Bayesian Networks' Reliability and Multiway Networks for Gaussian and Non-Gaussian Distributed Data

Sijia Li

School of Mathematics

University of Leeds

A thesis submitted for the degree of

Doctor of Philosophy

7th October 2022

To Tilia and Aeneas, for the courage of living in the strange times.

Joint publications

Most of the work in Chapter 3 has been published, as follows:

- Li, S., López-García, M. and Cutillo, L., 2021, December. Simulation-based Evaluation of the Reliability of Bayesian Hierarchical Models for sc-RNAseq Data. In 2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS) (pp. 345-352). IEEE.
- Code availability for Chapter 3: https://github.com/lilythepooh/BASiCS-Reliability.git

Most of work in Chapter 4 has been published, as follows:

- Li, S., López-García, M., Lawrence, N.D. and Cutillo, L., 2022, May. Two-way Sparse Network Inference for Count Data. In International Conference on Artificial Intelligence and Statistics (pp. 10924-10938). PMLR.
- Code availability for Chapter 4: https://github.com/luisacutillo78/Scalable_Bigraphical_Lasso.git

Acknowledgements

First and foremost, I would like to thank my supervisors, Dr Luisa Cutillo and Dr Martín López-García, for their patient guidance and kind help throughout this journey. I am very grateful for all their advice and support, both academically and personally.

I would like to thank my supervisor in the first year, Dr John Paul Gosling, for giving me this opportunity to start my PhD journey. I am very grateful for his guidance and help in my first year. I wish to thank our collaborator, Dr Neil Lawrence, for his contribution to the Scalable Bigraphical Lasso project. I would like to thank Dr Mike Croucher for his support in optimizing the MATLAB code. I would also like to thank Dr Stephen Griffiths, Dr Jan Palczeski, and the School of Mathematics for their generous support and trust. This research is kindly funded through the Engineering and Physical Science Research Council Doctoral Training Partnership, which has given me much opportunities to develop as a better researcher and a better person.

I would like to express my gratitude to the School of Mathematics reception team, who have been ever so warm and helpful to me since my first day here, especially Ms Helen Copeland, Ms Emma Rimmington, Ms Tilly Hindle and Ms Kim Darwood. Special thanks to Ms Helen Copeland. She is my first friend in Leeds. New friendships are another great gain of my PhD journey. I thank my academic sister, Anastasia Frantsuzova, for her guidance in both PGR life and British life. I thank my social bubble during COVID, Merin Joseph, for taking me out to walks and keeping me sane during the lockdown. I thank Muyang Zhang, Rukia Nuermaimaiti and Wajiha Rehman, for their longstanding moral support. I thank Joseph Elmes, Giacomo Baldo and Jacob Cancino-Romero for their kind help.

I thank Aurélie Astoul, Anna Sigalou, Anna Guseva, Girish Nivarti and Jean Peyen for the fun times we had together. I would also like to thank my counseller Ms Penny Mathern through the Student Counselling and Welling Being Service at the university. my sessions with her in late 2020 has been very helpful.

I would like to thank my parents for their never-changing love, care and support. Thanks to my mother, Ms Lianjun Yang, Her strength, kindness and endurance has always inspired me to strive. I salute to my father, who showed great courage before passing away from cancer nine years ago. Thanks to all my other friends, although we can only talk through internet these days. Their support has been a great comfort and the source of joy during this journey.

I thank Suede, the 1975 and Taylor Swift for the company of their music along the way. I thank the many writers whose books have brought me enjoyment and strength throughout the years. I thank the CO-OP supermarket in Leeds University Union for giving me my first paid job and for selling Cappucino for £1. I thank Slingshot Simulations in Nexus and the EPSRC IAA team of the university for the opportunity of a internship and professional development. I also thank Stewart Adams and John Nicholson for inventing Ibuprofen, without which I won't be able to function during my period.

Last but not least, I thank Leeds and its people for their kindness and heartfelt warmth. The time I spent in Leeds has made me who I am, and this lovely Yorkshire city will always be a hometown in my heart.

Abstract

This thesis focuses on two topics in the area of statistical modelling applications.

The first topic concerns the evaluation of the reliability of Bayesian Hierarchical Models for scRNAseq data. Bayesian Hierarchical Models (BHM) are used in various application fields such as biology, social science and engineering for identification of confounding factors, thus enabling the extraction of the information of interest. BHMs are typically formulated by specifying the data model, the parameters model and the prior distributions. The posterior inference of a BHM depends on both the model specification and the computational algorithm used. We use the term "reliability" to indicate a methodology's ability to recover the "ground truth" or the underlying distribution embedded in the data. Testing the reliability of a BHM is an open question. The most straightforward way to test the reliability of a BHM inference is to compare the posterior distributions with the ground truth value of the model parameters, when available. However, when dealing with experimental data, the true value of the underlying parameters is typically unknown. In these situations, numerical experiments based on synthetic datasets generated from the model itself offer a natural approach to check model performance and posterior estimates. In this thesis, we show how to test the reliability of a BHM. We introduce a change in the model assumptions to allow for prior contamination, and develop a simulation-based evaluation framework to assess the reliability of the inference of a given BHM. We illustrate our approach on a specific BHM used for Bayesian analysis of scRNAseq Data (BASiCS).

The second topic considers the problem of efficient multi-way network inference for non-Gaussian data. Classically, statistical datasets have more data points than features ($n > p$). The standard model of classical statistics caters for the case where data points are considered conditionally independent given the parameters. However, for $n \approx p$ or $p > n$ data such models are poorly determined. Kalaitzis *et al.* (2013) introduced the Bigraphical Lasso, a method for two-way network inference in both samples and features. Greenewald *et al.* (2019) introduced an algorithm for the inference of the multi-way version. Both methods estimate sparse precision matrices based on the Cartesian product of Gaussian Markov random field graphs. However, the theoretical foundation of such models has some gaps in the previous literature, to the best of my knowledge. In this thesis we formally give and prove a theorem as the theoretical foundation of multi-way graphical models. Moreover, the original Bigraphical Lasso algorithm is not applicable in case of large p and n due to memory requirements. In this thesis we present Scalable Bigraphical Lasso, a novel version of the algorithm which exploits eigenvalue decomposition of the Cartesian product graph, and matrix algebra, to reduce the memory requirements from $O(n^2 p^2)$ to $O(n^2 + p^2)$, and to improve the computational efficiency. We also present the Scalable K-graphical Lasso method for multi-way network inference, leveraging eigenvalue decomposition to simultaneously infer hidden structures in tensor-valued data. Finally, many datasets in different application fields, such as biology, medicine and social science, come as non-Gaussian data, for which Gaussian based models such as the original Bigraphical model and its multi-way version are not applicable. Thus, we extend our multi-way network inference approach so that it can be used for non-Gaussian data. In summary, our methodology accounts for the dependencies across different directions in datasets, reduces the computational complexity for high dimensional data and enables us to deal with both discrete and continuous data. Numerical studies on both synthetic and real datasets are presented to showcase the performance of our methods.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Many statistical models have been proposed to discover new knowledge and to infer hidden structures from real world data. In this thesis we focus on two applied statistical problems.

Real world problems are ubiquitously affected by various hidden factors. We focus on an example in biology: The recent rapid evolution of high-throughput sequencing technologies has enabled the quantification of gene expression at the single-cell level. These data are called *single-cell RNA sequencing* (scRNAseq) data. The gene expression data recorded by scRNAseq are affected by several hidden factors such as technical noise, cell size and biological variation. When dealing with this type of data people have used *Bayesian Hierarchical Models* (BHMs). BHMs could allow the identification of confounding factors, thus enabling the extraction of the information of interest. A great advantage of BHMs for scRNAseq is that their inference often borrows information from across genes and cells. Therefore, they draw a more comprehensive biological picture by putting the intercellular dynamics into consideration. An example of BHM models for scRNAseq data is the BASiCS framework developed in Vallejos *et al.* (2015, 2016) and Eling *et al.* (2018). This framework allows them to infer relevant parameters from observed data, and the method will determine some particular characteristics through parameters representing factors which affect gene expression data. However, due to the complexity of Bayesian Hierarchical Models and the high-dimensional nature of biological data, these models

could be computationally expensive and time-consuming. The high cost makes the validation of these methods even more important and urgent. However, validation for this methodology in the sense of comparing the ground truth with the inference results was missing. This is because the ground truth of these hidden factors inferred from BHMs is often unknown in biology. In Chapter 3 of this thesis, we address this issue by exploiting a simulation-based framework for evaluation of BHMs.

Another problem is the learning of hidden structures from real world data. These structures can appear in more than one dimensions. For example, videos contain structures in pixels and in frames (Greenewald, 2017); Traffic data concerns relationship of various factors such as space, time and direction (Ahn *et al.*, 2022); Gene expression data encodes conditional dependencies across cells (tissues) and across genes (Almet *et al.*, 2021; Cang & Nie, 2020; Svensson *et al.*, 2020). Recently, two-way and multi-way network models based on Gaussian distribution and graphical models are proposed to infer these dependency structures simultaneously. Consider a dataset in the form of a matrix, Kalaitzis *et al.* (2013) propose Bigraphical model, which consists of a Gaussian distribution model with an inverse covariance matrix structured as a *Kronecker sum* (KS) of two matrices. Here the KS structure of the inverse covariance matrix corresponds to the Cartesian product of networks between the rows and between the columns of the data matrix, respectively. The Bigraphical Lasso algorithm is presented in Kalaitzis *et al.* (2013) to solve this problem. However, Bigraphical Lasso has the issue of computational efficiency, especially for high dimensional data. Another limitation of Bigraphical Lasso is that it can only work with Gaussian data, while a lot of real world data are non-Gaussian data, such as count data. We address these issues in Chapter 4, where we propose a novel Scalable Bigraphical Lasso by utilising eigen-decomposition and matrix algebra. We also extend our approach for application on non-Gaussian data by introducing a *Gaussian Copula* approach, where the structure embedded in non-Gaussian data are projected to the relationship between latent Gaussian variables.

The exploration above leads us to a new problem: real world data sometimes come in the form of *tensors* with more than two dimensions. For example, in the case of three dimensions, the data is stored in a cube-like structure where three indices are needed to fix a data point. Greenewald *et al.* (2019) considered a multi-way network model, which consists of a Gaussian distribution model with a inverse covariance

matrix structured as a *Kronecker sum* (KS) of multiple matrices. Here the KS structure of the inverse covariance matrix corresponds to the Cartesian product of networks between each direction of the tensor-valued data. TeraLasso method is proposed by Greenewald *et al.* (2019) to solve this model. However, it can only work on Gaussian data, while in the real world some tensor-valued data are non-Gaussian. In Chapter 5, we propose a novel Scalable K-Graphical Lasso method based on our Scalable Bigraphical Lasso method in Chapter 4, and we extend our method for application on non-Gaussian tensor-valued data by introducing a Gaussian Copula approach.

1.2 Main Contributions

The following points summarise the main contributions of this thesis:

- Exploration of a simulation-based evaluation framework for Bayesian Hierarchical Models, including the study of the effect of a contaminated prior on the posterior results.
- A modified `BASiCS` package, providing the choice of mixed prior on a spectrum.
- Implementation of the Simulation-based Calibration method for the BASiCS framework.
- Proof of the existence of Gaussian Markov Random field for the Cartesian Product of two Gaussian Markov Random field graphs.
- Development of Scalable Bigraphical Lasso, a novel eigen-decomposition based algorithm for two-way network inference with a *Kronecker sum* (KS) structure.
- Comparisons with other two-way (multi-way) network inference methods with a KS structure, namely Bigraphical Lasso (Kalaitzis *et al.*, 2013) and TeraLasso (Greenewald *et al.*, 2019), in terms of network recovery accuracy and running time.
- Extension of the two-way network model to non-Gaussian data with a non-paranormal Gaussian Copula approach.

- Application of nonparanormal Scalable Bigraphical Lasso to image clustering.
- Application of nonparanormal Scalable Bigraphical Lasso to single-cell RNA sequencing data.
- Development of Scalable K-graphical Lasso, an extension of Scalable Bigraphical Lasso, enabling multi-way network inference from K-way tensor-valued data, $K > 2$.
- Comparisons with another multi-way network inference methods with a KS structure, namely TeraLasso (Greenewald *et al.*, 2019), in terms of network recovery accuracy and running time.
- Extension of the multi-way network model to non-Gaussian data with a nonparanormal Gaussian Copula approach.
- Application of nonparanormal Scalable KGraphical Lasso to image clustering.

1.3 Thesis Outline

This thesis is structured as the following:

- In Chapter 2, we introduce the mathematical background on Bayesian Statistics and graphical models, including our proof of the existence of Gaussian Markov Random field for the Cartesian Product of two Gaussian Markov Random field graphs.
- In Chapter 3, we present our work on Simulation-based evaluation of Bayesian Hierarchical Models, with BASiCS (Eling *et al.*, 2018; Vallejos *et al.*, 2015, 2016) as an example. We show the limitation of the BASiCS framework based on evidence we found from our exploration, and we pointed out the potential direction for improvement for future work.
- In Chapter 4, we present our Scalable Bigraphical algorithm for matrix-variate data and its extension to non-Gaussian data with a nonparanormal approach. Our experiment on synthetic Gaussian data shows that, when comparing with

Bigraphical Lasso, our method improves a lot in terms of efficiency while maintaining high accuracy. Our experiments on synthetic count data and real data have shown the applicability of our method on non-Gaussian data.

- In Chapter 5, we present our Scalable K-graphical algorithm for tensor-valued data and its extension to non-Gaussian tensor-valued data with a nonparanormal approach. Our experiment on synthetic Gaussian data shows that, when comparing with TeraLasso, our method has significantly better performance when the size of the tensor is small, while when the size of the tensor is large, the performance of our method is still comparable with TeraLasso. We also show the applicability of our method on non-Gaussian data with a synthetic count data example and a real data example.
- In Chapter 6, we conclude this thesis and discuss future work.

Chapter 2

Preliminaries

This chapter provides some general background of Bayesian Statistics and Graphical Models, focusing on the mathematical and statistical concepts relevant to the research presented in this thesis.

2.1 Bayesian Statistics

In Chapter 3 of this thesis, analysis of a type of model called Bayesian Hierarchical Model is carried out, hence we introduce some relevant basic concepts of Bayesian Statistics in this section.

