\|^2-nh\log n \tag{4.4}$$

Since n,h,p and λ are all constants with respect to the parameter of interest, $\mathbf{B}$, we can say that optimizing the objective function in equation (4.3) is equivalent to optimizing:

$$\mathcal{O}(\mathbf{B})=n\sum_{s=1}^{h}\log\|T_s-\mathbf{X}B_s\|^2+\lambda\sum_{j=1}^{p}\log(\|b_j\|+\tau) \tag{4.5}$$

Lemma 7. Let $\hat{\mathbf{B}}=(\hat{B}_1,....,\hat{B}_h)$ be an estimator of $\mathbf{B}=(B_1,....,B_h)$ by minimizing equation (4.2). Then $\hat{B}_s$ satisfies the following conditions for all s:

$$\frac{2n\left[\mathbf{X}_1^{\intercal}(\mathbf{X}_1\hat{B}_{1,s})-\mathbf{X}_1^{\intercal}T_s\right]}{\|T_s-\mathbf{X}_1\hat{B}_{1,s}\|^2}+p_s'(\hat{\mathbf{B}}_1) = \mathbf{0}_{d_0\times 1} \tag{4.6}$$

$$\left\|\frac{2n\left[\mathbf{X}_2^{\intercal}(\mathbf{X}_2\hat{B}_{2,s})-\mathbf{X}_2^{\intercal}T_s\right]}{\|T_s-\mathbf{X}_2\hat{B}_{2,s}\|^2}\right\|_{\infty} = \mathbf{0}_{(p-d_0)\times 1} \tag{4.7}$$

$$\lambda_{\min}\{\nabla^2\mathcal{L}(\hat{\mathbf{B}}_1)\} > \max_s\left(\lambda\kappa(p_s';\hat{\mathbf{B}}_1)\right), \tag{4.8}$$

where $p_s'(\hat{\mathbf{B}}_1)=\frac{\partial p(\mathbf{B}_1)}{\partial B_{1,s}}|_{\mathbf{B}_1=\hat{\mathbf{B}}_1}$ is a vector of length d_0, with j-th element being $\frac{\partial p(\mathbf{B}_1)}{\partial B_{js}}|_{\mathbf{B}_1=\hat{\mathbf{B}}_1}$, $\lambda_{\min}\{\cdot\}$ indicates the smallest eigenvalue of the matrix,

$\kappa(p'_s;\nu) = \lim_{\epsilon\to 0}\max_j \sup_{t_1<t_2\in(|v_j|-\epsilon,|v_j|+\epsilon)} -\frac{p'_s(t_2)-p'_s(t_1)}{t_2-t_1}$ is the local concavity of the penalty around ν, and $p'_s(t_1)$ or $p'_s(t_2)$ indicates the derivative with respect to B_{js}, an element of $\mathbf{B}$.

Proof of Lemma 7

The objective function from equation (4.2) is a constrained optimization problem which we can break down into our loss function and our penalty or constraint functions. Let

$$\mathcal{O}(\mathbf{B}) = n\sum_{s=1}^{h}\log\|T_s - \mathbf{X}B_s\|^2 + \lambda\sum_{j=1}^{p}\log(\|\mathbf{b}_j\|+\tau) = \mathcal{L}(\mathbf{B}) + p(\mathbf{B}), \tag{4.9}$$

where $p(\mathbf{B}) = \lambda\sum_{j=1}^{p}\log(\|\mathbf{b}_j\|+\tau)$.

By the Karush-Kuhn-Tucker conditions, we know that the minimizer $\hat{\mathbf{B}} = [\hat{B}_1, ..., \hat{B}_p]$ satisfies the first order necessary condition:

$$\frac{\partial\mathcal{L}(\mathbf{B})}{\partial B_{js}} + \frac{\partial p(\mathbf{B})}{\partial B_{js}}\bigg\|_{\mathbf{B}=\hat{\mathbf{B}}} = 0, \tag{4.10}$$

where

$$\frac{\partial\mathcal{L}(\mathbf{B})}{\partial B_{js}} = -2n\frac{X_j^\intercal(T_s - \mathbf{X}B_s)}{\|T_s - \mathbf{X}B_s\|^2}\text{ , and }\quad \frac{\partial p(\mathbf{B})}{\partial B_{js}} = \frac{\lambda B_{js}}{\|\mathbf{b}_j\|(\|\mathbf{b}_j\|+\tau)}. \tag{4.11}$$

Notice then, that we can express the set of equations (4.10) as:

$$\frac{2n\left(\mathbf{X}^\intercal(\mathbf{X}\hat{B}_s) - \mathbf{X}^\intercal T_s\right)}{\|T_s - \mathbf{X}\hat{B}_s\|^2} + p'_s(\hat{\mathbf{B}}) = \mathbf{0}, \quad \forall\, s\in\{1,2.,\ldots,h\}, \tag{4.12}$$

where $p'_s(\hat{\mathbf{B}}) = \frac{\partial p(\mathbf{B})}{\partial B_s}\|_{\mathbf{B}=\hat{\mathbf{B}}}$ is a vector of length p, with j-th element being $\frac{\partial p(\mathbf{B})}{\partial B_{js}}\|_{\mathbf{B}=\hat{\mathbf{B}}}$.

For a consistent estimate, $p'_s(\mathbf{B}_2^{(0)}) = \mathbf{0}\ \forall s$. Therefore, we can rewrite equation (4.12)

into its consistency and sparsity necessary conditions:

$$\frac{2n\left(\mathbf{X}_1^\intercal(\mathbf{X}_1 B_{1,s}^{(0)}) - \mathbf{X}_1^\intercal T_s\right)}{\|T_s - \mathbf{X}_1 B_{1,s}^{(0)}\|^2} + p_s'(\mathbf{B}_1^{(0)}) = \mathbf{0} \tag{4.13}$$

$$\left\| \frac{2n\left(\mathbf{X}_2^\intercal(\mathbf{X}_2 B_{2,s}^{(0)}) - \mathbf{X}_2^\intercal T_s\right)}{\|T_s - \mathbf{X}_2 B_{2,s}^{(0)}\|^2} \right\|_\infty = \mathbf{0} \tag{4.14}$$

Finally, since our penalty function $p(\|\mathbf{b}_j\|)$ is not convex, and thus our objective function may not be convex. Therefore, in addition to the first order condition, the estimator must also satisfy the second order sufficient condition that $\nabla^2\mathcal{O}(\mathbf{B}^{(0)})$ is positive definite. This can be written as:

$$\lambda_{\min}\{\nabla^2\mathcal{L}(\mathbf{B}_1^{(0)})\} > \lambda\kappa(p_s'; \mathbf{B}_1^{(0)}) \tag{4.15}$$

where, $\lambda_{\min}$ indicates the smallest eigenvalue across the second derivative matrices for each slice and

$$\kappa(p_s'; \nu) = \lim_{\epsilon \to 0} \max_{1 \le j \le |S|} \sup_{t_1 < t_2 \in (|v_j| - \epsilon, |v_j| + \epsilon)} -\frac{p_s'(t_2) - p_s'(t_1)}{t_2 - t_1}$$

is the local concavity of the penalty around ν.

Theorem 2. Given the conditions 1.1 to 1.4, with probability at least $1 - 2\left[d_0 n^{-1} + (p - d_0)\exp\left(-n^\alpha \log n\right)\right]$, there exists a penalized likelihood estimator $\hat{\mathbf{B}} = (\hat{B}_1, ..., \hat{B}_h)$ where $\hat{B}_s = (\hat{B}_{1,s}^\intercal, \hat{B}_{2,s}^\intercal)^\intercal$ which satisfies:

(1) Sparsity: $P(\hat{B}_{2,s} = \mathbf{0}) \to 1 \; \forall \; s$

(2) L_∞ loss: $\|\hat{B}_{1,s} - B_{1(0),s}\|_\infty = o_p(n^{-\gamma_0}\sqrt{\log n}) \; \forall \; s$.

Proof of Theorem 2

From Lemma 6, we will be using equation (4.2) as our objective function of interest.

Second, we reiterate the conditions for the estimator of B from Lemma 7:

$$\begin{aligned}
\frac{2n\left[\mathbf{X}_1^\intercal(\mathbf{X}_1\hat{B}_{1,s}) - \mathbf{X}_1^\intercal T_s\right]}{\|T_s - \mathbf{X}_1\hat{B}_{1,s}\|^2} + p_s'(\hat{\mathbf{B}}_1) &= \mathbf{0}_{d_0\times 1} \\
\left\| \frac{2n\left[\mathbf{X}_2^\intercal(\mathbf{X}_2\hat{B}_{2,s}) - \mathbf{X}_2^\intercal T_s\right]}{\|T_s - \mathbf{X}_2\hat{B}_{2,s}\|^2} \right\|_\infty &= \mathbf{0}_{(p-d_0)\times 1} \\
\lambda_{\min}\{\nabla^2\mathcal{L}(\hat{\mathbf{B}}_1)\} &> \max_s\left(\lambda\kappa(p_s';\hat{\mathbf{B}}_1)\right),
\end{aligned}$$

Now, let us consider the following set of events. Let $\boldsymbol{\xi_s} = (\xi_{s1},...,\xi_{sp})^\intercal = \mathbf{X}^\intercal(T_s - \mathbf{X}^\intercal B_s^{(0)}) = \mathbf{X}^\intercal\boldsymbol{\epsilon_s}$, and the events be:

$$\mathcal{E}_{1,s} = \left\{\|\boldsymbol{\xi}_{1,s}\|_\infty \le \sigma_s^{-1/2}n^{1/2}\sqrt{\log n}\right\} \quad \text{and} \quad \mathcal{E}_{2,s} = \left\{\|\boldsymbol{\xi}_{2,s}\|_\infty \le \sigma_s^{-1/2}n^{1/2}\sqrt{n^\alpha \log n}\right\}$$

where $\boldsymbol{\xi}_{1,s} = (\xi_{s1},...,\xi_{sd_0})^\intercal$, $\boldsymbol{\xi}_{2,s} = (\xi_{s(d_0+1)},...,\xi_{sp})^\intercal$. Ultimately, our goal is to show that the estimator which satisfies equations (4.6)-(4.8) falls into the intersection of all of these events with probability converging to 1. First, we show that the intersection of all these events converge in probability. For this, we need the following result:

Proposition 4 from (Fan and Lv 2011) using *Hoeffding's Inequality*:

Assuming $T_s = (T_{1s},...,T_{ns})^\intercal$ are bounded in $[c,d]$. Let $a = (a_1,...,a_n)^\intercal$. Then $\mathbf{P}\left(|a^\intercal(T_s - \mathbf{X}B_s^{(0)})| > ||a||^2\epsilon\right) \le 2e^{-\sigma_s\epsilon^2}$, where $\epsilon \in (0,\infty)$ and $\sigma_s = 2/(d-c)^2$.