In Probability Theory, given two events A and B with probabilities $P(A)$ and $P(B)$, we define the conditional probability $P(A|B)$ as a measure of the probability of A occurring given that B has occurred (Pfeiffer, 2013). Bayes (1763) discussed a special case of Bayes' theorem, based on the conditional probability $P(A|B) = \frac{P(B\cap A)}{P(B)}$, with $P(B) \neq 0$. Laplace (1814) introduces the theorem in a more general way, which is called Bayes Theorem:

Theorem 2.1 [Bayes' Theorem] When the set of events $\{A_1, \ldots, A_M\}$ represents all the possible causes for event B, then for any $A_i \in \{A_1, \ldots, A_M\}$ we have

$$\begin{aligned} P(A_i|B) &= \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^{M} P(A_j)P(B|A_j)} \\ &= \frac{P(B|A_i)P(A_i)}{P(B)}. \end{aligned}$$

Here, $P(A_i)$ is our belief on the probability of event A_i before observing B, also called the *prior*. When we observe B, this theorem can be used to update our belief on A_i. As $P(B)$ is usually unknown, it can be treated as an unknown constant. The *posterior* $P(A_i|B)$ can be computed using the *proportional update*:

$$P(A_i|B) \propto P(B|A_i)P(A_i).$$

Consider an observation of data $\boldsymbol{y}$, the parameter ζ_1 governing the data generating process $\pi(\boldsymbol{y}|\zeta_1)$. We denote our prior belief on ζ_1 as $\pi(\zeta_1)$. Applying Bayes' theorem, we have

$$\pi(\zeta_1|\boldsymbol{y}) \propto \pi(\boldsymbol{y}|\zeta_1)\pi(\zeta_1). \tag{2.1}$$

2.1.1 Bayesian Hierarchical Models

Bayesian Hierarchical Models (BHMs) are a type of probabilistic graphical model based on Bayes' Theorem. BHMs are widely used in various application fields such as biology (Dondelinger *et al.*, 2013; Fang *et al.*, 2018), medicine (Lawson, 2018; Scannell *et al.*, 2020), engineering (Babaleye *et al.*, 2019; Mishra *et al.*, 2018) and social science (Costa & Ortale, 2012; Yoshioka *et al.*, 2022). Bayesian Hierarchical models can allow the identification of confounding factors, thus enabling the extraction of the information of interest conditionally on other factors in a complex system.

In Equation (2.1), we can use another unknown parameter ζ_2 to describe our prior belief on the generating process of ζ_1, i.e. we can introduce the prior $\pi(\zeta_1|\zeta_2)$. In this case ζ_2 is called the *hyperparameter*. Similarly, we have a *hyperprior* belief on the generating process of ζ_2, $\pi(\zeta_2)$. In this framework, we can build a Bayesian Hierarchical Model (BHM):

$$\text{Stage 1: } \boldsymbol{y}|\zeta_1,\zeta_2 \sim \pi(\boldsymbol{y}|\zeta_1,\zeta_2)$$

$$\text{Stage 2: } \zeta_1|\zeta_2 \sim \pi(\zeta_1|\zeta_2)$$

$$\text{Stage 3: } \zeta_2 \sim \pi(\zeta_2)$$

Applying the Bayes' theorem, we can update our belief on ζ_1 and ζ_2:

$$\begin{aligned}\pi(\zeta_1,\zeta_2|\boldsymbol{y}) &\propto \pi(\boldsymbol{y}|\zeta_1,\zeta_2)\pi(\zeta_1,\zeta_2)\\ &= \pi(\boldsymbol{y}|\zeta_1,\zeta_2)\pi(\zeta_1|\zeta_2)\pi(\zeta_2).\end{aligned}$$

It is possible that a BHM has more than three stages, more than one parameter and more than one hyperparameters. In such case we can consider a general Bayesian Hierarchical Model with data (observation) vector $\boldsymbol{y}$ and parameters $\{\boldsymbol{\zeta}^{(1)},\ldots,\boldsymbol{\zeta}^{(M)}\}$:

$$\boldsymbol{y}|\boldsymbol{\zeta}^{(1)} \sim \pi(y|\boldsymbol{\zeta}^{(1)})$$

$$\boldsymbol{\zeta}^{(i)}|\boldsymbol{\zeta}^{(i+1)} \sim \pi(\boldsymbol{\zeta}^{(i)}|\boldsymbol{\zeta}^{(i+1)}),\ i = 1,\ldots,M-1,$$

$$\boldsymbol{\zeta}^{(M)} \sim \pi\left(\boldsymbol{\zeta}^{(M)}\right).$$

Applying Bayes' theorem, we have the posterior distribution:

$$\boldsymbol{\zeta}^{(i)}|\boldsymbol{\zeta}^{(i+1)},\boldsymbol{y} \sim \pi(\boldsymbol{\zeta}^{(i)}|\boldsymbol{\zeta}^{(i+1)},\boldsymbol{y}),\ i = 1,\ldots M-1.$$

2.1.2 Identifiability

In statistical modelling, identifiability analysis aims to find out if the model parameters can be uniquely determined from the distribution of the observed samples (Paulino & de Bragança Pereira, 1994). Consider a probability space $\left(\mathscr{Y},\sigma\left(\mathscr{Y}\right),\mathscr{P}_{\boldsymbol{\zeta}}\right)$, where $\mathscr{Y}$ is the sample space of observed data, $\sigma\left(\mathscr{Y}\right)$ is the event space where an event is a subset of $\mathscr{Y}$, and $\mathscr{P}_{\boldsymbol{\zeta}} = \left\{P_{\boldsymbol{\zeta}} : \boldsymbol{\zeta} \in \mathscr{Z}\right\}$ is the family of parametric distributions in $\sigma\left(\mathscr{Y}\right)$, with $\mathscr{Z}$ being an open subset of $\mathbb{R}^M$. Following Dasgupta *et al.* (2007), Paulino & de Bragança Pereira (1994) and Rothenberg (1971), we give the following definition:

Definition 2.1 The family $\mathscr{P}_{\boldsymbol{\zeta}}$ is said to be (globally) identifiable if

$$P_{\boldsymbol{\zeta}^{(1)}} = P_{\boldsymbol{\zeta}^{(2)}} \Longrightarrow \boldsymbol{\zeta}^{(1)} = \boldsymbol{\zeta}^{(2)}.$$

For example, without any constraints, $P_{(\lambda_1,\lambda_2)} = \text{Poisson}\left(\lambda_1\lambda_2\right)$ is non-identifiable, since $(\lambda_1,\lambda_2) = (1,2)$ and $(\lambda_1,\lambda_2) = (2,1)$ define the same distribution, and so $P_{(1,2)} = P_{(2,1)} \not\Longrightarrow (1,2) = (2,1)$.

2.1.3 Monte Carlo Markov Chain

For a given data-generating model $\pi(\boldsymbol{y}|\boldsymbol{\zeta})$, observation $\boldsymbol{y}$, and parameter vector $\boldsymbol{\zeta}$, the calculation of the posterior density $\pi\left(\boldsymbol{\zeta}|\boldsymbol{y}\right) \propto \pi\left(\boldsymbol{\zeta}\right)\pi(\boldsymbol{y}|\boldsymbol{\zeta})$ is often intractable, hence in practice we often obtain posterior samples from $\pi\left(\boldsymbol{\zeta}|\boldsymbol{y}\right)$ with numerical methods. Markov Chain Monte Carlo (MCMC) methods are a class of algorithms used to infer the posterior distribution numerically. Here we only introduce the Adaptive Metropolis-within-Gibbs Sampling algorithm from Roberts & Rosenthal (2009) as described in Algorithm 1. In short, Adaptive Metropolis refers to drawing candidate values from a *proposal distribution* with adaptive parameters. In Roberts & Rosenthal (2009), the proposal distributions are either Normal distributions or log-Normal distributions, and the adaptive parameters are the standard deviation of the Normal distributions or log-Normal distributions. The adaptive parameters are updated every 50 iterations; and Gibbs Sampling draws each element ζ_i at iteration n conditionally on the fixed value of $\left(\zeta_1^{(n)},\ldots,\zeta_{i-1}^{(n)},\zeta_{i+1}^{(n-1)},\ldots,\zeta_M^{(n-1)}\right)$, where $\zeta_j^{(n)}$ refers to the sampling values of ζ_j at iteration n. In the following algorithm, the proposal distribution is a log-Normal distribution.

Algorithm 1 Adaptive Metropolis-within-Gibbs Sampler

Require: Data generating model $\pi(\boldsymbol{y}|\boldsymbol{\zeta})$; prior distribution $\pi(\boldsymbol{\zeta})$; the initial value of $\boldsymbol{\zeta} = (\zeta_1,\ldots,\zeta_M)$, $\boldsymbol{\zeta}^{(0)} = \left(\zeta_1^{(0)},\ldots,\zeta_M^{(0)}\right)$; the standard deviations, $\varsigma_1,\ldots,\varsigma_M$, of the proposal distributions for $\zeta_1,\ldots,\zeta_M$; the number of iterations, N.

Initialise

while n in $(1:N)$ **do**

for m in $(1:M)$ **do**

Draw a candidate value $\zeta_m^* \sim \text{log-Normal}\left(\zeta_m^{(n-1)},\varsigma_m^2\right)$.

Calculate the logarithm of the acceptance ratio r_a according to the marginal distribution $\pi\left(\zeta_m|\boldsymbol{y},\zeta_1^{(n)},\ldots,\zeta_{m-1}^{(n)},\zeta_{m+1}^{(n-1)},\ldots,\zeta_M^{(n-1)}\right)$:

$$\log(r_a) = \log\left[\frac{\pi\left(\zeta_m^*|\boldsymbol{y},\zeta_1^{(n)},\ldots,\zeta_{m-1}^{(n)},\zeta_{m+1}^{(n-1)},\ldots,\zeta_M^{(n-1)}\right)}{\pi\left(\zeta_m^{(n-1)}|\boldsymbol{y},\zeta_1^{(n)},\ldots,\zeta_{m-1}^{(n)},\zeta_{m+1}^{(n-1)},\ldots,\zeta_M^{(n-1)}\right)}\right].$$

Draw a number $r_u \sim \text{Uniform}(0,1)$.

if $\log(r_a) \leq \log(r_u)$ **then**
$\zeta_m^{(n)} = \zeta_m^{(n-1)}$.
else
$\zeta_m^{(n)} = \zeta_m^*$.
end if
if 50 exactly divides n **then**
$\varsigma_m = \varsigma_m + \delta(n)$, where $\delta(n) = \min\left\{0.01, \left(\frac{n}{50}\right)^{-\frac{1}{2}}\right\}$.
end if
end for
Obtain the posterior sample at the n-th iteration: $\boldsymbol{\zeta}^{(n)} = \left(\zeta_1^{(n)}, \ldots, \zeta_M^{(n)}\right)$.
end while
Return the posterior sample $\left\{\boldsymbol{\zeta}^{(1)}, \ldots, \boldsymbol{\zeta}^{(N)}\right\}$.

2.2 Graphical Models

In Chapter 4 and Chapter 5 of this thesis, we extend a Gaussian graphical model for non-Gaussian data applications and developed an efficient algorithm to infer the relationships embedded in data. In this section, we introduce some relevant basic concepts of graphical models (Lauritzen, 1996; Whittaker, 1990).

2.2.1 Graph theory

A graph, or a network, is a mathematical object defined by a pair $G = (V, E)$, where V is a finite set of *vertices* and $E \subseteq V \times V$ is the set of *edges*. An edge in E is written as an ordered pair of distinct vertices, $e = (v_i, v_j) \in E$. According to Lauritzen (1996), if (v_i, v_j) and (v_i, v_j) are both in E for any $v_i, v_j \in V$, then the graph is *undirected*, otherwise it is a *directed* graph. In chapter 4 and chapter 5 of this thesis, we only work on undirected graphs. If $(v_i, v_j) \in E$, v_i and v_j are called *neighbours* in the graph (Kindermann, 1980). In Graph Theory, the structure of a finite graph is commonly represented by a square matrix, called the *adjacency matrix* (Harary, 1962). The value of each element $A_{ij} \in \mathbb{R}$ of the adjacency matrix can represent the strength of the link between v_i and v_j, these values are called *weights* (Acharya, 1980). The closer the

weight A_{ij} is to 0, the weaker the link between v_i and v_j is, so that

$$A_{ij} = 0 \iff (v_i, v_j) \notin E.$$

In other words, there are no direct edges from v_i to v_j when $A_{ij} = 0$. Note that for an undirected graph, the adjacency matrix is symmetric.

According to Girvan & Newman (2002), one of the properties a network can have is *community structure*. A network has this property if its vertices can be partitioned into subsets or *clusters* where the vertices are densely connected within each cluster and loosely connected between different clusters. We present an illustrative example in illustration in Figure 2.1.

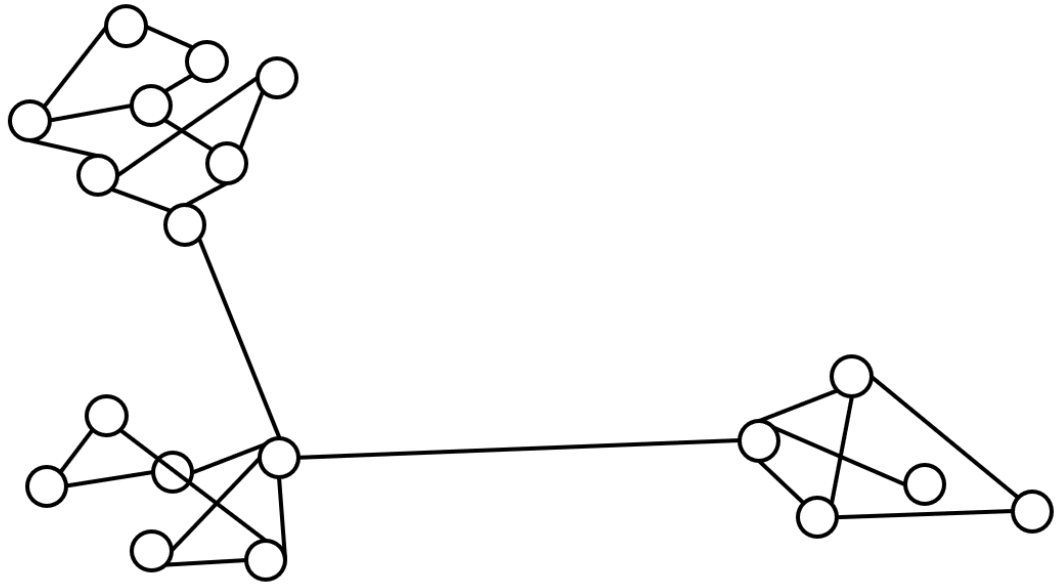

Figure 2.1: An example of a graph with three clusters. We can see that there are many connections inside each cluster, but very few connections between different clusters.