The assumption of boundedness depends on the transformation applied to the response. Our current application uses the B-spline transformation (where T_s is the s-th basis of B-spline transformation of response variable Y) or the slice inverse transformation (where $t_{sk} = 1$ if Y_k belongs to the s-th slice and 0 otherwise) which conforms to this assumption.

Consider when $a = X_j$ for $j = 1,...,p$, and thus $\|a\|^2 = \|X_j\|^2 = \sqrt{n}$. Then, to

apply the above inequality to $\mathcal{E}_{1,s}$ and $\mathcal{E}_{2,s}$, let $\epsilon = \sigma_s^{-1/2}\sqrt{\log n}$ and $\epsilon = \sigma_s^{-1/2}\sqrt{n^\alpha \log n}$, respectively, such that:

$$\begin{aligned}
\mathbf{P}\left(|X_j^\intercal(T_s - \mathbf{X}B_s^{(0)})| > \|X_j\|^2\epsilon\right) &\leq 2e^{-\sigma_s\epsilon^2} \\
&\Rightarrow P\left(|\xi_j| > \sigma_s^{-1/2}n^{1/2}\sqrt{\log n}\right) < 2n^{-1}, \\
&\quad \text{for } j = 1, ..., d_0. \\
\mathbf{P}\left(|X_j^\intercal(T_s - \mathbf{X}B_s^{(0)})| > \|X_j\|^2\ \epsilon\right) &\leq 2e^{-\sigma_s\epsilon^2} \\
&\Rightarrow P\left(|\xi_j| > \sigma_s^{-1/2}n^{1/2}\sqrt{n^\alpha \log n}\right) < 2e^{-(n^\alpha \log n)}, \\
&\quad \text{for } j = d_0 + 1, ..., p.
\end{aligned}$$

Then, by Bonferroni's inequality

$$\begin{aligned}
P(\mathcal{E}_{1,1} \cap ... \cap \mathcal{E}_{1,h} \cap \mathcal{E}_{2,1} \cap ... \cap \mathcal{E}_{2,h}) &\geq 1 - h\sum_{j=1}^{d_0} \mathbf{P}\left(|\xi_j| > \sigma_s^{-1/2}n^{1/2}\sqrt{\log n}\right) \\
&\quad -h\sum_{j=d_0+1}^{p} \mathbf{P}\left(|\xi_j| > \sigma_s^{-1/2}n^{1/2}\sqrt{n^\alpha \log n}\right) \\
&\geq 1 - 2h\left[d_0 n^{-1} + (p - d_0)e^{-(n^\alpha \log n)}\right].
\end{aligned}$$

Recall that true coefficients $B_s^{(0)} \in \mathbf{R}^p$ can be split into two parts: $B_{1,s}^{(0)}$, and $B_{2,s}^{(0)}$, where $B_{1,s}^{(0)} \in \mathbf{R}^{d_0}$ and $B_{2,s}^{(0)} = \mathbf{0} \in \mathbf{R}^{p-d_0}$. Let $\hat{B}_s = (\hat{B}_{1,s}^\intercal, \hat{B}_{2,s}^\intercal)^\intercal \in \mathbf{R}^p$ be the estimated coefficients for transformation s. We seek to prove that under the event $\mathcal{E}_{11} \cap ... \cap \mathcal{E}_{1h} \cap \mathcal{E}_{21} \cap ... \cap \mathcal{E}_{2h}$, there is an estimate such that $\hat{B}_{2,s} = \mathbf{0}$, and $\hat{B}_{1,s} \to B_1^{(0)}$ by L_∞ norm.

Step 1: Consistency in the d_0-dimensional subspace

For sufficiently large n, we first examine the existence of a solution $\hat{B}_{1,s} \in \mathbf{R}^{d_0}$ for

equation (4.6) inside the hypercube

$$\mathcal{N} = \left\{ \delta \in \mathbf{R}^{d_0} : \|\boldsymbol{\delta} - B_{1,s}^{(0)}\|_\infty = d_n = O_p(n^{-\gamma_0}\sqrt{\log n}) \right\}. \tag{4.16}$$

Define

$$\begin{aligned}
\varphi_s(\delta) &= \frac{2\left\{(\mathbf{X}_1\delta)^\intercal \mathbf{X}_1 - T_s^\intercal \mathbf{X}_1\right\}}{\|T_s - \mathbf{X}_1\delta\|^2} + p_s'(\delta) \\
&= \left[\frac{2}{\|T_s - \mathbf{X}_1\delta\|^2}\left\{(\mathbf{X}_1\delta)^\intercal \mathbf{X}_1 - T_s^\intercal \mathbf{X}_1 + (\mathbf{X}_1 B_{1,s}^{(0)})^\intercal \mathbf{X}_1 - (\mathbf{X}_1 B_{1,s}^{(0)})^\intercal \mathbf{X}_1\right\}\right] + p_s'(\delta) \\
&= \frac{2}{\|T_s - \mathbf{X}_1\delta\|^2}\left\{\left(\delta^\intercal - B_{1,s}^{(0)\intercal}\right)(\mathbf{X}_1^\intercal \mathbf{X}_1) - \xi_{1s}^\intercal\right\} + p_s'(\delta).
\end{aligned}$$

Thus equation (4.6) is equivalent to $\varphi_s(\delta) = 0$. Next, we show that there exists a solution inside the hypercube $\mathcal{N}$ to satisfy $\varphi_s(\boldsymbol{\delta}) = 0$. Let

$$\bar{\varphi_s}(\delta) = \frac{\|T_s - \mathbf{X}_1\delta\|^2}{2\mathbf{X}_1^\intercal \mathbf{X}_1}\varphi_s(\delta) = \delta - B_{1,s}^{(0)\intercal} + \mathbf{u}_s,$$

where $\mathbf{u}_s = -\frac{\|T_s - \mathbf{X}_1\delta\|^2}{2\mathbf{X}_1^\intercal \mathbf{X}_1}\left[\frac{\xi_{1s}^\intercal}{\|T_s - \mathbf{X}_1\delta\|^2} + p_s'(\delta)\right]$. Since, for any $\delta \in \mathcal{N}$, $|\delta_j| \geq |b_{j,s}^{(0)}| - d_n$, we have

$$\min_{j=1,\ldots,d_0} |\delta_j| \geq \min_{j=1,\ldots,d_0} |b_{j,s}^{(0)}| - d_n = d_n.$$

By the monotonicity of $p_s'(\delta)$, $\|p_s'(\delta)\|_\infty \leq p_s'(d_n)$, and thus with the definition of $\mathcal{E}_1$

$$\left\| \frac{\xi_{1s}^\intercal}{\|T_s - \mathbf{X}_1\delta\|^2} + p_s'(\delta) \right\|_\infty \leq \frac{\sigma_s^{-1/2} n^{\frac{1}{2}}\sqrt{\log n}}{\|T_s - \mathbf{X}_1\delta\|^2} + p_s'(d_n).$$

Therefore,

$$\begin{aligned}
\|\mathbf{u}_s\|_\infty &\leq \frac{\|T_s - \mathbf{X}_1\delta\|^2}{\|2\mathbf{X}_1^\intercal\mathbf{X}_1\|_\infty}\left\|\frac{\xi_{1s}^\intercal}{\|T_s - \mathbf{X}_1\delta\|^2} + p_s'(\delta)\right\|_\infty \\
&= \frac{1}{\|2\mathbf{X}_1^\intercal\mathbf{X}_1\|_\infty}\left[\sigma_s^{-1/2}n^{\frac{1}{2}}\sqrt{\log n} + \|T_s - \mathbf{X}_1\delta\|^2 p_s'(d_n)\right] \\
&= \frac{1}{2d_0 n}\left[\sigma_s^{-1/2}n^{\frac{1}{2}}\sqrt{\log n} + \|T_s - \mathbf{X}_1\delta\|^2 p_s'(d_n)\right].
\end{aligned}$$

Now, notice that within our hypercube:

$$\begin{aligned}
\|T_s - \mathbf{X}_1\delta\|^2 &\leq \|T_s - \mathbf{X}_1 B_{1,s}^{(0)}\|^2 + \|\mathbf{X}_1(B_{1,s}^{(0)} - \delta)\|^2 \\
&\leq \|T_s - \mathbf{X}_1 B_{1,s}^{(0)}\|^2 + d_n d_0 n.
\end{aligned}$$

$\|T_s - \mathbf{X}_1 B_{1,s}^{(0)}\|^2 = \sum_{i=1}^n (T_{is} - \mathbf{X}_{1,i}B_{1,s}^{(0)})^2$, where $\mathbf{X}_{1,i}$ denotes the i-th row of $\mathbf{X}_1$. We assume $T_{is} - \mathbf{X}_{1,i}B_{1,s}^{(0)}$ is bounded, which is an implicit assumption for any regression problem, and thus $\|T_s - \mathbf{X}_1 B_{1,s}^{(0)}\|^2 = O_p(n)$. Hence, we get:

$$\|\mathbf{u}\|_\infty = O_p\left[d_0^{-1}n^{-\frac{1}{2}}\sigma_s^{-\frac{1}{2}}\sqrt{\log n} + d_0^{-1}p_s'(d_n) + d_n p_s'(d_n)\right]$$

Then, from conditions (1.1 to 1.3), we can see that each term is of order $o(n^{-\gamma_0}\sqrt{\log n})$, hence $\|\mathbf{u}\|_\infty = o(n^{-\gamma_0}\sqrt{\log n})$.

For a constant $C > 0$ and sufficiently large n, if $\delta_j - B_{js} = Cn^{-\gamma_0}\sqrt{\log n}$ then $\bar{\varphi}_{js}(\delta) \geq Cn^{-\gamma_0}\sqrt{\log n} - \|\mathbf{u}\|_\infty \geq 0$. Similarly, if $\delta_j - B_{js} = -Cn^{-\gamma_0}\sqrt{\log n}$ then $\bar{\varphi}_{js}(\delta) \leq -Cn^{-\gamma_0}\sqrt{\log n} + \|\mathbf{u}\|_\infty \leq 0$.

Then, by the continuity of function $\bar{\varphi}(\delta) = (\bar{\varphi}_1(\delta), ..., \bar{\varphi}_s(\delta))$ and Miranda's existence theorem, there is a solution $\hat{B}_{1,s}$ for $\bar{\varphi}(\delta) = \mathbf{0}$ in $\mathcal{N}$, hence a solution for equation (4.6) in $\mathcal{N}$. Since the above conclusion holds for any $C > 0$, we conclude that $\|\boldsymbol{\delta} - B_1^{(0)}\|_\infty = o(n^{-\gamma_0}\sqrt{\log n})$.

Step 2: Sparsity

Let $\hat{B}_s \in \mathbf{R}^p$ with $\hat{B}_{1,s} \in \mathcal{N}$ as the solution for equation (4.6) and $\hat{B}_{2,s} = 0$. To prove the sparsity, we verify that equation (4.7) holds for $\hat{B}_s$.

$$\left\| \frac{2\left((\mathbf{X}_2 B_{2,s})^\intercal \mathbf{X}_2 - T_s^\intercal \mathbf{X}_2\right)}{\|T_s - \mathbf{X}B_s\|^2} \right\|_\infty \quad < \quad p_s'(0+)$$

under the event $\mathcal{E}_1 \cap \mathcal{E}_2$.

Similarly to Step 1, we first rewrite $2\left[(\mathbf{X}_2\hat{B}_{2,s})^\intercal \mathbf{X}_2 - T_s^\intercal \mathbf{X}_2\right]/\|T_s - \mathbf{X}\hat{B}_s\|^2$ in terms of $\boldsymbol{\xi}_{2,s}$:

$$\begin{aligned} \frac{2\left[(\mathbf{X}_2\hat{B}_{2,s})^\intercal \mathbf{X}_2 - T_s^\intercal \mathbf{X}_2\right]}{\|T_s - \mathbf{X}\hat{B}_s\|^2} &= \frac{2}{\|T_s - \mathbf{X}\hat{B}_s\|^2}\left[\mathbf{X}_2^\intercal(T_s - \mathbf{X}_1 B_{1,s}^{(0)}) - \mathbf{X}_2^\intercal(\mathbf{X}_1\hat{B}_{s,1} - \mathbf{X}_1 B_{1,s}^{(0)})\right] \\ &= \frac{2}{\|T_s - \mathbf{X}\hat{B}_s\|^2}\left[\boldsymbol{\xi}_{2,s} - \mathbf{X}_2^\intercal \mathbf{X}_1(\hat{B}_{s,1} - B_{1,s}^{(0)})\right]. \end{aligned}$$

First, from Condition 2.1, we can see that: $\frac{\|\hat{\xi}_{2,s}\|_\infty}{\|T_s - \mathbf{X}\hat{B}_s\|^2} = \frac{\sigma^{-\frac{1}{2}} n^{\frac{1+\alpha}{2}}\sqrt{\log n}}{\|T_s - \mathbf{X}\hat{B}_s\|^2} < (1 - K)p_s'(0+)$. Now, to examine the second term $\frac{2}{\|T_s - \mathbf{X}\hat{B}_s\|^2}\mathbf{X}_2^\intercal \mathbf{X}_1(\hat{B}_{s,1} - B_{1,s}^{(0)})$:

From Step 1, we have:

$$\hat{B}_{1,s} - B_{1,s}^{(0)} = (2\mathbf{X}_1^\intercal \mathbf{X}_1)^{-1}\left[\mathbf{X}_1^\intercal \hat{\epsilon}_1 + \|T_s - \mathbf{X}\hat{B}_s\|^2 p(\hat{B}_{s,1})\right]. \tag{4.17}$$

Therefore, by conditions 2.1 and 2.2,

$$\begin{aligned}\left\|\mathbf{X}_2^\intercal\mathbf{X}_1(\hat{B}_{1,s}-B_{1,s}^{(0)})\right\|_\infty &= \frac{\|\mathbf{X}_2^\intercal\mathbf{X}_1\|_\infty}{\|\mathbf{X}_1^\intercal\mathbf{X}_1\|_\infty}\left\{\hat{\xi}_{1,s}+\|T_s-\mathbf{X}\hat{B}_s\|^2\|p_s'(\hat{B}_s)\|_\infty\right\}\\ &\leq O_p(n^{\frac{1}{2}+\nu}\sqrt{\log n})+\frac{\|\mathbf{X}_2^\intercal\mathbf{X}_1\|_\infty}{\|\mathbf{X}_1^\intercal\mathbf{X}_1\|_\infty}\|T_s-\mathbf{X}\hat{B}_s\|^2p_s'(d_n)\\ &\leq Kp_s'(0+).\end{aligned}$$

Then, we have

$$\left\|\frac{2\left((\mathbf{X}_2B_{2,s})^\intercal\mathbf{X}_2-T_s^\intercal\mathbf{X}_2\right)}{\|T_s-\mathbf{X}B_s\|^2}\right\|_\infty < (1-K)p_s'(0+)+Kp_s'(0+)=p_s'(0+).$$

Therefore for sufficiently large n, equation (4.7) holds.

Finally, equation (4.8) would be satisfied by condition 3 for sufficiently large n.

B.1.1 Consistency of modified-PC Algorithm

Lemma 8.

For any $\gamma>0$, $\sup_{i,j,\mathbf{s}\in\Pi_{i,j}}\mathbf{P}\left[|T_{CI}-\mathbb{E}(T_{CI})|>\gamma\right]\leq\exp\left\{-\frac{2n\gamma^2}{R^4}\right\}$. Where R^2 is the largest possible element of $\tilde{\mathbf{K}}_{X_{i,\mathbf{s}}|X_\mathbf{s}}\tilde{\mathbf{K}}_{X_j|X_\mathbf{s}}$.

Proof of Lemma 8

Recall that $T_{CI}^{(i,j|k)}=\frac{1}{n}Tr(\tilde{\mathbf{K}}_{X_{i,\mathbf{k}}|X_\mathbf{k}}\tilde{\mathbf{K}}_{X_j|X_\mathbf{k}})=\frac{1}{n}\sum_l^n\lambda_l(\tilde{\mathbf{K}}_{X_{i,\mathbf{k}}|X_\mathbf{k}}\tilde{\mathbf{K}}_{X_j|X_\mathbf{k}})$, where $\lambda_l(\cdot)$ are the $l=1...n$ eigenvalues ranked in order of largest to smallest. In addition, from [Shawe-Taylor et al. (2005)] we know the following probabilistic error bound of eigenvalues of a kernel matrix, where for any $\gamma>0$:

$$P\left\{\left|\frac{1}{n}\sum_{l=1}^n\lambda_l(K)-\frac{1}{n}\sum_{l=1}^n\mathbb{E}(\lambda_l(K))\right|>\gamma\right\}\leq\exp\left\{-\frac{2n\gamma^2}{R^4}\right\} \tag{4.18}$$

Then,

$$\begin{aligned} & P\left\{\left|T_{CI}^{(i,j|k)} - \mathbb{E}(T_{CI}^{(i,j|k)})\right| > \gamma\right\} \\ = \;& P\left\{\left|\frac{1}{n}\sum_{l}^{n}\lambda_l(\tilde{\mathbf{K}}_{X_{i,\mathbf{k}}|X_{\mathbf{k}}}\tilde{\mathbf{K}}_{X_j|X_{\mathbf{k}}}) - \frac{1}{n}\sum_{l}^{n}\mathbb{E}(\lambda_l(\tilde{\mathbf{K}}_{X_{i,\mathbf{k}}|X_{\mathbf{k}}}\tilde{\mathbf{K}}_{X_j|X_{\mathbf{k}}}))\right| > \gamma\right\} \\ \leq \;& \exp\left\{-\frac{2n\gamma^2}{R^4}\right\} \end{aligned}$$

Theorem 4. Assume a perfect estimation of $\mathcal{G}_{\mathcal{M}}$ in Step 1 as well as conditions (1.1) and (2.3). Let the estimate from Step 2 be $\hat{\mathcal{G}}^{\alpha_n}_{\text{skel},n}$, where α_n is the significance level used in the conditional independence testing for Step 2. Then, there exists an $\alpha_n \to_{n\to\infty} 0$ such that:

$$\mathbf{P}[\hat{\mathcal{G}}_{\text{skel},n}(\alpha_n) = \mathcal{G}_{\text{skel},n}] = 1 - O\left(\exp(n^{\alpha} - C(n^{2(1-d)})\right) \to_{n\to\infty} 1$$

for some constant $C > 0$.

Proof of Theorem 4 For the proof of Theorem 4, recall the following condition:

Condition 2.3 If $X_i \perp X_j | X_{\mathbf{s}}$, we put a lower bound on the size of $\inf_{i,j|\mathbf{s}} T_{CI} \geq c_n$, where $c_n = O(n^{\frac{1}{2}-d})$; $0 < d < 1/2$.