Consider graphs $G_1 = (V_1, E_1), \ldots, G_M = (V_M, E_M)$, then the *Cartesian product* (called the box product $\square$, in Knauer & Knauer (2019)) of $G_1, \ldots, G_M$ is defined as (Sabidussi, 1959a)

$$G_1 \square \ldots \square G_M = (V, E),$$

where

$$V = V_1 \times \cdots \times V_M,$$

and

$$E = \left\{ \left(v_i, v_j\right), \left(v_i, v'_j\right) | v_i \in V_i, \left(v_j, v'_j\right) \in E_j, i, j = 1, \ldots, M \right\} \bigcup \left\{ \left(v_i, v_j\right), \left(v'_i, v_j\right) | \left(v_i, v'_i\right) \in E_i, v_j \in V_j, i, j = 1, \ldots, M \right\}.$$

In other words, for any $v'_i, v'_j \in V$, edge $\left(v'_i, v'_j\right) \in E$ if and only if $\left(v_i, v'_i\right) \in E_i, v_j = v'_j$ with $v_i \in V_i$, or $v_i = v'_i, \left(v_j, v'_j\right) \in E_j$ with $v_j \in E_j$.

For two graphs G_1 and G_2 with adjacency matrices $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$ respectively, according to Cvetković *et al.* (1979), the adjacency matrix of their Cartesian product $G_1 \square G_2$ can be represented as the *Kronecker sum* of $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$:

$$\boldsymbol{A}^{(1)} \oplus \boldsymbol{A}^{(2)} = \boldsymbol{A}^{(1)} \otimes \boldsymbol{I}^{(1)} + \boldsymbol{I}^{(2)} \otimes \boldsymbol{A}^{(2)},$$

where $\boldsymbol{I}^{(1)}$ represents the identity matrix with the same dimension as $\boldsymbol{A}^{(1)}$, $\boldsymbol{I}^{(2)}$ represents the identity matrix with the same dimension as $\boldsymbol{A}^{(2)}$, and $\boldsymbol{A} \otimes \boldsymbol{B}$ represents the *Kronecker product* (also called *tensor product*). For $n_1 \times n_2$ matrix $\boldsymbol{A} = \left(a_{ij}\right)$ and $p_1 \times p_2$ matrix $\boldsymbol{B} = \left(b_{ij}\right)$, the Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B}$ is defined as

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \ldots & a_{1n_2}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \ldots & a_{2n_2}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_1 1}\boldsymbol{B} & a_{n_1 2}\boldsymbol{B} & \ldots & a_{n_1 n_2}\boldsymbol{B} \end{bmatrix}.$$

2.2.2 Markov random field

Given a graph $G = (V, E)$, $V = \{v_1, \ldots, v_n\}$ and a set of random variables $\mathbf{X}_V = \left\{X_{v_1}, \ldots, X_{v_n}\right\}$, where each X_{v_i} corresponds to v_i in the graph, $i = 1, \ldots, n$, we have the following definition for *Markov random field* (Kindermann, 1980).

Definition 2.2 A probability measure $P_{\boldsymbol{\zeta}}$ on $\mathbf{X}_V = \left\{X_{v_1}, \ldots, X_{v_n}\right\}$ is said to define a Markov random field with respect to G if the local characteristics depend only on the knowledge of the neighbours. In other words, for any $v \in V$, if we denote the set of its neighbours as $N(v) = \left\{v' \in V | \left(v, v'\right) \in E \text{ or } \left(v', v\right) \in E\right\}$, then

$$P\left(\mathbf{X}_V | \mathbf{X}_{V \setminus \{v\}}\right) = P\left(\mathbf{X}_V | \mathbf{X}_{V \setminus N(v)}\right),$$

where $V \setminus V_1 = \{v \in V | v \notin V_1\}$.

When the probability measure $P_{\boldsymbol{\zeta}}$ mentioned above is a multivariate normal distribution, $\left(X_{v_1},\ldots,X_{v_n}\right) \sim \mathbf{mN}\left(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_{n\times n}\right)$, with the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}_{n\times n}$, then we have a *Gaussian Markov random field*. The advantage of multivariate normal distributions is that for precision matrix (inverse convariance matrix) $\boldsymbol{\Omega}_{n\times n} = \boldsymbol{\Sigma}^{-1}_{n\times n}$, one has

$$\Omega_{ij} = 0 \Longleftrightarrow \left(v_i, v_j\right) \notin E, \text{for all } i \neq j, i, j = 1,\ldots,n,$$

More formally, we have the following definition for Gaussian Markov random fields. (Rue & Held, 2005)

Definition 2.3 A random vector $\mathbf{X} = \left(X_{v_1},\ldots,X_{v_n}\right)^\top \in \mathbb{R}^n$ is called a Gaussian Markov random field with respect to graph $G = (V,E)$ with mean $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{\Omega}_{n\times n} = \boldsymbol{\Sigma}^{-1}_{n\times n}$, if and only if its density has the form

$$\pi\left(\mathbf{X}\right) = (2\pi)^{-n/2} \left|\boldsymbol{\Omega}_{n\times n}\right|^{1/2} \exp\left[-\frac{1}{2}\left(\mathbf{X}-\boldsymbol{\mu}\right)^\top \boldsymbol{\Omega}_{n\times n}\left(\mathbf{X}-\boldsymbol{\mu}\right)\right],$$

and

$$\Omega_{ij} = 0 \Longleftrightarrow \left(v_i, v_j\right) \notin E, \text{for all } i \neq j, i, j = 1,\ldots,n.$$

For convenience, we call the graph G associated with $\mathbf{X} \sim \mathbf{mN}\left(\boldsymbol{\mu}, \boldsymbol{\Omega}_{n\times n}\right)$ a Gaussian Markov random field graph. The precision matrix $\boldsymbol{\Omega}_{n\times n}$ encodes conditional Independence between random variables $X_{v_1},\ldots,X_{v_n}$. The support of $\boldsymbol{\Omega}_{n\times n}$ describes the structure of a Gaussian Markov random field graph.

Definition 2.4 The support of a $m \times m$ matrix $\mathbf{B}$, $\mathbf{B}^{Support}$, is defined as below:

$$B_{ij}^{Support} = \begin{cases} 1, & B_{ij} \neq 0 \\ 0, & B_{ij} = 0. \end{cases}$$

In the case of precision matrix $\boldsymbol{\Omega}_{n\times n}$, $\Omega_{ij}^{Support} = 1$ indicates the existence of an undirected edge in the corresponding network between node i and node j. Also, since the partial correlation coefficient between X_{v_i} and X_{v_j}, R_{ij}, can be calculated as (Lauritzen, 1996)

$$R_{ij} = -\frac{\Omega_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}},$$

we note that a negative Ω_{ij} indicates positive correlation between v_i and v_j.

When considering the Cartesian product of two Gaussian Markov random field graphs, the statement in Theorem 2.2 has been used in previous works (Greenewald *et al.*, 2019; Kalaitzis *et al.*, 2013). However, to the best of our knowledge, it has not been properly formalised. Here, we introduce and prove Theorem 2.2.

Theorem 2.2 Consider two Gaussian Markov random field graphs G_1 and G_2 defined by the $n \times n$ precision matrix $\mathbf{\Omega}_1$ and the $p \times p$ precision matrix $\mathbf{\Omega}_2$ respectively. Then the Gaussian Markov random field graph G, defined by the Kronecker sum $\mathbf{\Omega} = \mathbf{\Omega}_1 \oplus \mathbf{\Omega}_2$, will be the Cartesian product $G = G_1 \square G_2 = (V, E)$.

Proof. Denote the support of $\mathbf{\Omega}_1$ as $\mathbf{\Omega}_1^{support}$, and the support of $\mathbf{\Omega}_2$ as $\mathbf{\Omega}_2^{support}$, then $\mathbf{\Omega}_1^{support}$ and $\mathbf{\Omega}_2^{support}$ are the adjacency matrices of G_1 and G_2 respectively. According to Cvetković *et al.* (1979), we have $\mathbf{\Omega}_1^{support} \oplus \mathbf{\Omega}_2^{support}$ as the adjacency matrix of $G_1 \square G_2$.

It is also worth to note that the support of $\mathbf{\Omega}_1 \oplus \mathbf{\Omega}_2$ can be written as

$$(\mathbf{\Omega}_1 \oplus \mathbf{\Omega}_2)^{support} = \mathbf{\Omega}_1^{support} \oplus \mathbf{\Omega}_2^{support}.$$

Therefore, we can take a random vector $\mathbf{X} = \left(X_1, \ldots, X_{np}\right)^\top \sim \mathbf{mN}\left(\mathbf{0}, \mathbf{\Omega}^{-1}\right)$ being a Gaussian Markov random field with respect to graph $G = G_1 \square G_2$ defined by the precision matrix $\mathbf{\Omega} = \mathbf{\Omega}_1 \oplus \mathbf{\Omega}_2$, and $G = G_1 \square G_2$ is a Gaussian Markov random field graph. The density of $\mathbf{X}$ has the form

$$\pi(\mathbf{X}) = (2\pi)^{-np/2} |\mathbf{\Omega}|^{1/2} \exp\left(-\frac{1}{2}\mathbf{X}^\top \mathbf{\Omega} \mathbf{X}\right),$$

and

$$\Omega_{ij} = 0 \Longleftrightarrow \left(\mathbf{\Omega}_1^{support} \oplus \mathbf{\Omega}_2^{support}\right)_{ij} = 0 \Longleftrightarrow \left(v_i, v_j\right) \notin E, \text{for all } i \neq j, i, j = 1, \ldots, np.$$

□

2.2.3 Matrix calculus

In this subsection, we list some relevant properties of matrix calculus for later use.

Properties of Kronecker product (Magnus & Neudecker, 1999) Consider matrices **A**, **B**, **C** and **D**, we have

1. *Associativity:* $\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C} = (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C}$;
2. *Bilinearity:* If $\mathbf{A} + \mathbf{B}$ and $\mathbf{C} + \mathbf{D}$ exist, then
$$(\mathbf{A} + \mathbf{B}) \otimes (\mathbf{C} + \mathbf{D}) = \mathbf{A} \otimes \mathbf{C} + \mathbf{A} \otimes \mathbf{D} + \mathbf{B} \otimes \mathbf{C} + \mathbf{B} \otimes \mathbf{D};$$
3. *Mixed Product:* If **AC** and **BD** exist, then
$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}.$$

With these properties, we can extend Theorem 2.2 to the Cartesian product of M Gaussian Markov random field graphs defined by precision matrices $\{\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_M\}$, since

$$\mathbf{\Omega}_1 \oplus \mathbf{\Omega}_2 \oplus \cdots \oplus \mathbf{\Omega}_{M-1} \oplus \mathbf{\Omega}_M = (\mathbf{\Omega}_1 \oplus \mathbf{\Omega}_2 \oplus \cdots \oplus \mathbf{\Omega}_{M-1}) \oplus \mathbf{\Omega}_M$$

can be viewed as the Cartesian product of $G^{(M-1)}$ and G_M, where $G^{(M-1)} = G_1 \square \ldots \square G_{M-1}$. This process can be repeated for $M-1$ times.

Some knowledge of matrix differential calculus is also useful in later chapters. Readers can consult Magnus & Neudecker (1999); Petersen *et al.* (2008) for further details and the full mathematical background. Before introducing matrix differential calculus rules, we define the differentiation with respect to matrix.

Definition 2.5 Consider a $m \times n$ random variable matrix $\mathbf{X} = (X_{ij})$ and a differentiable function f of matrix **X**, the scalar derivative of f with respect to **X** is defined as

$$\frac{\partial f}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial f}{\partial X_{11}} & \cdots & \frac{\partial f}{\partial X_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial X_{m1}} & \cdots & \frac{\partial f}{\partial X_{mn}} \end{bmatrix}.$$

Matrix differential calculus rules (Magnus & Neudecker, 1999; Petersen *et al.*, 2008) Consider random variable matrices **X** and **Y**, a constant matrix **A**, a differentiable function f of matrix **X**. We list some of the most relevant matrix differential calculus rules as follows:

1. $\frac{\partial(\mathbf{AX})}{\partial \mathbf{X}} = \mathbf{A}$;
2. $\frac{\partial \ln|\mathbf{X}|}{\partial \mathbf{X}} = \mathrm{tr}\left(\mathbf{X}^{-1}\right)$, where $|\mathbf{X}|$ represents the determinant of $\mathbf{X}$, and $\mathrm{tr}(\mathbf{X})$ represents the trace of $\mathbf{X}$;
3. $\frac{\partial(\mathbf{X}\otimes\mathbf{Y})}{\partial \mathbf{X}} = \mathbf{I}\otimes\mathbf{Y}$, $\frac{\partial(\mathbf{X}\otimes\mathbf{Y})}{\partial \mathbf{Y}} = \mathbf{X}\otimes\mathbf{I}$;
4. $\frac{\partial \mathrm{tr}[f(\mathbf{X})]}{\partial \mathbf{X}} = \frac{\partial f}{\partial \mathbf{X}}$;
5. If $\mathbf{X}$ is symmetric then we have

$$\frac{\partial f}{\partial \mathbf{X}} = \left[\frac{\partial f}{\partial \mathbf{X}}\right] + \left[\frac{\partial f}{\partial \mathbf{X}}\right]^{\top} - \mathbf{I}\circ\left[\frac{\partial f}{\partial \mathbf{X}}\right],$$

where $\circ$ is the Hadamard product. For $m\times n$ matrices $\mathbf{A} = \left(A_{ij}\right)$ and $\mathbf{B} = \left(B_{ij}\right)$, the Hadamard product $\mathbf{A}\circ\mathbf{B}$ is defined as

$$(\mathbf{A}\circ\mathbf{B})_{ij} = A_{ij}B_{ij}.$$

Furthermore, we note some derivatives with respect to a matrix element X_{ij}:

1. If $\mathbf{X}$ is symmetric, $\frac{\partial \mathbf{X}}{\partial X_{ij}} = \mathbf{J}^{ij} + \mathbf{J}^{ji} - \mathbf{J}^{ij}\mathbf{J}^{ji}$, where in the matrix $\mathbf{J}^{ij}$, the only element different from zero is $J_{ij} = 1$;
2. $\frac{\partial \mathbf{X}\otimes\mathbf{A}}{\partial X_{ij}} = \frac{\partial \mathbf{X}}{\partial X_{ij}}\otimes\mathbf{A}$.