An error occurs in the modified PC-algorithm when either a Type I or Type II error occurs during the conditional test for nodes of edge (i, j) with conditioning set $\mathbf{k} \in \mathbf{\Pi}_{i,j}$. Denote these errors as $E_{i,j|\mathbf{k}} = E^{I}_{i,j|\mathbf{k}} \cup E^{II}_{i,j|\mathbf{k}}$, then:

$$\begin{aligned} \mathbf{P}\left[\text{An Error Occurs in Step 2}\right] &\leq \mathbf{P}\left[\bigcup_{i,j|\mathbf{k}\in\mathbf{\Pi}_{i,j}}\left(E_{i,j|\mathbf{k}}\right)\right] \\ &\leq 2^{2q_n}\sup_{i,j|\mathbf{k}\in\mathbf{\Pi}_{i,j}}\mathbf{P}[E_{i,j|\mathbf{k}}]. \end{aligned}$$

The above inequality can be derived using the cardinality of the set $\mathbf{\Pi}_{i,j}$. Recall that $\mathbf{\Pi}_{i,j} = \{A_{\mathcal{G},i,j} \backslash D_{\mathcal{G},i,j}, D_{\mathcal{G},i,j} \subseteq C_{\mathcal{G},i,j}\}$ (at least one of the set in $\mathbf{\Pi}_{i,j}$ includes all common parents of X_i and X_j, but excludes any common descendants). Therefore $|\mathbf{\Pi}_{i,j}| \leq |\mathcal{P}(A)_{\mathcal{G},i,j}| \leq 2^{|\text{adj}(\mathcal{G},X_i)|+|\text{adj}(\mathcal{G},X_j)|} \leq 2^{2q_n}$, where $\mathcal{P}(\cdot)$ is the power set, q_n is the maximum degree of any one node.

Let the cdf of $\check{T}_{CI}$ be $F(\cdot)$. Then,

$$\begin{aligned} E^{I}_{i,j|\mathbf{k}} &: T_{CI} > F^{-1}(1-\alpha_n) \quad ; \quad X_i \perp X_j | X_{\mathbf{k}}. \\ E^{II}_{i,j|\mathbf{k}} &: T_{CI} \leq F^{-1}(1-\alpha_n) \quad ; \quad X_i \not\perp X_j | X_{\mathbf{k}}. \end{aligned}$$

Choose $\alpha_n = 1 - F\left(c_n/2\right)$, then using Lemma 8 and the assumption that $\inf_{i,j|\mathbf{k}} T^{(i,j|k)}_{CI} \geq c_n$ if $X_i \not\perp X_j | X_{\mathbf{k}}$, we can get:

$$\begin{aligned} \sup_{i,j|\mathbf{k} \in K^{m_n}_{ij}} \mathbf{P}[E^{I}_{i,j|\mathbf{k}}] &= \sup_{i,j|\mathbf{k} \in K^{m_n}_{ij}} \mathbf{P}\left[T^{(i,j|k)}_{CI} > \frac{c_n}{2}\right] \\ &= \sup_{i,j|\mathbf{k} \in K^{m_n}_{ij}} \mathbf{P}\left[\left|T^{(i,j|k)}_{CI} - \mathbb{E}\left(T^{(i,j|k)}_{CI}\right)\right| > \frac{c_n}{2} - \frac{1}{n} Tr(\check{\mathbf{w}}\check{\mathbf{w}}^{\intercal})\right] \\ &\leq \exp\left\{-\frac{2n\left[c_n/2 - (1/n)Tr(\check{\mathbf{w}}\check{\mathbf{w}}^{\intercal})\right]^2}{R^4}\right\}, \end{aligned}$$

and

$$\begin{aligned} \sup_{i,j|\mathbf{k} \in K^{m_n}_{ij}} \mathbf{P}[E^{II}_{i,j|\mathbf{k}}] &= \sup_{i,j|\mathbf{k} \in K^{m_n}_{ij}} \mathbf{P}\left[T^{(i,j|k)}_{CI} \leq \frac{c_n}{2}\right] \\ &\leq \sup_{i,j|\mathbf{k} \in K^{m_n}_{ij}} \mathbf{P}\left[\left|T^{(i,j|k)}_{CI} - \mathbb{E}\left(T^{(i,j|k)}_{CI}\right)\right| > \frac{c_n}{2}\right] \\ &\leq \exp\left\{-\frac{nc_n^2}{2R^4}\right\}. \end{aligned}$$

We note that R^4 is not dependent on n and assume it is bounded by some constant. For example, for the Gaussian kernel, it is fixed at 1. Secondly, assume $Tr(\check{\mathbf{w}}\check{\mathbf{w}}^{\intercal}) = n^{\delta}$,

$(1/2 - d) > \delta - 1$, i.e., $\delta < 3/2 - d$. Then this term is dominated by $c_n/2$. Therefore, we can write:

$$\sup_{i,j|\mathbf{k}\in K_{ij}^{m_n}} \mathbf{P}[E_{i,j|\mathbf{k}}^{I}] \leq O(\exp\left\{-C_1(nc_n^2)\right\}) \tag{4.19}$$

for some a positive constant C_1 that's not dependent on n. Similarly, we have:

$$\sup_{i,j|\mathbf{k}\in K_{ij}^{m_n}} \mathbf{P}[E_{i,j|\mathbf{k}}^{II}] \leq O(\exp\left\{-C_2(nc_n^2)\right\}) \tag{4.20}$$

Then, using the assumptions: 1) $\log(p) = O(n^\alpha)$ for some $0 < \alpha < 1$, 2) $c_n^{-1} = O(n^{\frac{1}{2}-d})$ for some $0 < d < 1 - \frac{\alpha}{2}$ and 3) $q_n = \max(d_0) = O(n^\nu) : 0 \leq \nu < \frac{1}{2}$:

$$\begin{aligned}
\mathbf{P}\left[\text{An Error Occurs in Step 2}\right] &\leq O\left(pq_n 2^{2q_n}\exp(-C_3(nc_n^2))\right) \\
&\leq O\left(n^\nu \exp(2n^\nu + n^\alpha - C_3(n^{2(1-d)})\right) = o(1)
\end{aligned}$$

since $2(1-d) > 1 > \max(\alpha, \nu)$, and thus $n^{2(1-d)}$ dominates all other terms.

APPENDIX C: DERIVATION OF PENALTY WEIGHTS FOR SCZINB

We calculate second derivative for the following log-likelihood function.

$$\mathcal{L} = \Sigma_{i=1}^{n} \left\{ \log \left[\pi_i I(y_i = 0) + (1 - \pi_i) f_{\text{NB}}(y_i; \mu_i, \phi) \right] \right\} \tag{4.21}$$

C.1 When $y_i = 0$

Derivatives for β:

$$\begin{aligned} \mathcal{L}_i|_{y_i=0} &= \log[\pi_i + (1 - \pi_i) f_{\text{NB}}(0; \mu_i, \phi)] \\ \frac{\partial}{\partial \beta_k} \mathcal{L}_i|_{y_i=0} &= \frac{1-\pi_i}{\pi_i + (1-\pi_i) f_{\text{NB}}(0; \mu_i, \phi)} \frac{\partial}{\partial \beta_k} f_{\text{NB}}(0; \mu_i, \phi) \end{aligned} \tag{4.22}$$

Since,

$$\begin{aligned} \frac{\partial}{\partial \beta_k} f_{\text{NB}}(0; \mu_i, \phi) &= \frac{\partial (1 + \phi \mu_i)^{-\frac{1}{\phi}}}{\partial \mu_i} \frac{\partial \mu_i}{\partial \beta_k} \\ &= -(1 + \phi \mu_i)^{-\frac{1}{\phi} - 1} \mu_i X_{ik} \end{aligned} \tag{4.23}$$

Let $t_1 = (1 + \phi \mu_i)$, then,

$$\begin{aligned} \frac{\partial}{\partial \beta_k} \mathcal{L}_i|_{y_i=0} &= \{-\mu_i (1 - \pi_i) X_{ik}\} \frac{t_1^{-\frac{1}{\phi} - 1}}{\pi_i + (1 - \pi_i) t_1^{-\frac{1}{\phi}}} \\ \frac{\partial^2}{\partial \beta_k^2} \mathcal{L}_i|_{y_i=0} &= \{-\mu_i (1 - \pi_i) X_{ik}\} \frac{\partial}{\partial \beta_k} \frac{t_1^{-\frac{1}{\phi} - 1}}{\pi_i + (1 - \pi_i) t_1^{-\frac{1}{\phi}}} + \left\{ -\frac{\partial}{\partial \beta_k} \mu_i (1 - \pi_i) X_{ik} \right\} \frac{t_1^{-\frac{1}{\phi} - 1}}{\pi_i + (1 - \pi_i) t_1^{-\frac{1}{\phi}}} \end{aligned} \tag{4.24}$$

Looking at each term of the expression separately:

$$\begin{aligned}
& \{-\mu_i(1-\pi_i)X_{ik}\}\,\frac{\partial}{\partial\beta_k}\frac{t_1^{-\frac{1}{\phi}-1}}{\pi_i+(1-\pi_i)t_1^{-\frac{1}{\phi}}} \\
= \; & \{-\mu_i(1-\pi_i)X_{ik}\}\,\frac{\left[\frac{\partial}{\partial\beta_k}t_1^{-\frac{1}{\phi}-1}\right]\left[\pi_i+(1-\pi_i)t_1^{-\frac{1}{\phi}}\right]-\left[\frac{\partial}{\partial\beta_k}\left(\pi_i+(1-\pi_i)t_1^{-\frac{1}{\phi}}\right)\right]\left[t_1^{-\frac{1}{\phi}-1}\right]}{\left[\pi_i+(1-\pi_i)t_1^{-\frac{1}{\phi}}\right]^2} \\
= \; & \mu_i^2X_{ik}^2(1-\pi_i)\frac{(\phi+1)t_1^{-\frac{1}{\phi}-2}\left(\pi_i+(1-\pi_i)t_1^{-\frac{1}{\phi}}\right)-(1-\pi_i)t_1^{-2\left(\frac{1}{\phi}+1\right)}}{\left[\pi_i+(1-\pi_i)t_1^{-\frac{1}{\phi}}\right]^2} \\
= \; & \mu_i^2X_{ik}^2(1-\pi_i)\frac{(\phi+1)t_1^{-\frac{1}{\phi}-2}\pi_i+\phi t_1^{-2\left(\frac{1}{\phi}+1\right)}(1-\pi_i)}{\left[\pi_i+(1-\pi_i)t_1^{-\frac{1}{\phi}}\right]^2}
\end{aligned} \tag{4.25}$$

$$\left\{\frac{\partial}{\partial\beta_k}\mu_i(1-\pi_i)X_{ik}\right\}\frac{t_1^{-\frac{1}{\phi}-1}}{\pi_i+(1-\pi_i)t_1^{-\frac{1}{\phi}}} \;=\; \frac{t_1^{-\frac{1}{\phi}-1}}{\pi_i+(1-\pi_i)t_1^{-\frac{1}{\phi}}}\left(\mu_iX_{ik}^2(1-\pi_i)\right) \tag{4.26}$$

Let $\lambda_i = \frac{\pi_i}{1-\pi_i}$, then we can simplify to:

$$\begin{aligned}
\tfrac{\partial^2}{\partial\beta_k^2}\mathcal{L}_i\big|_{y_i=0} &= \mu_i^2X_{ik}^2(1-\pi_i)\frac{(\phi+1)t_1^{-\frac{1}{\phi}-2}\pi_i+\phi t_1^{-2\left(\frac{1}{\phi}+1\right)}(1-\pi_i)}{\left[\pi_i+(1-\pi_i)t_1^{-\frac{1}{\phi}}\right]^2}-\frac{t_1^{-\frac{1}{\phi}-1}}{\pi_i+(1-\pi_i)t_1^{-\frac{1}{\phi}}}\left(\mu_iX_{ik}^2(1-\pi_i)\right) \\
&= \mu_iX_{ik}^2\left\{\frac{\mu_i(\phi+1)t_1^{-\frac{1}{\phi}-2}\lambda_i+\mu_i\phi t_1^{-2\left(\frac{1}{\phi}+1\right)}}{\left[\lambda_i+t_1^{-\frac{1}{\phi}}\right]^2}-\frac{t_1^{-\frac{1}{\phi}-1}}{\lambda_i+t_1^{-\frac{1}{\phi}}}\right\} \\
&= \mu_iX_{ik}^2\left\{\frac{\mu_i(\phi+1)t_1^{-\frac{1}{\phi}}\lambda_i+\mu_i\phi t_1^{-2\left(\frac{1}{\phi}\right)}-t_1^{-\frac{1}{\phi}+1}\left(\lambda_i+t_1^{-\frac{1}{\phi}}\right)}{t_1^2\left[\lambda_i+t_1^{-\frac{1}{\phi}}\right]^2}\right\} \\
&= \mu_iX_{ik}^2\left\{\frac{(\phi+1)t_1^{\frac{1}{\phi}}\lambda_i\mu_i-t_1^{\frac{1}{\phi}+1}\lambda_i-1}{t_1^2\left[\lambda_it_1^{\frac{1}{\phi}}+1\right]^2}\right\} \\
&= \mu_iX_{ik}^2\left\{\frac{t_1^{\frac{1}{\phi}}\lambda_i(\phi\mu_i+\mu_i-t_1)-1}{t_1^2\left[\lambda_it_1^{\frac{1}{\phi}}+1\right]^2}\right\} = \mu_iX_{ik}^2\left\{\frac{t_1^{\frac{1}{\phi}}\lambda_i(\mu_i-1)-1}{t_1^2\left[\lambda_it_1^{\frac{1}{\phi}}+1\right]^2}\right\}
\end{aligned} \tag{4.27}$$

Derivatives for γ

First, note that $\frac{\partial}{\partial \gamma_k}\pi_i = \pi_i(1-\pi_i)X_{ik}$. Then,

$$\begin{aligned}
\mathcal{L}_i|_{y_i=0} &= \log[\pi_i + (1-\pi_i)f_{\text{NB}}(0;\mu_i,\phi)] \\
\frac{\partial}{\partial \gamma_k}\mathcal{L}_i|_{y_i=0} &= \frac{1-f_{\text{NB}}(0;\mu_i,\phi)}{\pi_i+(1-\pi_i)f_{\text{NB}}(0;\mu_i,\phi)}\pi_i(1-\pi_i)X_{ik} \\
\frac{\partial^2}{\partial \gamma_k^2}\mathcal{L}_i|_{y_i=0} &= \left\{\frac{1-f_{\text{NB}}(0;\mu_i,\phi)}{\pi_i+(1-\pi_i)f_{\text{NB}}(0;\mu_i,\phi)}(1-2\pi_i) - \left[\frac{1-f_{\text{NB}}(0;\mu_i,\phi)}{\pi_i+(1-\pi_i)f_{\text{NB}}(0;\mu_i,\phi)}\right]^2 \pi_i(1-\pi_i)\right\} \\
&\quad X_{ik}^2\pi_i(1-\pi_i)
\end{aligned} \tag{4.28}$$

Similarly to above, we can simplify to:

$$\frac{\partial^2}{\partial \gamma_k^2}\mathcal{L}_i|_{y_i=0} = \left\{\frac{\lambda_i t_1^\theta}{\left(\lambda_i t_1^\theta+1\right)^2} - \frac{\lambda_i}{(1+\lambda_i)^2}\right\} X_{ik}^2 \tag{4.29}$$

C.2 When $y_i > 0$

Derivatives for β:

$$\begin{aligned}
\mathcal{L}_i|_{y_i>0} &= \log[(1-\pi_i)f_{\text{NB}}(y_i;\mu_i,\phi)] \\
&= \log\left(\frac{(1-\pi_i)\Gamma(y_i+\frac{1}{\phi})}{y_i!\Gamma\left(\frac{1}{\phi}\right)}\right) + y_i\log(\phi\mu_i) - (y_i+\tfrac{1}{\phi})\log(1+\phi\mu_i) \\
\frac{\partial}{\partial \beta_k}\mathcal{L}_i|_{y_i>0} &= \left[\frac{y_i}{\mu_i} - \frac{(y_i+\frac{1}{\phi})\phi}{1+\phi\mu_i}\right]\mu_i X_{ik} = y_i X_{ik} - (y_i\phi+1)X_{ik}\frac{\mu_i}{1+\phi\mu_i} \\
\frac{\partial^2}{\partial \beta_k^2}\mathcal{L}_i|_{y_i>0} &= -(y_i\phi+1)X_{ik}\left\{\frac{(1+\phi\mu_i)-\mu_i\phi}{(1+\phi\mu_i)^2}\right\}\mu_i X_{ik} \\
&= -\frac{\mu_i(y_i\phi+1)}{(1+\phi\mu_i)^2}X_{ik}^2
\end{aligned} \tag{4.30}$$

Derivatives for γ:

$$\begin{aligned}
\mathcal{L}_i|_{y_i>0} &= \log(1-\pi_i) + \log\left(\frac{\Gamma(y_i+\frac{1}{\phi})(\phi\mu_i)^{y_i}}{y_i!\Gamma\left(\frac{1}{\phi}\right)(1+\phi\mu_i)^{\left(y_i+\frac{1}{\phi}\right)}}\right) \\
\frac{\partial}{\partial \gamma_k}\mathcal{L}_i|_{y_i>0} &= -(1-\pi_i)^{-1}\left[X_{ik}\pi_i(1-\pi_i)\right] = -\pi_i X_{ik} \\
\frac{\partial^2}{\partial \gamma_k^2}\mathcal{L}_i|_{y_i>0} &= -X_{ik}\left[X_{ik}\pi_i(1-\pi_i)\right] = -\pi_i(1-\pi_i)X_{ik}^2
\end{aligned} \tag{4.31}$$

Altogether, the diagonal values of the Hessian can be expressed as:

$$\begin{aligned}
\frac{\partial^2}{\partial \beta_k^2}\mathcal{L}_i &= \sum_{i=1}^{n} \mathrm{I}(y_i = 0) \frac{X_{ik}^2 \mu_i \left[(\mu_i - 1) \exp(X_i\gamma)(1+\phi\mu_i)^{\frac{1}{\phi}} - 1 \right]}{(1+\phi\mu_i)^2 [\exp(X_i\gamma)(1+\phi\mu_i)^{\frac{1}{\phi}} + 1]} \\
&\quad - \sum_{i=1}^{n} \mathrm{I}(y_i > 0) \frac{X_{ik}^2 \mu_i (1+\phi y_i)}{(1+\phi\mu_i)^2} \\
\frac{\partial^2}{\partial \gamma_k^2}\mathcal{L}_i &= \sum_{i=1}^{n} \mathrm{I}(y_i = 0) \frac{X_{ik}^2 \exp(X_i\gamma)(1+\phi\mu_i)^{\frac{1}{\phi}}}{\left[\exp(X_i\gamma)(1+\phi\mu_i)^{\frac{1}{\phi}} + 1 \right]^2} - \sum_{i=1}^{n} \frac{X_{ik}^2 \exp(X_i\gamma)}{(1+\exp(X_i\gamma))^2}
\end{aligned}$$

APPENDIX D: DERIVATION FOR SCZINB APPROXIMATION

The log likelihood for a negative binomial model is

$$l(\mathbf{y},\beta,\phi) = \sum_{i=1}^{n}\left[\log\left(\frac{\Gamma(y_i+1/\phi)}{y_i!\Gamma(1/\phi)}\right)+y_i\log\left(\frac{\phi\mu_i}{1+\phi\mu_i}\right)-\frac{1}{\phi}\log(1+\phi\mu_i)\right] \quad (4.32)$$