We also define the Frobenius norm of $m\times n$ matrix $\mathbf{A} = \left(A_{ij}\right)$ as

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|A_{ij}\right|^2}.$$

Finally, consider the $m\times n$ matrix $\mathbf{A} = \left(A_{ij}\right)$, the *vectorisation* of $\mathbf{A}$ is defined as stacking each column of $\mathbf{A}$ into a vector of length mn, in other words,

$$\mathrm{vec}(\mathbf{A}) = (A_{11}, A_{21}, \ldots, A_{m1}, \ldots, A_{1n}, \ldots, A_{mn})^{\top}.$$

2.2.4 Background on tensors

To work on K-way *tensor*-valued data in Chapter 5, here we introduce some mathematical background on tensor calculation. In simple words, tensors are a generalisation of vectors (Renteln, 2013). Formally, we define a tensor as follows:

Definition 2.6 A K-way or K-th order tensor $\mathcal{Y}$ is an element of the tensor product of K vector spaces. In other words, there exists $\mathbf{Y}_1 \in \mathbb{R}^{d_1}, \ldots, \mathbf{Y}_K \in \mathbb{R}^{d_K}$, such that

$$\mathcal{Y} = \mathbf{Y}_1 \otimes \cdots \otimes \mathbf{Y}_K.$$

A one-way tensor is a vector, a two-way tensor is a matrix, when $K \geq 3$, we call it a multi-way tensor.

A *fibre* of a tensor is the higher-order analogue of rows and columns. A fibre is obtained by fixing all indices but one. Consider a K-way tensor $\mathcal{Y}$, of which the dimensions are $d_1 \times d_2 \times \cdots \times d_K$. A *mode-$k$* fibre can be denoted as $\mathbf{Y}_{i_1,\ldots,i_{k-1},i_{k+1},\ldots,i_K}$, and it is a vector of length d_k. A *slice* of a tensor is the two-way sections of a tensor, defined by fixing all but two indices. A slice defined by fixing all but indices in mode-k_1 and mode-k_2 can be denoted as $\mathbf{Y}_{:,\ldots,:,i_{k_1},:,\ldots,:,i_{k_2},:,\ldots,:}$ $(k_1 < k_2, k_1, k_2 = 1, \ldots, K)$.

The *matricization* of $\mathcal{Y}$ along mode k (i.e. the k-th way) $\mathbf{Y}^{(k)}$ is obtained by arranging all the mode-k fibres of $\mathcal{Y}$. The resulting $\mathbf{Y}^{(k)}$ is a $d_k \times m_k$ matrix, where $m_k = \frac{\prod_{i=1}^{K} d_i}{d_k}$ (Kolda & Bader, 2009). Figure 2.2 illustrates the three different types of slices for a three-way tensor.

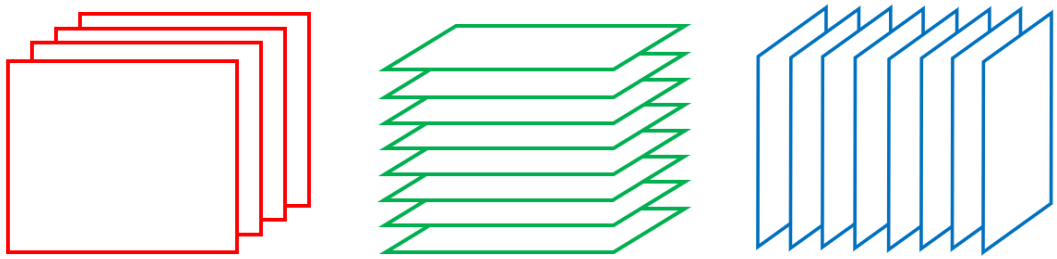

Figure 2.2: Types of slices for a three-way tensor $\mathcal{Y}$.
(a): Horizontal slices: $\mathbf{Y}_{i_1,:,:}$, $i_1 = 1, \ldots, d_1$.
(b): Lateral slices: $\mathbf{Y}_{:,i_2,:}$, $i_2 = 1, \ldots, d_2$.
(c): Frontal slices; $\mathbf{Y}_{:,:,i_3}$, $i_3 = 1, \ldots, d_3$.

Example 2.1 Let us consider a three-way tensor $\mathcal{Y} \in \mathbb{R}^{3\times4\times2}$ with frontal slices $\mathbf{Y}_{:,:,1}$ and $\mathbf{Y}_{:,:,2}$ as follows:

$$\mathbf{Y}_{:,:,1} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix},$$

and

$$\mathbf{Y}_{:,:,2} = \begin{bmatrix} 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 \end{bmatrix}.$$

Then the matricization along mode-1 is

$$\mathbf{Y}^{(1)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 13 & 14 & 15 & 16 \\ 5 & 6 & 7 & 8 & 17 & 18 & 19 & 20 \\ 9 & 10 & 11 & 12 & 21 & 22 & 23 & 24 \end{bmatrix},$$

the matricization along mode-2 is

$$\mathbf{Y}^{(2)} = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 \\ 2 & 6 & 10 & 14 & 18 & 22 \\ 3 & 7 & 11 & 15 & 19 & 23 \\ 4 & 8 & 12 & 16 & 20 & 24 \end{bmatrix},$$

and the matricization along mode-3 is

$$\mathbf{Y}^{(3)} = \begin{bmatrix} 1 & 5 & 9 & 2 & 6 & 10 & 3 & 7 & 11 & 4 & 8 & 12 \\ 13 & 17 & 21 & 14 & 18 & 22 & 15 & 19 & 23 & 16 & 20 & 24 \end{bmatrix}.$$

2.2.5 Gaussian copulas

Gaussian *copulas* are commonly used to extend Gaussian models to applications on non-Gaussian data. Here we first introduce the definition of copulas then give some further theoretical background on copulas.

Definition 2.7 (Sklar, 1973) An M-dimensional copula is a function $C : [0,1]^M \mapsto [0,1]$, which satisfies the following conditions:

1. $C(1,\ldots,1,u_m,1,\ldots,1) = u_m$ for each $m \leq M$ and all $u_m \in [0,1]$;
2. $C(u_1,\ldots,u_M) = 0$ if $u_m = 0$ for any $m \leq M$;
3. C is increasing in each component $u_m \in [0,1]$.

The theoretical foundation of copulas is given by Sklar's Theorem (Sklar, 1973):

Theorem 2.3 [Sklar's Theorem] For any random variables $X_1, \ldots, X_m$ with joint Cumulative Density Function (CDF) $F(x_1, \ldots, x_m)$ and marginal CDFs $F_1(x_1) = P(X_1 \leq x_1)$, $\ldots, F_m(X_m) = P(X_m \leq x_m)$, there exists a copula such that

$$F(x_1, \ldots, x_m) = C(F_1(x_1), \ldots, F_m(x_m)).$$

Liu *et al.* (2009, 2012) defined the nonparanormal distribution:

Definition 2.8 A random vector $\mathbf{X} = (X_1, \ldots, X_m)^\top$ has a nonparanormal distribution if there exist a set of monotone and univariate functions $\{f_k\}_{k=1}^m$ such that $\mathbf{Z} \equiv f(\mathbf{X}) \sim \mathbf{mN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $f(\mathbf{X}) = (f_1(X_1), \ldots, f_m(X_m))$. It then can be written that

$$\mathbf{X} \sim \mathbf{NPN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, f).$$

Following Sklar's Theorem, Liu *et al.* (2009) mentioned that for the nonparanormal distribution, we have

$$F(X_1, \ldots, X_m) = \Phi_{\boldsymbol{\mu}, \boldsymbol{\Sigma}}\left(\Phi^{-1}(F_1(X_1)), \ldots, \Phi^{-1}(F_m(X_m))\right),$$

where $\Phi_{\boldsymbol{\mu}, \boldsymbol{\Sigma}}$ is the multivariate Gaussian CDF and Φ is the univariate standard Gaussian CDF.

Furthermore, we can deduce the corresponding copula

$$C(F_1(X_1), \ldots, F_m(X_m)) = \Phi_{\boldsymbol{\mu}, \boldsymbol{\Sigma}}\left(\Phi^{-1}(F_1(X_1)), \ldots, \Phi^{-1}(F_m(X_m))\right).$$

This leads to $\mathbf{X} \sim \mathbf{NPN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, f)$, where $f = \{f_k\}_{k=1}^m$, $f_k(X_k) = \Phi^{-1}(F_k(X_k))$, and

$$(f_1(X_1), \ldots, f_m(X_m)) \sim \mathbf{mN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

Now we consider the $p \times n$ random matrix $\mathbf{Y} = (Y_{ij})$, $i = 1, \ldots, p$, $j = 1, \ldots, n$. For each row vector of $\mathbf{Y}$, $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{in})^\top$, $i = 1, \ldots, p$, we consider the CDFs of the marginal distributions $F_1^{(r)}, \ldots, F_j^{(r)}, \ldots, F_n^{(r)}$, where the superscript (r) denotes marginal distributions in row vectors. Then by Sklar's theorem, for the CDF of a n-dimensional multivariate normal distribution $\Phi_{(\mathbf{0}_n, \boldsymbol{\Psi}_{n\times n}^{-1})}$, there exists copula $C^{(r)}$ such that

$$\Phi_{\{\mathbf{0}_n, \boldsymbol{\Psi}_{n\times n}^{-1}\}}\left(\Phi^{-1}\left(F_1^{(r)}(Y_{i1})\right), \ldots, \Phi^{-1}\left(F_n^{(r)}(Y_{in})\right)\right) = C^{(r)}\left(F_1^{(r)}(Y_{i1}), \ldots, F_n^{(r)}(Y_{in})\right).$$

That is, there exist functions $f^{(r)} = \left\{f_j^{(r)}\right\}_{j=1}^{n}$ such that for each row vectors of $\mathbf{Y}$, $\mathbf{Y}_i = (Y_{i1},\ldots,Y_{in})^\top$, $i = 1,\ldots,p$, $Z_i^{(r)} \equiv f^{(r)}(\mathbf{Y}_i) \sim \mathbf{mN}\left(\mathbf{0}_n, \mathbf{\Psi}_{n\times n}^{-1}\right)$, where $f^{(r)}(\mathbf{Y}_i) = \left(f_1^{(r)}(Y_{i1}),\ldots,f_n^{(r)}(Y_{in})\right)$. Then we say $\mathbf{Y}_i = (Y_{i1},\ldots,Y_{in})^\top$ has a nonparanormal distribution and write

$$\mathbf{Y}_i \sim \mathbf{NPN}\left(\mathbf{0}_n, \mathbf{\Psi}_{n\times n}^{-1}, f^{(r)}\right).$$

Lemma 2.1 (Liu *et al.*, 2009) If $\mathbf{X} \sim \mathbf{NPN}\left(\boldsymbol{\mu}, \mathbf{\Sigma}, f\right)$ is nonparanormal and each f_k is differentiable, then $\mathbf{X}_{k_1}$ is independent of $\mathbf{X}_{k_2}$ conditionally on all the other element X_k in the vector $\mathbf{X}$, if and only if $B_{k_1k_2} = 0$, where $\boldsymbol{B} = \mathbf{\Sigma}^{-1}$.

According to Lemma 2.1 (Liu *et al.*, 2009)[Lemma 2], for $\mathbf{Y}_i \sim \mathbf{NPN}\left(\mathbf{0}_n, \mathbf{\Psi}_{n\times n}^{-1}, f^{(r)}\right)$, the dependencies between $Y_{i1}, \ldots, Y_{in}, i = 1,\ldots,p$ can be illustrated by a Gaussian Markov random field Graph $G_r = (V_r, E_r)$ corresponding to precision matrix $\mathbf{\Psi}_{n\times n}$. This is equivalent to having the latent variable $\mathbf{Z}^{(r)} = f^{(c)}(\mathbf{Y}_i) \sim \mathbf{mN}\left(\mathbf{0}_n, \mathbf{\Psi}_{n\times n}^{-1}\right), i = 1,\ldots,p$.

Similarly, for each column vector of $\mathbf{Y}$, $\mathbf{Y}_j = \left(Y_{1j},\ldots,Y_{pj}\right)^\top$, $j = 1,\ldots,n$, we consider the CDF of marginal distributions $F_1^{(c)},\ldots,F_i^{(c)},\ldots,F_n^{(c)}$, where the superscript (c) denotes marginal distributions in column vector. Then by Sklar's theorem, for the CDF of a p-dimensional multivariate normal distribution $\Phi_{\left(\mathbf{0}_p, \mathbf{\Psi}_{p\times p}^{-1}\right)}$, there exists copula $C^{(c)}$ such that

$$\Phi_{\left(\mathbf{0}_p, \mathbf{\Theta}_{p\times p}^{-1}\right)}\left(\Phi^{-1}\left(F_1^{(c)}\left(Y_{1j}\right)\right),\ldots,\Phi^{-1}\left(F_p^{(c)}\left(Y_{pj}\right)\right)\right) = C^{(c)}\left(F_1^{(c)}\left(Y_{1j}\right),\ldots,F_n^{(c)}\left(Y_{pj}\right)\right).$$

That is, there exist functions $f^{(c)} = \left\{f_i^{(c)}\right\}_{i=1}^{p}$ such that for each column vector of $\mathbf{Y}$, $\mathbf{Y}_j = \left(Y_{1j},\ldots,Y_{pj}\right)^\top$, $j = 1,\ldots,n$, $\mathbf{Z_j^{(c)}} \equiv f^{(c)}\left(\mathbf{Y}_j\right) \sim \mathbf{mN}\left(\mathbf{0}_p, \mathbf{\Theta}_{p\times p}^{-1}\right)$, where $f^{(c)}\left(\mathbf{Y}_j\right) = \left(f_1^{(c)}\left(Y_{1j}\right),\ldots,f_p^{(c)}\left(Y_{pj}\right)\right)$. Then we say $\mathbf{Y}_j = \left(Y_{1j},\ldots,Y_{pj}\right)^\top$ has a nonparanormal distribution and write

$$Y_j \sim \mathbf{NPN}\left(\mathbf{0}_p, \mathbf{\Theta}_{p\times p}^{-1}, f^{(c)}\right).$$

The dependence between $Y_{1j}, \ldots, Y_{pj}$ can be illustrated by a Gaussian Markov random field Graph $G_c = (V_c, E_c)$ corresponding to precision matrix $\mathbf{\Theta}_{p\times p}$. This is equivalent to having the latent variable $\mathbf{Z}^{(c)} = f^{(c)}(\mathbf{Y}_j) \sim \mathbf{mN}\left(\mathbf{0}_p, \mathbf{\Theta}_{p\times p}^{-1}\right), j = 1,\ldots,n$.