Consider the generic form of a GLM model

$$l(\mathbf{y},\beta,\varphi) = \sum_{i=1}^{n} l_i = \sum_{i=1}^{n}\left\{\varphi^{-1}[y_i\theta_i - b(\theta_i)] + c(y_i,\varphi)\right\}$$

Assuming the over-dispersion parameter ϕ is fixed, then a negative binomial distribution belongs to the exponential family. Thus matching it with the generic form of a GLM model, we have

$$\begin{aligned}
\varphi &= 1 \\
\theta_i &= \log\left(\frac{\phi\mu_i}{1+\phi\mu_i}\right), \quad \frac{\partial\theta_i}{\partial\mu_i} = \frac{1}{V(\mu_i)} = \frac{1}{\mu_i+\phi\mu_i^2} \\
b(\theta_i) &= \frac{1}{\phi}\log(1+\phi\mu_i) = -\frac{1}{\phi}\log[1-\exp(\theta_i)], \quad b'(\theta_i)=\mu_i, \quad b''(\theta_i)=V_i \\
c(y_i,\varphi) &= \frac{\Gamma(y_i+1/\phi)}{y_i!\Gamma(1/\phi)}
\end{aligned}$$

Let $\eta_i = x_i^T\beta = g(\mu_i)$, where g is a link function. In our model, $\eta_i = g(\mu_i) = \log(\mu_i)$.

To derive the the MLE of β_j, we start with the score function and Fisher's information matrix. The score function is

$$S_j = \frac{\partial l(\mathbf{y},\beta,\varphi)}{\partial\beta_j} = \sum_{i=1}^{n}\frac{\partial l_i}{\partial\theta_i}\frac{\partial\theta_i}{\partial\mu_i}\frac{\partial\mu_i}{\partial\eta_i}\frac{\partial\eta_i}{\partial\beta_j} = \sum_{i=1}^{n}(y_i-\mu_i)\frac{1}{V(\mu_i)}\frac{1}{g'(\mu_i)}x_{ij}. \quad (4.33)$$

Let I_{jk} be the (j,k)-th element of the Fisher's information matrix,

$$I_{jk} = \sum_{i=1}^{n} E\left[\frac{\partial l_i}{\beta_j}\frac{\partial l_i}{\beta_k}\right] = \sum_{i=1}^{n} E\left\{\frac{(y_i-\mu_i)^2}{\left[V(\mu_i)g'(\mu_i)\right]^2}x_{ij}x_{ik}\right\} = \sum_{i=1}^{n}\left\{\frac{1}{V(\mu_i)\left[g'(\mu_i)\right]^2}x_{ij}x_{ik}\right\},$$

since $E[(y_i-\mu_i)^2] = V(\mu_i)$.

Let $I^{(t-1)} = I(\beta)|_{\beta=\beta^{(t-1)}}$ and $S^{(t-1)} = \partial l/\partial\beta|_{\beta=\beta^{(t-1)}}$. By Fisher scoring, the update of β from the $(t-1)$-th iteration to the t-th iteration is

$$\beta^{(t)} = \beta^{(t-1)} + \left[I^{(t-1)}\right]^{-1} S^{(t-1)} \Rightarrow I^{(t-1)}\beta^{(t)} = I^{(t-1)}\beta^{(t-1)} + S^{(t-1)}.$$

Let W be a diagonal $n\times n$ matrix, with the i-th diagonal element $w_i = 1/\{V(\mu_i)\left[g'(\mu_i)\right]^2\}$ for $i = 1, ..., n$. Then based on equations (4.33) and (4.34), the score function and information matrix can be written as

$$S = XW\zeta \quad \text{and} \quad I = X^TWX,$$

where ζ is a vector of length n and $\zeta_i = (y_i-\mu_i)g'(\mu_i)$. When W is evaluated based on $\beta^{(t-1)}$, we write it as $W^{(t-1)}$. Then the Fisher scoring equation can be written as

$$[X^TW^{(t-1)}X]\beta^{(t)} = X^TW^{(t-1)}X\beta^{(t-1)} + XW^{(t-1)}\zeta = X^TW^{(t-1)}\left[\eta^{(t-1)}+\zeta\right].$$

Therefore, $\beta^{(t)}$ is the solution of weighted least squares with working response being $\xi = \eta + \zeta$, and $\xi_i = x_i\beta + (y_i-\mu_i)g'(\mu_i)$, and weight for the i-th observation is $1/\{V(\mu_i)\left[g'(\mu_i)\right]^2\}$. Here we use log link function $g(\mu) = \log(\mu)$, and thus

$$\xi_i = x_i\beta + (y_i-\mu_i)/\mu_i, \quad \text{and } w_i = \frac{\mu_i^2}{\mu_i+\mu_i^2\phi} = \frac{\mu_i}{1+\mu_i\phi}.$$

APPENDIX E: PSEUDOCODE FOR SCZINB

Given response vector y, covariate matrix X and tuning parameters (λ, τ). (Note that first column of covariate matrix is **1** for the intercept.):

(1) Scale X columns to standard deviation 1.

(2) Initialize:

- $\beta = \gamma = 0$
- $\theta = 1$
- $\Delta_\theta = 1.8e + 308$

(3) While Loop over θ Estimates. (End Criteria: $\Delta_\theta < 1e - 2$ or 5 iterations):

(a) Store Previous Value:

- $\theta_s = \theta$

(b) Initialize:

- $\ell = -1.8e + 308$
- $\Delta = \Delta_k = 1.8e + 308$

(c) While Loop over EM Algorithm (β, γ). (End Criteria: $\Delta < 1e - 5$ or 1000 iterations):

i. Store Previous Values:

- $\ell_s = \ell$

ii. Calculate current estimated means:

- $\pi = \frac{\exp(\eta_{(\gamma)})}{1+\exp(\eta_{(\gamma)})}; \eta_{(\gamma)} = X\gamma$
- $\mu = \exp(\eta_{(\beta)}); \eta_{(\beta)} = X\beta$

iii. Update expected value of latent variable: $\hat{z} = I(y = 0)\frac{\pi}{\pi+(1-\pi)f_{\text{NB}}(0;\mu,\theta)}$

iv. For Loop over each gene (Update β_k, γ_k, $k \in \{1, 2, ..., p\}$):

- While Loop over β_k, γ_k estimates. (End Criteria: $\Delta_k < 1e-5$)

  A. Store old estimates:

  - $\beta_s = \beta_k$
  - $\gamma_s = \gamma_k$

  B. Recalculate current estimated means:

  - $\pi = \frac{\exp(\eta_{(\gamma)})}{1+\exp(\eta_{(\gamma)})}; \eta_{(\gamma)} = X\gamma$
  - $\mu = \exp(\eta_{(\beta)}); \eta_{(\beta)} = X\beta$

  C. Update Beta

  - Calculate IRLS Values:
    - $v_i^{(\beta)} = (1 - \hat{z}_i)\frac{\mu_i^2}{\mu_i + \mu_i^2\phi}$
    - $\xi_i^{(\beta)} = X_i\beta + \frac{y_i - \mu_i}{\mu_i}$
  - Calculate update values:
    - $\bar{b}_k = \frac{\sum_i v_i^{(\beta)}(\xi_i^{(\beta)} - X_{i,-k}\beta_{-k})X_{ik}}{\sum_i v_i^{(\beta)} X_{ik}^2}$
    - $\delta^{(\beta)} = I(k > 1)\frac{\lambda}{2\sum_i v_i^{(\beta)} X_{ik}^2(|\beta_k^{(t)}| + |\gamma_k| + \tau)}$
  - Update:

$$\beta_k = \begin{cases} \bar{b}_k - \delta^{(\beta)}, & \text{if } \bar{b}_k > \delta^{(\beta)} \\ \bar{b}_k + \delta^{(\beta)}, & \text{if } \bar{b}_k < -\delta^{(\beta)} \\ 0 & \text{otherwise} \end{cases} \tag{4.34}$$

  D. Update Gamma

  - Calculate IRLS Values:
    - $v_i^{(\gamma)} = \pi_i(1 - \pi_i)$
    - $\xi_i^{(\gamma)} = X_i\gamma + \frac{\hat{z}_i - \pi_i}{v_i^{(\gamma)}}$
  - Calculate update values:

- $\bar{g}_k = \frac{\sum_i v_i^{(\gamma)}(\xi_i^{(\gamma)} - X_{i,-k}\beta_{-k})X_{ik}}{\sum_i v_i^{(\gamma)} X_{ik}^2}$
- $\delta^{(\gamma)} = I(k > 1)\frac{\lambda}{2\sum_i v_i^{(\gamma)} X_{ik}^2(|\beta_k| + |\gamma_k^{(t)}| + \tau)}$

- Update:

$$\gamma_k = \begin{cases} \bar{g}_k - \delta^{(\gamma)}, & \text{if } \bar{g}_k > \delta^{(\gamma)} \\ \bar{g}_k + \delta^{(\gamma)}, & \text{if } \bar{g}_k < -\delta^{(\gamma)} \\ 0 & \text{otherwise} \end{cases} \tag{4.35}$$

E. Calculate end criteria: $\Delta_k = |\beta_k - \beta_s| + |\gamma_k - \gamma_s|$.

v. Recalculate log likelihood:

A. Recalculate current estimated means:

- $\pi = \frac{\exp(\eta_{(\gamma)})}{1+\exp(\eta_{(\gamma)})}; \eta_{(\gamma)} = X\gamma$
- $\mu = \exp(\eta_{(\beta)}); \eta_{(\beta)} = X\beta$

B. Calculate log likelihood components:

- $\ell_0 = \sum_i \{\log[\pi_i I(y_i = 0) + (1 - \pi_i) f_{\text{NB}}(y_i; \mu_i, \phi)]\}$
- $\rho = \sum_2^p \lambda \log(|\beta_k| + |\gamma_k| + \tau)$

C. $\ell = \ell_0 - \rho$

vi. Calculate end criteria: $\Delta = |\ell_s - \ell|$

(d) Restimate θ using MLE with $y, \mu, \hat{z}$.