In order to understand the whole picture of the dependency structure in $p \times n$ dataset $\mathbf{Y}$, we combine the dependency structure in rows and the dependency

structure in columns by considering the Cartesian product between G_c and G_r:

$$G_c \square G_r = (V_r \times V_c, \{(v_{i_1}, v_{i_2}), (v_{i_1}, v'_{i_2}) | v_{i_1} \in G_c, (v_{i_2}, v'_{i_2}) \in E_r\} \bigcup \{(v_{i_1}, v_{i_2}), (v'_{i_1}, v_{i_2}) | v_{i_2} \in G_r, (v_{i_1}, v'_{i_1}) \in E_c\}).$$

According to Theorem 2.2 in Subsection 2.2.2, For $G = G_c \square G_r$, we can find a $p \times n$ random matrix $\mathbf{Z}$, such that $\text{vec}(\mathbf{Z}) \sim \mathbf{mN}\left(0, \left(\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}\right)^{-1}\right)$ and defines the Gaussian Markov random graph G. We propose to view $\mathbf{Z}$ as the latent variable projection of $\mathbf{Y}$. More specifically, for $\text{vec}(\mathbf{Y}) = \left(Y_{11}, \ldots, Y_{ij}, \ldots, Y_{pn}\right)$, consider the cumulative density function of marginal distributions in $\text{vec}(\mathbf{Y})$: $F_{11}, \ldots, F_{ij}, \ldots, F_{pn}$. Then by Sklar's theorem, for the cumulative density function of the np-dimensional distribution $\Phi_{\left(\mathbf{0}_{np}, \left(\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}\right)^{-1}\right)}$, there exists copula C such that

$$\Phi_{\left(\mathbf{0_{np}}, \left(\mathbf{\Psi_{n\times n}} \oplus \mathbf{\Theta_{p\times p}}\right)^{-1}\right)}\left(\Phi^{-1}\left(F_{11}\left(Y_{11}\right)\right), \ldots, \Phi^{-1}\left(F_{pn}\left(Y_{pn}\right)\right)\right) = C\left(F_{11}\left(Y_{11}\right), \ldots, F_{pn}\left(Y_{pn}\right)\right),$$

where $\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}$ is the corresponding precision matrix. That is, there exists functions $f = \left\{f_{ij}\right\}_{\{i,j\}}$ such that for $\text{vec}(\mathbf{Y}) = \left(Y_{11}, \ldots, Y_{pn}\right)$, $\text{vec}(\mathbf{Z}) \equiv f(\text{vec}(\mathbf{Y})) \sim \mathbf{mN}\left(\mathbf{0}_{np}, \mathbf{\Omega}^{-1}\right)$, where $f(\text{vec}(\mathbf{Y})) = \left(f_{11}\left(Y_{11}\right), \ldots, f_{pn}\left(Y_{pn}\right)\right)$. Therefore, the Gaussian Markov random field graph associated with the precision matrix $\mathbf{\Psi}_{n\times n} \otimes \mathbf{\Theta}_{p\times p}$ represents the overall dependency structure encoded in $\mathbf{Y}$.

Chapter 3

Simulation-based Evaluation of the Reliability of Bayesian Hierarchical Models for scRNAseq Data

3.1 Introduction

Bayesian Hierarchical Models (BHM) take into account relations between variables (Congdon, 2014) by assuming a joint probability distribution for a set of parameters to be related to the observation of interest. Lately, BHMs have been used in biomedical applications. Using the term "reliability" to indicate a methodology's ability to recover the "ground truth" or the underlying distribution embedded in the data, we note that validating the reliability of BHMs with high-dimensional parameters is a challenging task, especially when applied to noisy biological data. *Single-cell RNA sequencing* (scRNAseq) is a recent technique to quantify RNA molecules at single-cell level, thus providing insights into the gene expression profile of each cell (Tang *et al.*, 2009).

A recent example of BHM applied to scRNAseq data is the Bayesian Analysis of Single-Cell Sequencing Data (BASiCS) framework introduced in Vallejos *et al.* (2015, 2016), Eling *et al.* (2018). BASiCS aims to provide a structural method to analyse scRNAseq count data while separating various latent variables affecting gene expression, and therefore detecting gene expression heterogeneity in downstream analysis. In its early release, BASiCS was introduced as a non-regression model (Vallejos *et al.*, 2015, 2016), assuming independence between the mean and variance

factors in the model. The latest version of BASiCS is presented as a regression model in Eling *et al.* (2018), considering the confounding effect between mean and variance (Brennecke *et al.*, 2013). The downstream analysis in this framework depends on the posterior inference of the variables in the BHM.

However, due to the complexity of BHMs and the high-dimensional nature of biological data, these models can be computationally expensive and time-consuming. When constructing these models, the choice of the prior for the latent variables is rarely assessed. One of the greatest advantages of Bayesian statistics is to adjust our belief according to the data we are given. The idea is that the posterior would be closer to the "truth" when observing enough data. Given the computational cost associated to testing any quantitative result with biological experiments, it is worth investigating whether these complex models combined with high-dimensional data guarantee a reliable result or at least a result with acceptable uncertainty.

In this chapter, we examine the reliability of both the non-regression BASiCS (Vallejos *et al.*, 2015, 2016) and the regression BASiCS (Eling *et al.*, 2018). Both BASiCS models propose the posterior median to estimate relevant variables in the downstream analysis. To validate this estimation, we work on synthetic datasets generated from the corresponding prior model. To explore the influence of prior specification, we also modify the original `R` package to introduce a continuous range of choices for the prior distribution of the biological variation variable, in order to test the model robustness under perturbed prior models. Finally, we show how the Simulation-based Calibration method recently developed in Talts *et al.* (2018) can be adapted here to validate high-dimensional parameter inferences.

3.1.1 Biological background

Single-cell RNA sequencing

Many phenotypes are defined by proteins, while the type and quantity of proteins are determined by gene expression at the cellular level. When a gene is expressed, the corresponding segment of DNA is transcribed into a type of RNA as an intermediate step to be translated into proteins. These RNAs are called Messenger RNAs (Blackburn *et al.*, 2006). Therefore, counting the number of corresponding Messenger RNAs in the cell can indicate the expression activity of the corresponding gene.

The high-throughout sequencing technique developed by Margulies *et al.* (2005) enabled RNA sequencing. Traditional bulk RNA sequencing is performed in a pool of cell populations extracted from tissues. The RNA information is averaged over millions of individual cells, which can mask biologically important heterogeneity across cells. In 2009, Tang *et al.* (2009) published the first RNA sequencing study at the single-cell level. Since then, many single-cell RNA sequencing (scRNAseq) protocols have been developed (Bhargava *et al.*, 2013; Hashimshony *et al.*, 2012; Islam *et al.*, 2011, 2012; Macosko *et al.*, 2015; Nakagawa & Hashimoto, 2020; Sasagawa *et al.*, 2013). Most current protocols follow the following steps:

1. Isolate single cells.
2. Lyse the cells.
3. Reverse transcription to obtain cDNA from mRNA, thus capture the information of mRNA.
4. Pre-amplify the cDNAs.
5. Prepare cDNA libraries for sequencing.
6. Quantitative Analysis.

The process of cDNA amplification can result in a disproportional representation of all the cDNAs in the sample, thereby affecting the downstream analysis. Therefore, all the new protocols developed in the past few years incorporate unique molecule identifiers (UMIs) into the primer oligonucleotide used for transcription reversion. Here, the oligonucleotide molecules are used to build cDNAs, like bricks used to build houses. UMIs barcode the cDNA obtained from each mRNA in the cell. The number of copies of an mRNA in a given cell lysate is hence equivalent to the number of UMIs that map to the particular mRNA (Fan *et al.*, 2015; Islam *et al.*, 2014; Jaitin *et al.*, 2014).

Currently, single-cell RNA sequencing still faces a few challenges:

1. Low capture efficiency of cDNAs.
2. Current methods to obtain single cells from tissue would introduce bias by changing the environment of cells.

3. The validation methods for quantitative analysis results are limited.

In this chapter we address challenge 3, focusing on a BHM framework BASiCS (Eling *et al.*, 2018; Vallejos *et al.*, 2015, 2016) for the quantitative analysis for scRNAseq data. BASiCS makes use of information from *spike-in* genes, which we introduce in the next subsection.

Spike-in genes

Existing biological experimental techniques still introduce technical noises. In the past few decades, external controls have been used to assess data quality (Fodor *et al.*, 1993, 1991; Heid *et al.*, 1996; Higuchi *et al.*, 1993; Lockhart *et al.*, 1996; Schena *et al.*, 1995; Wittwer *et al.*, 1997). Such "external control" are external molecules mixed into their experimental sample at an early stage. The technical variation that occurs in the experiment can be assessed through the measurement of these external molecules, also called "spike-in molecules".

However, for years, the spike-in molecules used in the experiments were developed specifically for different platforms, which brings limited utility of such traceable references. The lack of a standardised reference material also makes it difficult to compare results across different protocols and platforms (Devonshire *et al.*, 2010). To expedite the approval of newly recognised bio-markers in diagnostics and drug discovery, regulatory bodies also require standardised results with reference material (FDA, 2006).

Since 2003, the External RNA Control Consortium has been developing a set of RNA standards to be used as a true industry-wide standard control (Baker *et al.*, 2005; Devonshire *et al.*, 2010; External-RNA-Controls-Consortium, 2005). Brennecke *et al.* (2013) proposes to embed the data obtained from ERCC spike-ins into the quantitative analysis step, separating biological and technical variation of gene expression counts (Brennecke *et al.*, 2013). Since then, ERCC spike-ins have become an important tool for noise inference and quality control.

3.1.2 BASiCS framework

The non-regression BASiCS model

The BASiCS framework (Vallejos *et al.*, 2015, 2016) aims to provide a structural method to analyse scRNAseq count data to detect highly variable genes (HVGs) and lowly variable genes (LVGs). HVGs are expressed differently across cells because they specialise in specific functions of certain cells. On the opposite, LVGs are expressed at a stable level across cells, as they participate in general cellular activities.

A $q \times n$ scRNAseq count matrix $\mathbf{Y}$, with n cells, q_0 biological genes and $q - q_0$ spike-in genes, can be represented as follows:

$$\mathbf{Y} = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & \dots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{q1} & Y_{q2} & \dots & Y_{qn} \end{bmatrix},$$

where Y_{ij} represents the mRNA count of gene i in cell j. Here we consider q_0 biological genes, which are naturally in the cells, and $q - q_0$ spike-in genes, which are added during the experiment, in specific amounts, to help quantify the technical noise. During the count process, all the mRNA molecules in each cell j are analysed to see if they are associated to gene i. Therefore, a binomial process is a noted choice for the generation of these datasets.

More formally, consider the Bernoulli process of going through all the N_j mRNA molecules in the cell j to check if each of them correspond to gene i. Let Y_{ij} denote the total number of successes in a large number N_j of Bernoulli trials with low successful probability p_{ij}, where p_{ij} is low, because there is a large amount of different genes in a cell. Then $Y_{ij} \sim \text{Binomial}\left(N_j, p_{ij}\right)$.

$$P\left(Y_{ij} = k\right) = \binom{N_j}{k} p_{ij}^k (1 - p_{ij})^{N_j - k}, k = 0, 1, \dots, N_j.$$

When $N_j \to +\infty$, $p_{ij} \to 0$, and $\lambda_{ij} = N_j p_{ij} > 0$, we have

$$\begin{aligned}\lim_{N_j \to +\infty} P\left(Y_{ij} = k\right) &= \lim_{N_j \to +\infty} \binom{N_j}{k} \left(\frac{\lambda_{ij}}{N_j}\right)^k \left(\frac{N_j - \lambda_{ij}}{N_j}\right)^{N_j - k} \\ &= \lim_{N_j \to +\infty} \frac{\lambda_{ij}^k}{k!} \frac{N_j!}{(N_j - k)!} \left(1 - \frac{\lambda_{ij}}{N_j}\right)^{N_j} \left(1 - \frac{\lambda_{ij}}{N_j}\right)^{-k} \\ &= \frac{\lambda_{ij}^k \exp(-\lambda_{ij})}{k!},\end{aligned}$$

since

$$\lim_{N_j \to +\infty} \frac{N_j!}{(N_j - k)!} = 1,$$

$$\lim_{N_j \to +\infty} \left(1 - \frac{\lambda_{ij}}{N_j}\right)^{N_j} = \exp\left(-\lambda_{ij}\right),$$

and

$$\lim_{N_j \to +\infty} \left(1 - \frac{\lambda_{ij}}{N_j}\right)^{-k} = 1.$$

Thus, the distribution of Y_{ij} ($i = 1, ..., q,\ j = 1, ..., n$) can be approximated by a Poisson distribution Poisson$\left(N_j p_{ij}\right)$ (Poisson, 1837). This explains why within the BASiCS framework, the gene i's expression count in cell j, Y_{ij}, is modelled via a Poisson distribution.

Vallejos *et al.* (2015, 2016) and Eling *et al.* (2018) note that the expected count of gene i's expression in cell j can be affected by several factors as listed in Table 3.1. The BASiCS framework then relies on a number of distributional assumptions for Y_{ij} and related parameters, linked to factors in Table 3.1, and described below as Assumptions 3.1-3.5. To sum up, a schematic representation of this hierarchical model is given in Figure 3.1.

Assumption 3.1 The unexplained technical noise only depends on cell-specific characteristics. For a given cell j, it affects the expression counts of all genes $i = 1, ..., q$ in the same manner. The expected count of gene i's expression in cell j could be affected by several factors as listed in Table 3.1. BASiCS assumes the following likelihood model:

$$Y_{ij} \left| \mu_i, \rho_{ij}, \Phi_j, \nu_j \right. \overset{ind}{\sim} \begin{cases} \text{Poisson}(\mu_i \Phi_j \nu_j \rho_{ij}), & \text{for } i \in \{1, ..., q_0\}, \\ \text{Poisson}\left(\mu_i \nu_j\right), & \text{for } i \in \{q_0 + 1, ..., q\}. \end{cases} \tag{3.1}$$

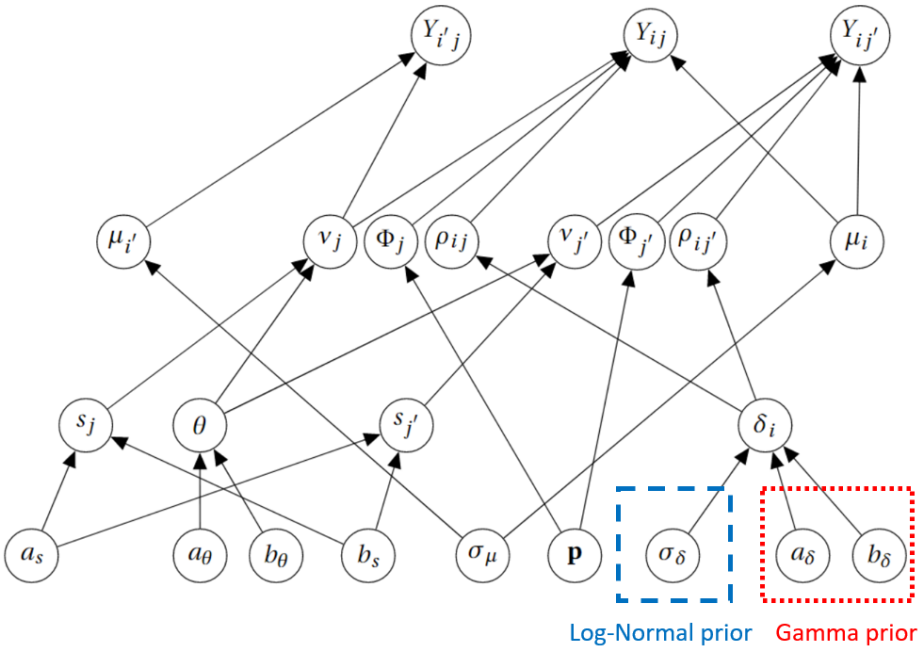

Figure 3.1: The directed acyclic graph of the non-regression BASiCS model. The two choices for the prior distribution for δ_i, log-Normal and Gamma distributions, are depicted. In the graph, we have biological genes $i \in \{1,\ldots,q_0\}$, spike-in genes $i' \in \{q_0+1,\ldots,q\}$ and cells $j, j' \in \{1,\ldots,n\}$.

Assumption 3.2 The expression variability of gene i in cell j follows a Gamma distribution depending on only gene i, but varies across different cells. In particular,

$$\rho_{ij}|\delta_i \overset{ind}{\sim} \text{Gamma}\left(\frac{1}{\delta_i}, \frac{1}{\delta_i}\right),\ i = 1,...,q_0,\ j = 1,...,n, \tag{3.2}$$

$$\text{E}(\rho_{ij}) = 1,$$

$$\text{Var}(\rho_{ij}) = \delta_i.$$

The biological variation factor for biological gene $i \in \{1,\ldots,q_0\}$ across all cells, δ_i, has two possible options for prior (Vallejos *et al.*, 2015, 2016):

$$\textbf{Log-normal prior:} \quad \delta_i|\sigma_\delta \overset{ind}{\sim} \text{log-Normal}\left(0, \sigma_\delta^2\right), \tag{3.3}$$

$$\textbf{Gamma prior:} \quad \delta_i|a_\delta, b_\delta \overset{ind}{\sim} \text{Gamma}\left(a_\delta, b_\delta\right), \tag{3.4}$$

with the corresponding standard deviation, shape and rate parameters $\sigma_\delta, a_\delta, b_\delta > 0$.

Variables	Biological gene $i = 1, ..., q_0$ in cell j	Spike-in gene $i = q_0 + 1, ..., q$ in cell j	Reason
The expected expression count of gene i, μ_i, $i = 1, \ldots, q_0, q_0 + 1, \ldots, q$.	√	√	For biological gene $i = 1, \ldots, q_0$, an overall expression rate across all cells. For spike-in gene $i = q_0 + 1, \ldots, q$, the amount of spike-in molecules that are added is known.
The size of cell j, Φ_j, $j = 1 \ldots, n$.	√		A larger cell size could indicate that the cell is in the later phase of the cell cycle, where it produces more protein to support its division, thereby we expect more total mRNA counts from biological genes in the cell. However, the number of spike-in molecules injected into each cell is constant and not affected.
The cell-to-cell unexplained technical noise ν_j, $j = 1, \ldots, n$.	√	√	Technical noise occurs when we prepare every cell j individually, therefore it affects all gene expression counts in each given cell j equally.
Heterogeneous expression of gene i in any given cell j, ρ_{ij}, $i = 1, \ldots, q_0$, $j = 1, \ldots, n$.	√		The expression variability of gene i across cells would affect the expression count of gene i in cell j

Table 3.1: Variation sources in gene expression.

Assumption 3.3 The technical variation factor follows a Gamma distribution. In particular,

$$\nu_j | \theta, s_j \overset{ind}{\sim} \text{Gamma}\Big(\frac{1}{\theta}, \frac{1}{s_j \theta}\Big),\ j = 1, ..., n, \tag{3.5}$$

where the shape parameter $\frac{1}{\theta} > 0$, and the rate parameter $\frac{1}{s_j \theta} > 0$, so that

$$\text{E}(\nu_j) = s_j,$$

$$\text{Var}(\nu_j) = s_j^2 \theta.$$

In BASiCS, it is assumed that the general technical noise factor across all cells

$$\theta | a_\theta, b_\theta \sim \text{Gamma}(a_\theta, b_\theta), \tag{3.6}$$

and the technical noise related to a specific cell j,

$$s_j | a_s, b_s \overset{i.i.d.}{\sim} \text{Gamma}(a_s, b_s),\ j = 1, \ldots, n \tag{3.7}$$

with the corresponding shape and rate parameters $a_\theta, b_\theta, a_s, b_s > 0$.

Assumption 3.4 The cell size variable follows a scaled Dirichlet distribution. In particular,

$$(\Phi_1, ..., \Phi_n) | \boldsymbol{p} \sim n\text{Dirichlet}(\boldsymbol{p}), \tag{3.8}$$

where $\boldsymbol{p} = (p_1, ..., p_n)$ is the concentration parameter of the Dirichlet distribution, $p_1, \ldots, p_n > 0$. The Dirichlet prior also restricts that

$$\frac{n}{\sum_{j=1}^{n} \Phi_j} = 1. \tag{3.9}$$

Assumption 3.5 The expected expression count of gene i follows a log-Normal distribution. In particular,

$$\mu_i | \sigma_\mu \overset{ind}{\sim} \text{log-Normal}\left(0, \sigma_\mu^2\right), \tag{3.10}$$

with $\sigma_\mu > 0$.

Identifiability of the non-regression BASiCS

Considering the distributions in (3.1) and (3.2), use the Poisson-Gamma mixture result in Greenwood & Yule (1920). Integrate out ρ_{ij} for $i = 1, \ldots, q_0$:

$$
\begin{aligned}
&\pi(Y_{ij}|\mu_i, \nu_j, \Phi_j, \delta_i) \\
&= \int_0^{+\infty} p_{Poisson}(Y_{ij}|\mu_i, \nu_j, \Phi_j, \rho_{ij}) \cdot p_{Gamma}(\rho_{ij}|\delta_i) \mathrm{d}\rho_{ij} \\
&= \int_0^{+\infty} \frac{\left(\mu_i \nu_j \Phi_j \rho_{ij}\right)^{Y_{ij}}}{Y_{ij}!} \exp\left(-\mu_i \nu_j \Phi_j \rho_{ij}\right) \cdot \frac{\left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}}{\Gamma\left(\frac{1}{\delta_i}\right)} \left(\rho_{ij}\right)^{\frac{1}{\delta_i}-1} \exp\left(-\frac{1}{\delta_i}\rho_{ij}\right) \mathrm{d}\rho_{ij} \\
&= \frac{\left(\mu_i \nu_j \Phi_j\right)^{Y_{ij}}}{Y_{ij}!} \cdot \frac{\left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}}{\Gamma\left(\frac{1}{\delta_i}\right)} \int_0^{+\infty} \exp\left(-\mu_i \nu_j \Phi_j \rho_{ij}\right) \rho_{ij}^{Y_{ij}+\frac{1}{\delta_i}-1} \exp\left(-\frac{1}{\delta_i}\rho_{ij}\right) \mathrm{d}\rho_{ij} \\
&= \frac{\left(\mu_i \nu_j \Phi_j\right)^{Y_{ij}}}{Y_{ij}!} \cdot \frac{\left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}}{\Gamma\left(\frac{1}{\delta_i}\right)} \int_0^{+\infty} \rho_{ij}^{Y_{ij}+\frac{1}{\delta_i}-1} \exp\left[-\left(\mu_i \nu_j \Phi_j + \frac{1}{\delta_i}\right) \cdot \rho_{ij}\right] \mathrm{d}\rho_{ij} \\
&= \frac{\left(\mu_i \nu_j \Phi_j\right)^{Y_{ij}} \left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}}{Y_{ij}!\Gamma\left(\frac{1}{\delta_i}\right)} \int_0^{+\infty} \left[\left(\frac{\delta_i}{\mu_i \nu_j \Phi_j \delta_i + 1}\right)\left(\frac{\mu_i \nu_j \Phi_j \delta_i + 1}{\delta_i}\right)\rho_{ij}\right]^{Y_{ij}+\frac{1}{\delta_i}-1} \\
&\qquad \cdot \exp\left(-\frac{\mu_i \nu_j \Phi_j \delta_i + 1}{\delta_i}\rho_{ij}\right) \mathrm{d}\rho_{ij} \\
&= \frac{\left(\mu_i \nu_j \Phi_j\right)^{Y_{ij}} \left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}}{Y_{ij}!\Gamma\left(\frac{1}{\delta_i}\right)} \int_0^{+\infty} \left(\frac{\delta_i}{\mu_i \nu_j \Phi_j \delta_i + 1}\right)^{Y_{ij}+\frac{1}{\delta_i}} \cdot \left[\frac{\mu_i \nu_j \Phi_j \delta_i + 1}{\delta_i}\rho_{ij}\right]^{Y_{ij}+\frac{1}{\delta_i}-1} \\
&\qquad \cdot \exp\left(-\frac{\mu_i \nu_j \Phi_j \delta_i + 1}{\delta_i}\rho_{ij}\right) \mathrm{d}\left(\frac{\mu_i \nu_j \Phi_j \delta_i + 1}{\delta_i}\rho_{ij}\right) \\
&= \frac{\left(\mu_i \nu_j \Phi_j\right)^{Y_{ij}} \left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}}{Y_{ij}!\Gamma\left(\frac{1}{\delta_i}\right)} \cdot \left(\frac{\delta_i}{\mu_i \nu_j \Phi_j \delta_i + 1}\right)^{Y_{ij}+\frac{1}{\delta_i}} \cdot \Gamma\left(\frac{1}{\delta_i} + Y_{ij}\right) \\
&= \frac{\Gamma\left(\frac{1}{\delta_i} + Y_{ij}\right)}{Y_{ij}!\Gamma\left(\frac{1}{\delta_i}\right)} \cdot \left(\mu_i \nu_j \Phi_j \delta_i\right)^{Y_{ij}} \cdot \left(\frac{1}{\mu_i \nu_j \Phi_j \delta_i + 1}\right)^{Y_{ij}} \cdot \left(\frac{1}{\mu_i \nu_j \Phi_j \delta_i + 1}\right)^{\frac{1}{\delta_i}} \\
&= \frac{\Gamma\left(\frac{1}{\delta_i} + Y_{ij}\right)}{Y_{ij}!\Gamma\left(\frac{1}{\delta_i}\right)} \left(\frac{\mu_i \nu_j \Phi_j \delta_i}{\mu_i \nu_j \Phi_j \delta_i + 1}\right)^{Y_{ij}} \cdot \left(1 - \frac{\mu_i \nu_j \Phi_j \delta_i}{\mu_i \nu_j \Phi_j \delta_i + 1}\right)^{\frac{1}{\delta_i}} \cdot
\end{aligned}
$$

We have the likelihood of the gene expression count of biological genes and spiked-in genes:

$$Y_{ij}|\mu_i,\delta_i,\Phi_j,\nu_j \overset{ind}{\sim} \begin{cases} \text{Neg-Binomial}\left(\frac{1}{\delta_i}, \frac{\Phi_j\nu_j\mu_i}{\Phi_j\nu_j\mu_i+\frac{1}{\delta_i}}\right),\ i=1,...,q_0,\ j=1,...,n; \\ \text{Poisson}\left(\nu_j\mu_i\right),\ i=q_0+1,...,q,\ j=1,...,n. \end{cases} \tag{3.11}$$

The model in (3.11) looks not identifiable, in the sense that it is to be expected that parameters μ_i, ν_j and Φ_j cannot be separately estimated from gene expression data for biological genes, since they appear multiplied as $\mu_i\nu_j\Phi_j$ in the expression above. However, spike-in genes facilitate identifiability.

Firstly, for spiked-in genes, $i = q_0+1,\ldots,q$, we note that the number of spiked-in molecules added to each cell is recorded. Therefore, using the spiked-in information across the cells $j \in \{1,\ldots,n\}$, the posterior distribution of ν_j can be inferred from $Y_{ij} \sim \text{Poisson}\left(\nu_j\mu_i\right)$, $i \in \{q_0+1,\ldots,q\}, j \in \{1,\ldots,n\}$. In particular,

$$\begin{aligned} \pi\left(\nu_j|Y_{ij},\mu_i\right) &\propto \pi(\nu_j)\prod_{i=q_0+1}^{q}\pi(Y_{ij}|\nu_j,\mu_i) \\ &= \frac{\left(\frac{1}{\theta}\right)^{\frac{1}{s_j\theta}}}{\Gamma\left(\frac{1}{\theta}\right)}\nu_j^{\frac{1}{\theta}-1}\exp\left[-\frac{1}{s_j\theta}\nu_j\right]\prod_{i=q_0+1}^{q}\frac{\mu_i^{Y_{ij}}\nu_j^{Y_{ij}}}{Y_{ij}!}\exp\left(-\mu_i\nu_j\right) \\ &= \frac{\left(\frac{1}{\theta}\right)^{\frac{1}{s_j\theta}}\prod_{i=q_0+1}^{q}\mu_i^{Y_{ij}}}{\left[\prod_{i=q_0+1}^{q}\left(Y_{ij}!\right)\right]\Gamma\left(\frac{1}{\theta}\right)}\nu_j^{\left[\sum_{i=q_0+1}^{q}Y_{ij}+\frac{1}{\theta}-1\right]}e^{-\left(\mu_i+\frac{1}{s_j\theta}\right)\nu_j}, \end{aligned}$$

where μ_i and Y_{ij} are known for $i \in \{q_0+1,\ldots,q\}$, $j \in \{1,\ldots,n\}$.