(e) Calculate end criteria: $\Delta_\theta = |\theta_s - \theta|$

APPENDIX F: ADDITIONAL FIGURES AND TABLES

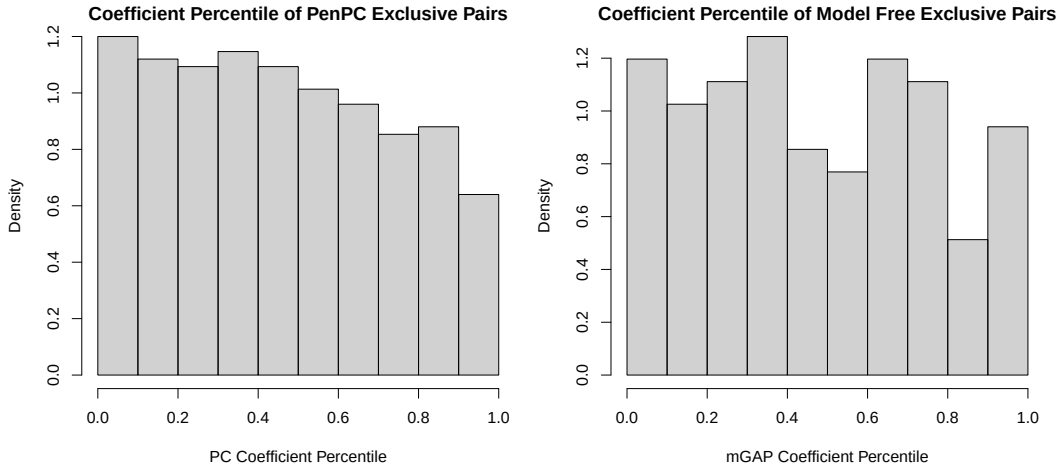
Table F1: Mean Results for Mardia's Test of Multivariate Normality for Residuals of Exclusive Pairs within Cross Validation Sample

	Percentile	Skew Stat	Skew P	Kurt. Stat	Kurt. P	Max Mah. Dis
PenPC	25%	0.196	8.381e-03	10.086	1.098e-07	23.744
	50%	0.428	5.077e-06	11.748	0.000	31.566
	75%	0.826	8.455e-12	13.398	0.000	43.017
Model	25%	0.258	1.434e-03	10.732	7.859e-11	25.599
Free	50%	0.481	1.089e-06	12.252	0.000	39.880
	75%	0.957	6.102e-13	14.834	0.000	46.736

Max Mah. Dis refers to maximum Mahalanobis Distance.
Kurt. refers to kurtosis.

Figure F1: Comparison of coefficient sizes for pairs exclusively selected by either PenPC or Model Free. For each pair, two coefficients per member are estimated (one for each member used as the response) by either the PC algorithm or mGAP, then the L_2 norm of these coefficients are calculated. Since the coefficients derived from PenPC and Model Free are not directly comparable, we show, in the following barplots, the percentile of the derived L_2 norms for each method separately.

(a) Original Sample



(b) Cross Validation Sample

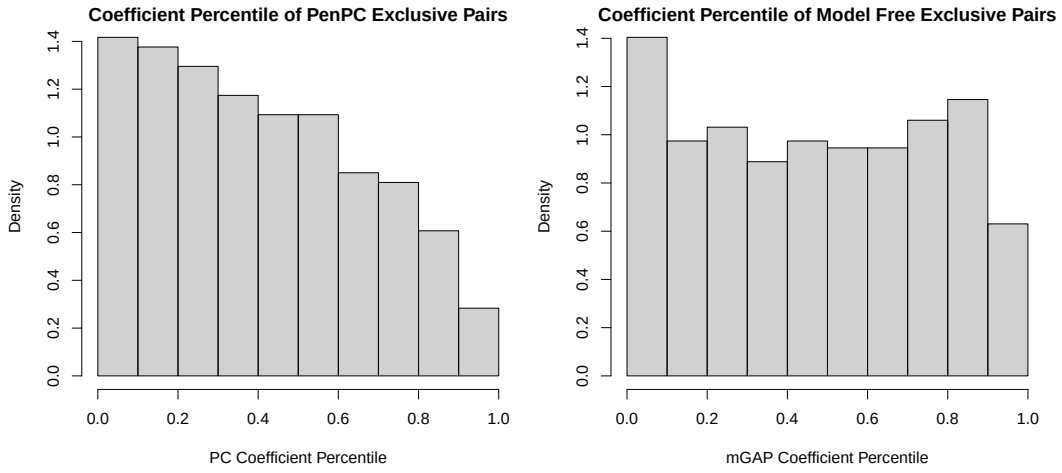
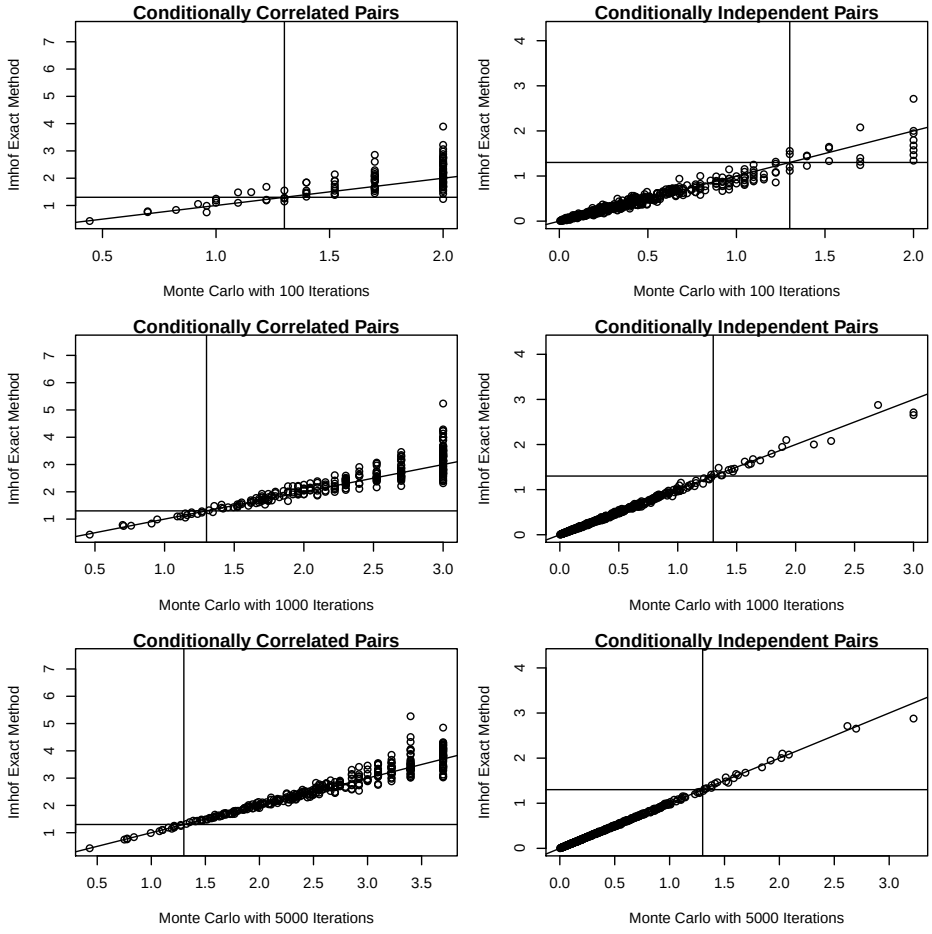


Figure F2: Comparison of $-\log_{10}$(P-Values) Found by Monte Carlo Simulation vs Imhof's Exact Method across 100, 1000, and 5000 iterations. Horizontal and vertical reference lines are for $-\log_{10}(0.05)$ and diagonal reference line is for $y = x$ (p-values are equivalent).



REFERENCES

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, 11:R106.

Barretina, M. J., Edelman, E. J., Heidorn, S. J., Greenman, C. D., Dastur, A., Lau, K. W., Greninger, P., Thompson, I. R., Luo, X., Soares, J., Liu, Q., Iorio, F., Surdez, D., Chen, L., Milano, R. J., Bignell, G. R., Tam, A. T., Davies, H., Stevenson, J. A., Barthorpe, S., Lutz, S. R., and et al. (2012). The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607.

Chen, T.-H., Sun, W., Fine, J. P., et al. (2016a). Designing penalty functions in high dimensional problems: The role of tuning parameters. *Electronic Journal of Statistics*, 10(2):2312–2328.

Chen, Y., Lun, A. T. L., and Smyth, G. K. (2016b). From reads to genes to pathways: differential expression analysis of RNA-Seq experiments using Rsubread and the edgeR quasi-likelihood pipeline. *F1000Research*, 5:1438.

Chickering, D. M. (2003). Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554.

Colombo, D. and Maathuis, M. H. (2014). Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research*, 15(1):3741–3782.

Cook, R. D. (2000). Save: a method for dimension reduction and graphics in regression. *Communications in Statistics - Theory and Methods*, 29(9-10):2109–2121.

Daudin, J. J. (1980). Partial association measures and an application to qualitative regression. *Biometrika*, 67(3):581–590.

De Campos, C. P. and Ji, Q. (2011). Efficient structure learning of Bayesian networks using constraints. *The Journal of Machine Learning Research*, 12:663–689.

Drton, M. and Perlman, M. D. (2004). Model selection for Gaussian concentration graphs. *Biometrika*, 91(3):591–602.

Drton, M. and Perlman, M. D. (2007). Multiple Testing and Error Control in Gaussian Graphical Model Selection. *Statistical Science*, 22(3):430–449.