Since $\nu_j, j = 1,\ldots,n$ are inferred with spiked-in information, the remaining identifiability conflict is between $\Phi_j, j = 1,\ldots,n$ and the expected count of biological genes $\mu_i = 1,\ldots,q_0$. However, the restriction (3.9) in Dirichelet distribution ensures the identifiability of $\Phi_j, j = 1,\ldots,n$ and $\mu_i, i = 1,\ldots,q_0$. This restriction imposes an arbitrary scale to Φ_j, but it does not affect the relative differences between the μ_i nor the δ_i.

The regression BASiCS

According to Brennecke *et al.* (2013), a strong relationship is typically observed between the variability and mean estimates. In this case, the interpretation of results

from the model in the previous section would be hindered. As Eling *et al.* (2018) argues, an intuitive approach would be to only compare variability δ_i of those genes with equal mean expression μ_i, but this is sub-optimal, especially when used between groups of cells, as there are a large number of genes expressed differently between populations. One such example given by Eling *et al.* (2018) is that reactive genes that change in mean expression upon changing conditions are excluded from the expression heterogeneity assessment by the intuitive solution. An alternative solution is to directly adjust variability measures to remove the confounding effect between mean and variability. For example, Peat *et al.* (2014) computes the empirical distance between the squared coefficient of variation of gene i (CV_i^2) to the rolling median CV^2 across genes with expression levels similar to gene i. In line with this approach, Eling *et al.* (2018) introduces the following joint prior distribution of the fraction of working gene μ_i and the gene-specific hyperparameter δ_i:

$$\mu_i|\sigma_\mu \overset{ind}{\sim} \log \text{Normal}\left(0, \sigma_\mu^2\right),\ i = 1, ..., q_0, \tag{3.12}$$

$$\delta_i|\mu_i, \boldsymbol{\beta}, \sigma_\delta^2, \lambda_i \overset{ind}{\sim} \log \text{Normal}\left(f(\mu_i), \frac{\sigma_\delta^2}{\lambda_i}\right),\ i = 1, ..., q_0. \tag{3.13}$$

The latter is equivalent to the following nonlinear regression model:

$$\log(\delta_i) = f(\mu_i) + \omega_i,\ i = 1, ..., q_0, \tag{3.14}$$

where $\omega_i|\sigma_\delta^2, \lambda_i \overset{ind}{\sim} \text{Normal}\left(0, \frac{\sigma_\delta^2}{\lambda_i}\right)$ is a latent gene-specific residual over-dispersion parameter, capturing departures from the overall trend across all genes expressed at a given mean expression μ_i. For a gene i, positive ω_i indicates more variation than expected for genes with similar expression level, and negative ω_i indicates less variation than expected for genes with similar expression level.

A similar approach is introduced by DESeq2 (Love *et al.*, 2014) in the context of bulk RNA sequencing. Here the regression BASiCS assumes the trend as follows:

$$f(\mu_i) = \alpha_0 + \alpha_1 \log\left(\mu_i\right) + \sum_{l=1}^{L} \beta_l g_l\left(\log\left(\mu_i\right)\right),\ i = 1, ..., q_0 \tag{3.15}$$

where $\boldsymbol{\beta} = \alpha_0, \alpha_1, \beta_1, ..., \beta_L$ are regression coefficients and $g_1(\cdot), ..., g_L(\cdot)$ represent a set of Gaussian Radial Basis Function (GRBF) kernels.

Definition 3.1 The GRBF kernels are defined as

$$g_l\left(\log\left(\mu_i\right)\right) = \exp\left\{-\frac{1}{2}\left[\frac{\log\left(\mu_i\right) - m_i}{h_l}\right]^2\right\},\ l = 1, ..., L \tag{3.16}$$

where m_l ($l = 1, ..., L$) are location hyperparameters for GRBF kernels and h_l ($l = 1, ..., L$) are scale hyperparameters for GRBF kernels.

Assumption 3.6 A priori, m_l ($l \in \{1, ..., L\}$), h_l ($l \in \{1, ..., L\}$) and σ_δ^2 are fixed. The priors for $\beta = \left(\alpha_0, \alpha_1, \beta_1, \ldots, \beta_L\right)$ in Equation (3.16) and its hyperparameters are proposed as follows:

$$\beta|\sigma^2 \overset{ind}{\sim} \text{Normal}\left(\mathbf{0}, \sigma^2 \boldsymbol{I}\right) \tag{3.17}$$

$$\sigma^2 \overset{ind}{\sim} \text{Inv-Gamma}\left(a_\sigma, b_\sigma\right) \tag{3.18}$$

$$\lambda_i|\eta \overset{ind}{\sim} \text{Gamma}\left(\frac{\eta}{2}, \frac{\eta}{2}\right),\ i \in \{1, \ldots, q_0\}. \tag{3.19}$$

Equations (3.5)-(3.19) forms the regression BASiCS model, as shown in Figure 3.2.

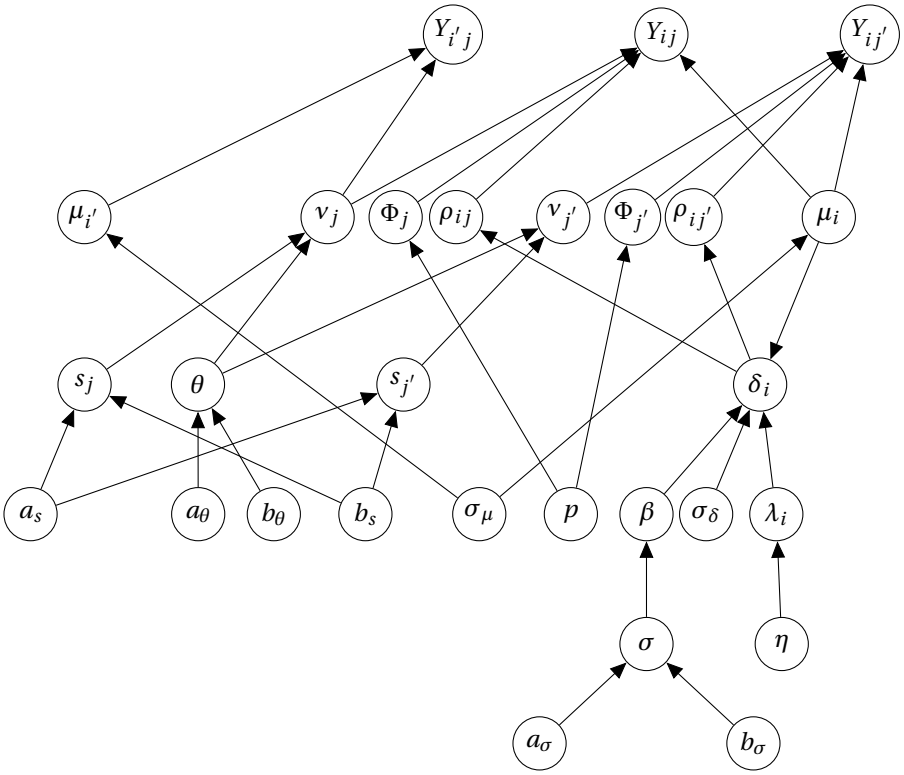

Figure 3.2: the regression BASiCS directed acyclic graph. In the graph, we have biological gene $i \in \{1,\ldots,q_0\}$, spike-in gene $i' \in \{q_0+1,\ldots,q\}$, cells $j, j' \in \{1,\ldots,n\}$.

Gene expression variability

Based on observed data $\mathbf{Y}$ and the prior distribution chosen from the models described in previous subsections, BASiCS infers the posterior distribution of μ_i, ν_j, θ, δ_i, Φ_j and s_j. To calculate the variance of gene expression count Y_{ij} for biological gene $i = 1,\ldots,q_0$ in cell $j = 1,\ldots,n$, we calculate $E\left(Y_{ij}\right)$ and $E\left(Y_{ij}^2\right)$ given the BASiCS model, which gives

$$\begin{aligned}
\mathrm{E}\left(Y_{ij}|\mu_i,\delta_i,\Phi_j,s_j,\theta\right) &= E\left(\mu_i\nu_j\Phi_j\rho_{ij}|\delta_i,s_j,\theta\right) \\
&= \mu_i E\left(\nu_j|s_j,\theta\right)\Phi_j E\left(\rho_{ij}|\delta_i\right) \\
&= \mu_i \cdot \frac{\frac{1}{\theta}}{\frac{1}{s_j\theta}} \cdot \Phi_j \cdot \frac{\frac{1}{\delta_i}}{\frac{1}{\delta_i}} \\
&= \mu_i s_j \Phi_j,
\end{aligned}$$

and

$$
\begin{aligned}
&\mathrm{E}\left[Y_{ij}\left(Y_{ij}-1\right)|\mu_i,\delta_i,\Phi_j,s_j,\theta\right]\\
&=\sum_{Y_{ij}=0}^{+\infty}Y_{ij}\left(Y_{ij}-1\right)\int_0^{+\infty}\int_0^{+\infty}\frac{\left(\mu_i\Phi_j\nu_j\rho_{ij}\right)^{Y_{ij}}}{Y_{ij}!}\exp\left(-\mu_i\nu_j\Phi_j\rho_{ij}\right)\frac{\left(\frac{1}{s_j\theta}\right)^{\frac{1}{\theta}}}{\Gamma\left(\frac{1}{\theta}\right)}\nu_j^{\frac{1}{\theta}-1}\exp\left(-\frac{1}{s_j\theta}\nu_j\right)\\
&\qquad\cdot\frac{\left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}}{\Gamma\left(\frac{1}{\delta_i}\right)}\rho_{ij}^{\frac{1}{\delta_i}-1}\exp\left(-\frac{1}{\delta_i}\rho_{ij}\right)\mathrm{d}\nu_j\mathrm{d}\rho_{ij}\\
&=\int_0^{+\infty}\rho_{ij}^{Y_{ij}+\frac{1}{\delta_i}-1}\exp\left(-\frac{1}{\delta_i}\rho_{ij}\right)\sum_{Y_{ij}=0}^{+\infty}\frac{Y_{ij}\left(Y_{ij}-1\right)\left(\frac{1}{s_j\theta}\right)^{\frac{1}{\theta}}\left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}\mu_i^{Y_{ij}}\Phi_j^{Y_{ij}}}{Y_{ij}!\Gamma\left(\frac{1}{\theta}\right)\Gamma\left(\frac{1}{\delta_i}\right)}\\
&\cdot\int_0^{+\infty}\frac{\left[\left(\mu_i\Phi_j\rho_{ij}+\frac{1}{s_j\theta}\right)\nu_j\right]^{Y_{ij}+\frac{1}{\theta}-1}}{\left(\mu_i\Phi_j\rho_{ij}+\frac{1}{s_j\theta}\right)^{Y_{ij}+\frac{1}{\theta}-1}}\exp\left[-\left(\mu_i\Phi_j\rho_{ij}+\frac{1}{s_j\theta}\right)\nu_j\right]\mathrm{d}\nu_j\mathrm{d}\rho_{ij}\\
&=\int_0^{+\infty}\rho_{ij}^{\frac{1}{\delta_i}-1}\exp\left(-\frac{1}{\delta_i}\rho_{ij}\right)\sum_{Y_{ij}=0}^{+\infty}\frac{Y_{ij}\left(Y_{ij}-1\right)\left(\frac{1}{s_j\theta}\right)^{\frac{1}{\theta}}\left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}\Phi_j^{Y_{ij}}\mu_i^{Y_{ij}}\rho_{ij}^{Y_{ij}}\Gamma\left(Y_{ij}+\frac{1}{\theta}\right)}{Y_{ij}!\Gamma\left(\frac{1}{\theta}\right)\Gamma\left(\frac{1}{\delta_i}\right)\left(\mu_i\Phi_j\rho_{ij}+\frac{1}{s_j\theta}\right)^{Y_{ij}+\frac{1}{\theta}}}\mathrm{d}\rho_{ij}\\
&=\int_0^{+\infty}\frac{\rho_{ij}^{\frac{1}{\delta_i}-1}\exp\left(-\frac{1}{\delta_i}\rho_{ij}\right)\left(\frac{1}{s_j\theta}\right)^{\frac{1}{\theta}}\left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}\frac{1}{\theta}\left(\frac{1}{\theta}+1\right)}{\Gamma\left(\frac{1}{\delta_i}\right)\left(\mu_i\Phi_j\rho_{ij}+\frac{1}{s_j\theta}\right)^{\frac{1}{\theta}}}\sum_{Y_{ij}=2}^{+\infty}\frac{\Gamma\left(Y_{ij}+\frac{1}{\theta}\right)}{\left(Y_{ij}-2\right)!\Gamma\left(\frac{1}{\theta}+2\right)}\left(\frac{\mu_i\Phi_j\rho_{ij}}{\mu_i\Phi_j\rho_{ij}+\frac{1}{\theta}}\right)^{Y_{ij}}\mathrm{d}\rho_{ij}\\
&=\int_0^{+\infty}\frac{\rho_{ij}^{\frac{1}{\delta_i}-1}\exp\left(-\frac{1}{\delta_i}\rho_{ij}\right)\left(\frac{1}{s_j\theta}\right)^{\frac{1}{\theta}}\left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}\frac{1}{\theta}\left(\frac{1}{\theta}+1\right)\mu_i^2\Phi_j^2\rho_{ij}^2}{\Gamma\left(\frac{1}{\delta_i}\right)\left(\mu_i\Phi_j\rho_{ij}+\frac{1}{s_j\theta}\right)^{\frac{1}{\theta}+2}}\sum_{k=0}^{+\infty}\frac{\Gamma\left(k+\frac{1}{\theta}+2\right)}{\Gamma(k+2)\Gamma\left(\frac{1}{\theta}\right)}\left(\frac{\mu_i\Phi_j\rho_{ij}}{\mu_i\Phi_j\rho_{ij}+\frac{1}{\theta}}\right)^{k}\mathrm{d}\rho_{ij}\\
&=\int_0^{+\infty}\frac{\rho_{ij}^{\frac{1}{\delta_i}-1}\exp\left(-\frac{1}{\delta_i}\rho_{ij}\right)\left(\frac{1}{s_j\theta}\right)^{\frac{1}{\theta}}\left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}\frac{1}{\theta}\left(\frac{1}{\theta}+1\right)\mu_i^2\Phi_j^2\rho_{ij}^2}{\Gamma\left(\frac{1}{\delta_i}\right)\left(\mu_i\Phi_j\rho_{ij}+\frac{1}{s_j\theta}\right)^{\frac{1}{\theta}+2}}\frac{1}{\left(1-\frac{\mu_i\Phi_j\rho_{ij}}{\mu_i\Phi_j\rho_{ij}+\frac{1}{s_j\theta}}\right)^{\frac{1}{\theta}+2}}\mathrm{d}\rho_{ij}\\
&=\frac{\mu_i^2\Phi_j^2\left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}}\frac{1}{\theta}\left(\frac{1}{\theta}+1\right)}{\left(\frac{1}{s_j\theta}\right)^2\Gamma\left(\frac{1}{\delta_i}\right)}\int_0^{+\infty}\frac{\left(\frac{1}{\delta_i}\rho_{ij}\right)^{\frac{1}{\delta_i}+1}}{\left(\frac{1}{\delta_i}\right)^{\frac{1}{\delta_i}+1}}\exp\left(-\frac{1}{\delta_i}\rho_{ij}\right)\delta_i\mathrm{d}\left[\left(\frac{1}{\delta_i}\right)\rho_{ij}\right]\\
&=\frac{\mu_i^2\Phi_j^2\frac{1}{\theta}\left(\frac{1}{\theta}+1\right)}{\left(\frac{1}{s_j\theta}\right)^2\Gamma\left(\frac{1}{\delta_i}\right)}\Gamma\left(\frac{1}{\delta_i}+2\right)\\
&=\mu_i^2s_j^2\Phi_j^2\left(1+\theta\right)\left(1+\delta_i\right).
\end{aligned}
$$