Eberwine, J., Sul, J.-Y., Bartfai, T., and Kim, J. (2014). The promise of single-cell sequencing. *Nature methods*, 11(1):25–27.

ErdËİos, P. and RÂťenyi, A. (1959). On random graphs. *Publ. Math. Debrecen*, 6:290–297.

Foygel, R. and Drton, M. (2010). Extended bayesian information criteria for gaussian graphical models. In *Advances in Neural Information Processing Systems*, pages 604–612.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Fukumizu, K., Bach, F. R., and Jordan, M. I. (2004). Kernel dimensionality reduction for supervised learning. *Advances in Neural Information Processing Systems*, 16:81.

Fukumizu, K., Gretton, A., Sun, X., and SchÃűlkopf, B. (2007). Kernel Measures of Conditional Dependence. In *NIPS*, volume 20, pages 489–496.

Garnett, J., Caponigro, G., Stransky, N., Venkatesan, K., Margolin, A. A., Kim, S., Wilson, C. J., LehÃąr, J., Kryukov, G. V., Sonkin, D., Reddy, A., Liu, M., Murray, L., Berger, M. F., Monahan, J. E., Morais, P., Meltzer, J., Korejwa, A., JanÃľ-Valbuena, J., Mapa, F. A., Thibault, J., and et al. (2012). Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature*, 483(7391):570–575.

Ha, M. J., Sun, W., and Xie, J. (2016a). PenPC: A Two-step Approach to Estimate the Skeletons of High Dimensional Directed Acyclic Graphs. *Biometrics*, 72(1):146–155.

Ha, M. J., Sun, W., and Xie, J. (2016b). PenPC: A two-step approach to estimate the skeletons of high-dimensional directed acyclic graphs. *Biometrics*, 72(1):146–155.

Hardcastle, T. J. and Kelly, K. A. (2010). baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data. *BMC Bioinformatics*, 11(1):422.

Harris, N. and Drton, M. (2013). PC Algorithm for Nonparanormal Graphical Models. *J. Mach. Learn. Res.*, 14(1):3365–3383.

Hoeffding, W. (1948). A Non-Parametric Test of Independence. *The Annals of Mathematical Statistics*, 19(4):546–557.

Huang, T.-M. (2010). Testing conditional independence using maximal nonlinear conditional correlation. *The Annals of Statistics*, 38(4):2047–2091.

Imhof, J. P. (1961). Computing the Distribution of Quadratic Forms in Normal Variables. *Biometrika*, 48(3/4):419.

Jansakul, N. and Hinde, J. P. (2008). Score Tests for Extra-Zero Models in Zero-Inflated Negative Binomial Models. *Communications in Statistics - Simulation and Computation*, 38(1):92–108.

Kalisch, M. and BÃijhlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *The Journal of Machine Learning Research*, 8:613–636.

Klein, A. M., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., Peshkin, L., Weitz, D. A., and Kirschner, M. W. (2015). Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201.

Kolodziejczyk, A., Kim, J. K., Svensson, V., Marioni, J., and Teichmann, S. (2015). The Technology and Biology of Single-Cell RNA Sequencing. *Molecular Cell*, 58(4):610–620.

Lam, C. and Fan, J. (2009). Sparsistency and rates of convergence in large covariance matrix estimation. *The Annals of Statistics*, 37(6B):4254–4278.

Lemeire, J., Meganck, S., and Cartella, F. (2010). Robust independence-based causal structure learning in absence of adjacency faithfulness. *on Probabilistic Graphical Models*, page 169.

Li, K.-C. (1991). Sliced Inverse Regression for Dimension Reduction. *Journal of the American Statistical Association*, 86(414):316–327.

Liu, H., Han, F., Yuan, M., Lafferty, J., and Wasserman, L. (2012). High-dimensional semiparametric Gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326.

Liu, H., Lafferty, J., and Wasserman, L. (2009). The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *The Journal of Machine Learning Research*, 10:2295–2328.

Loh, P.-L. and Wainwright, M. J. (2013). Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses. *The Annals of Statistics*, 41(6):3022–3049.

Love, M. I., Anders, S., Kim, V., and Huber, W. (2015). RNA-Seq workflow: gene-level exploratory analysis and differential expression. *F1000Research*, 4:1070.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Maathuis, M. H., Kalisch, M., and BÃijhlmann, P. (2009). Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133–3164.

MacQueen, J. B. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability: Statistics*, pages 281–288. University of California Press. Google-Books-ID: IC4Ku_7dBFUC.

Mahata, B., Zhang, X., Kolodziejczyk, A., Proserpio, V., Haim-Vilmovsky, L., Taylor, A., Hebenstreit, D., Dingler, F., Moignard, V., GÃűttgens, B., Arlt, W., McKenzie,

A. J., and Teichmann, S. (2014). Single-Cell RNA Sequencing Reveals T Helper Cells Synthesizing Steroids De Novo to Contribute to Immune Homeostasis. *Cell Reports*, 7(4):1130–1142.

Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57(3):519–530.

McDavid, A., Gottardo, R., Simon, N., and Drton, M. (2016). Graphical models for zero-inflated single cell gene expression. *arXiv preprint arXiv:1610.05857*.

Meinshausen, N. and BÃijhlmann, P. (2006). High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436–1462.

Padovan-Merhar, O. and Raj, A. (2013). Using variability in gene expression as a tool for studying gene regulation. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 5(6):751–759.

Patel, A. P., Tirosh, I., Trombetta, J. J., Shalek, A. K., and Gillespie, S. M. (2014). Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, 344(6190):1396–1401.

Pearl, J. (2009). *Causality*. Cambridge University Press.

PeâĂŹer, D., Regev, A., Elidan, G., and Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(suppl 1):S215–S224.

Picelli, S., Faridani, O. R., BjÃűrklund, A. K., Winberg, G., Sagasser, S., and Sandberg, R. (2014). Full-length RNA-seq from single cells using Smart-seq2. *Nature Protocols*, 9(1):171–181.

Pollen, A. A., Nowakowski, T. J., Shuga, J., Wang, X., Leyrat, A. A., Lui, J. H., Li, N., Szpankowski, L., Fowler, B., Chen, P., Ramalingam, N., Sun, G., Thu, M., Norris, M., Lebofsky, R., Toppani, D., Kemp, D. W., Wong, M., Clerkson, B., Jones, B. N., Wu, S., Knutsson, L., Alvarado, B., Wang, J., Weaver, L. S., May, A. P., Jones, R. C., Unger, M. A., Kriegstein, A. R., and West, J. A. A. (2014). Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature Biotechnology*, 32(10):1053–1058.

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140.

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

Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. (2013). Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263–2291.

Shawe-Taylor, J., Williams, C., Cristianini, N., and Kandola, J. (2005). On the eigenspectrum of the Gram matrix and the generalization error of kernel-PCA. *IEEE Transactions on Information Theory*.

Song, K. et al. (2009). Testing conditional independence via rosenblatt transforms. *The Annals of Statistics*, 37(6B):4011–4045.

Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, prediction, and search*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2nd ed edition.

Stegle, O., Teichmann, S. A., and Marioni, J. C. (2015). Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, 16(3):133–145.

Su, L. and White, H. (2007). A consistent characteristic function-based test for conditional independence. *Journal of Econometrics*, 141(2):807–834.

Sun, W. and Li, L. (2012). Multiple Loci Mapping via Model-free Variable Selection. *Biometrics*, 68(1):12–22.

SzÃľkely, G. J. and Rizzo, M. L. (2012). On the uniqueness of distance covariance. *Statistics & Probability Letters*, 82(12):2278–2282.

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., Lao, K., and Surani, M. A. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods*, 6(5):377–382.

Toh, B., Toh, B., Abastado, J.-P., and Abastado, J.-P. (2012). Myeloid cells. *Oncoimmunology*, 1(8):1360–1367.

van den Broek, J. (1995). A Score Test for Zero Inflation in a Poisson Distribution. *Biometrics*, 51(2):738–743.

Wall, M. E., Rechtsteiner, A., and Rocha, L. M. (2003). Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer.

Werhli, A. V. and Husmeier, D. (2007). Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical applications in genetics and molecular biology*, 6(1).

Wilding, G. E. and Mudholkar, G. S. (2008). Empirical approximations for Hoeffding?s test of bivariate independence using two Weibull extensions. *Statistical Methodology*, 5(2):160–170.

Yang, B., Zhang, J., Shang, J., and Li, A. (2011). A bayesian network based algorithm for gene regulatory network reconstruction. In *2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pages 1–4.

Yap, T. A., Gerlinger, M., Futreal, P. A., Pusztai, L., and Swanton, C. (2012). Intratumor Heterogeneity: Seeing the Wood for the Trees. *Science Translational Medicine*, 4(127):127ps10–127ps10.

Yavari, F., Towhidkhah, F., and Gharibzadeh, S. (2008). Gene Regulatory Network Modeling using Bayesian Networks and Cross Correlation. In *2008 Cairo International Biomedical Engineering Conference*, pages 1–4.

Yuan, M. and Lin, Y. (2006). Model Selection and Estimation in Regression with Grouped Variables. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 68(1):49–67.

Zhang, J. and Spirtes, P. (2016). The three faces of faithfulness. *Synthese*, 193(4):1011–1027.

Zhang, K., Peters, J., Janzing, D., and SchÃűlkopf, B. (2012). Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775*.

Zheng, G. X. Y., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., Gregory, M. T., Shuga, J., Montesclaros, L., Underwood, J. G., Masquelier, D. A., Nishimura, S. Y., Schnall-Levin, M., Wyatt, P. W., Hindson, C. M., Bharadwaj, R., Wong, A., Ness, K. D., Beppu, L. W., Deeg, H. J., McFarland, C., Loeb, K. R., Valente, W. J., Ericson, N. G., Stevens, E. A., Radich, J. P., Mikkelsen, T. S., Hindson, B. J., and Bielas, J. H. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8:14049.