Therefore, we can calculate the variance of the gene expression count of gene i in cell j, in particular,

$$\begin{aligned}
&\mathrm{Var}\left(Y_{ij}|\mu_i,\,\delta_i,\,\Phi_j,\,s_j,\,\theta\right)\\
&=\mathrm{E}\left(Y_{ij}^2|\mu_i,\,\delta_i,\,\Phi_j,\,s_j,\,\theta\right)-\mathrm{E}^2\left(Y_{ij}|\mu_i,\,\delta_i,\,\Phi_j,\,s_j,\,\theta\right)\\
&=\mathrm{E}\left(Y_{ij}\left(Y_{ij}-1\right)|\mu_i,\delta_i,\Phi_j,s_j,\theta\right)+\mathrm{E}\left(Y_{ij}|\mu_i,\delta_i,\Phi_j,s_j,\theta\right)-\mathrm{E}^2\left(Y_{ij}|\mu_i,\,\delta_i,\,\Phi_j,\,s_j,\,\theta\right)\\
&=\mu_i^2 s_j^2\Phi_j^2\left(1+\theta\right)\left(1+\delta_i\right)+\mu_i s_j\Phi_j-\mu_i^2 s_j^2\Phi_j^2\\
&=\mu_i s_j\Phi_j+\theta\left(\mu_i s_j\Phi_j\right)^2+\delta_i\left(\theta+1\right)\left(\mu_i s_j\Phi_j\right)^2.
\end{aligned}$$

On the right side of the above equation, only the third addend, containing δ_i, is related to the gene specific cell-to-cell heterogeneous expression. BASiCS denotes the proportion of expression variance caused by heterogeneous expression of biological gene i in cell j as ψ_{ij}:

$$\begin{aligned}
\psi_{ij}&\equiv\frac{\delta_i(\theta+1)(\mu_i s_j\Phi_j)^2}{\mu_i s_j\Phi_j+\theta(\mu_i s_j\Phi_j)^2+\delta_i(\theta+1)(\mu_i s_j\Phi_j)^2}\\
&=\frac{\delta_i(\theta+1)}{(\mu_i s_j\Phi_j)^{-1}+\theta+\delta_i(\theta+1)}.
\end{aligned}\tag{3.20}$$

From the posterior distributions, BASiCS estimates ψ_i, the proportion of expression variance caused by heterogeneous expression of gene i across all cells:

$$\psi_i\approx\frac{\delta_i(\theta+1)}{\left[\mu_i(s\Phi)^*\right]^{-1}+\theta+\delta_i(\theta+1)},\tag{3.21}$$

where

$$(s\Phi)^*\equiv\underset{j=1,\ldots,n}{\mathrm{median}}\{s_j\Phi_j\}.\tag{3.22}$$

BASiCS generates the distribution of ψ_i according to the posterior distribution of δ_i, μ_i, Φ_j, s_j and θ. BASiCS labels genes as Highly Variable Genes (HVGs) if

$$\pi_i^H(\gamma_H)\equiv P\left(\psi_i>\gamma_H\right)>\alpha_H,\tag{3.23}$$

and labels genes as Lowly Variable Genes (LVGs) if

$$\pi_i^L(\gamma_L)\equiv P\left(\psi_i<\gamma_L\right)>\alpha_L,\tag{3.24}$$

where the given variance contribution threshold γ_H and γ_L could be fixed prior to the analysis, and the given evidence threshold α_H and α_L could be optimised by letting

$$\begin{cases} \text{EFDR}_{\alpha_H} = \text{EFNR}_{\alpha_H}, \\ \text{EFDR}_{\alpha_L} = \text{EFNR}_{\alpha_L}. \end{cases} \tag{3.25}$$

Here the Expected False Discovery Rate (EFDR), i.e. the possibility that BASiCS does not label an HVG (LVG) as HVG (LVG), is defined as

$$\begin{cases} \text{EFDR}_{\alpha_H} \equiv \frac{\sum_{i=1}^{q_0}[1-\pi_i^H(\gamma_H)]\mathbb{1}\{\pi_i^H(\gamma_H)>\alpha_H\}}{\sum_{i=1}^{q_0}\mathbb{1}\{\pi_i^H(\gamma_H)>\alpha_H\}}, \\ \text{EFDR}_{\alpha_L} \equiv \frac{\sum_{i=1}^{q_0}[1-\pi_i^L(\gamma_L)]\mathbb{1}\{\pi_i^L(\gamma_L)>\alpha_L\}}{\sum_{i=1}^{q_0}\mathbb{1}\{\pi_i^L(\gamma_L)>\alpha_L\}}, \end{cases} \tag{3.26}$$

where $\mathbb{1}(x \in A)$ is the indicator function, $\mathbb{1}(x \in A) = 1$ if $x \in A$ and 0 otherwise.

The Expected False Negative Rate (EFNR), i.e. the possibility that BASiCS labels a non-HVG (non-LVG) as HVG (LVG) is defined as

$$\begin{cases} \text{EFNR}_{\alpha_H} \equiv \frac{\sum_{i=1}^{q_0}\pi_i^H(\gamma_H)\mathbb{1}\{\pi_i^H(\gamma_H)\leq\alpha_H\}}{\sum_{i=1}^{q_0}\mathbb{1}\{\pi_i^H(\gamma_H)\leq\alpha_H\}} \\ \text{EFNR}_{\alpha_L} \equiv \frac{\sum_{i=1}^{q_0}\pi_i^L(\gamma_L)\mathbb{1}\{\pi_i^L(\gamma_L)\leq\alpha_L\}}{\sum_{i=1}^{q_0}\mathbb{1}\{\pi_i^L(\gamma_L)\leq\alpha_L\}}. \end{cases} \tag{3.27}$$

3.1.3 Posterior predictive check

Rubin (1984) described the procedure of Posterior predictive check (PPC). In Gelman *et al.* (1996), the term "Posterior Predictive Check" is introduced to describe such a method to assess the fitness of a model, especially for Bayesian models. The idea behind PPC is that if a model fits the observed data well, then the posterior predictive data should be representative of the observed data. Here we briefly introduce the general setting and process of PPC.

Consider a joint distribution over measurements $\boldsymbol{y}$ and parameters $\boldsymbol{\zeta}$, with specified likelihood $\pi(\boldsymbol{y}|\boldsymbol{\zeta})$ and prior distribution $\pi(\boldsymbol{\zeta})$. Bayes' Theorem yields that for a set of observations $\tilde{\boldsymbol{y}}$, the posterior distribution $\pi(\boldsymbol{\zeta}|\tilde{\boldsymbol{y}}) \propto \pi(\tilde{\boldsymbol{y}},\boldsymbol{\zeta}) = \pi(\tilde{\boldsymbol{y}}|\boldsymbol{\zeta})\cdot\pi(\boldsymbol{\zeta})$. Denote the corresponding parameter space of $\boldsymbol{\zeta}$ as $\mathcal{Z}$, the posterior predictive distribution inferred with $\tilde{\boldsymbol{y}}$, $\pi(\boldsymbol{y}|\tilde{\boldsymbol{y}})$, is calculated by marginalising the distribution of $\boldsymbol{y}$ given $\boldsymbol{\zeta}$ over the parameter space $\mathcal{Z}$

$$\pi(\boldsymbol{y}|\tilde{\boldsymbol{y}}) = \int_{\mathcal{Z}} \pi(\boldsymbol{y}|\boldsymbol{\zeta})\,\pi(\boldsymbol{\zeta}|\tilde{\boldsymbol{y}})\,\mathrm{d}\boldsymbol{\zeta}.$$

Consider drawing a sequence of samples from the posterior distribution $\pi\left(\boldsymbol{\zeta}|\tilde{\boldsymbol{y}}\right) \propto \pi(\boldsymbol{\zeta})\pi\left(\tilde{\boldsymbol{y}}|\boldsymbol{\zeta}\right)$:

$$\boldsymbol{\zeta}(1), \ldots, \boldsymbol{\zeta}(L) \sim \pi\left(\boldsymbol{\zeta}|\tilde{\boldsymbol{y}}\right),$$

then for any given $l = 1, \ldots, L$, we draw one sample $\boldsymbol{y}(l)$ from each corresponding predictive distribution $\pi\left(\boldsymbol{y}|\boldsymbol{\zeta}(l)\right)$. The observed data $\tilde{\boldsymbol{y}}$ can be compared with the sample $\left\{\boldsymbol{y}(1), \ldots, \boldsymbol{y}(L)\right\}$. In some literature, the observed data $\tilde{\boldsymbol{y}}$ and sample $\left\{\boldsymbol{y}(1), \ldots, \boldsymbol{y}(L)\right\}$ are compared using different summary statistics, such as the maximum absolute values (Gelman *et al.*, 1996), maximum values, minimum values and mean (Gelman *et al.*, 2013). However, we decide to compare the value of the observed data and the posterior predictive values directly without using any statistics to transform the data. This is because the BASiCS model has a complicated structure with only one set of data matrix for posterior inference; that is, for each gene i in cell j, the observation Y_{ij} is unique. Therefore, taking any summary statistics from a unique Y_{ij} for comparison would not be possible. The PPC results on both non-regression and regression BASiCS are presented in Section 3.2.3. Essentially, PPC is designed to validate the model assumptions (Talts *et al.*, 2018). In order to assess the correctness of the computational aspect of the BASiCS inference algorithm, we introduce another method in the next section.

3.1.4 Simulation based calibration

Simulation based calibration (SBC) is a general procedure proposed in Talts *et al.* (2018) for validating inferences from Bayesian algorithms capable of generating posterior samples. Consider a joint distribution over measurements $\boldsymbol{y}$ and parameters $\boldsymbol{\zeta}$, with specified likelihood $\pi(\boldsymbol{y}|\boldsymbol{\zeta})$ and prior distribution $\pi(\boldsymbol{\zeta})$, so that

$$\pi(\boldsymbol{y}, \boldsymbol{\zeta}) = \pi(\boldsymbol{y}|\boldsymbol{\zeta}) \cdot \pi(\boldsymbol{\zeta}).$$

Bayes' Theorem yields that for a set of observations $\tilde{\boldsymbol{y}}$, the posterior distribution $\pi(\boldsymbol{\zeta}|\tilde{\boldsymbol{y}}) \propto \pi(\tilde{\boldsymbol{y}}, \boldsymbol{\zeta})$. Denote the corresponding parameter space of $\boldsymbol{\zeta}$ as $\mathcal{Z}$. Suppose we simulate a ground truth $\tilde{\boldsymbol{\zeta}} \in \mathcal{Z}$ from the prior

$$\tilde{\boldsymbol{\zeta}} \sim \pi(\boldsymbol{\zeta}),$$

and then we generate data from the corresponding data generating process

$$\tilde{\boldsymbol{y}} \sim \pi\left(\boldsymbol{y}|\tilde{\boldsymbol{\zeta}}\right).$$

It is clear that, by integrating the exact posteriors over the Bayesian joint distribution, one gets the prior distribution

$$\pi(\boldsymbol{\zeta}) = \int \pi\left(\boldsymbol{\zeta}|\tilde{\boldsymbol{y}}\right)\pi\left(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{\zeta}}\right)\pi\left(\tilde{\boldsymbol{\zeta}}\right)\mathrm{d}\tilde{\boldsymbol{y}}\mathrm{d}\tilde{\boldsymbol{\zeta}}. \tag{3.28}$$

Equation (3.28) is called the self consistency condition in Talts *et al.* (2018).

Consider drawing a sequence of samples from the posterior distribution $\pi(\boldsymbol{\zeta}|\tilde{\boldsymbol{y}}) \propto \pi(\boldsymbol{\zeta})\pi(\tilde{\boldsymbol{y}}|\boldsymbol{\zeta})$:

$$\{\boldsymbol{\zeta}(1),\ldots,\boldsymbol{\zeta}(L)\} \sim \pi(\boldsymbol{\zeta}|\tilde{\boldsymbol{y}}).$$

Condition (3.28) implies that $\tilde{\boldsymbol{\zeta}}$ and $\{\boldsymbol{\zeta}(1),\ldots,\boldsymbol{\zeta}(L)\}$ will be distributed according to the same distribution. Based on this, Talts *et al.* (2018) defined the following rank statistic:

Definition 3.2 Let $\tilde{\boldsymbol{\zeta}} \sim \pi(\boldsymbol{\zeta})$, $\tilde{\boldsymbol{y}} \sim \pi(\boldsymbol{y}|\tilde{\boldsymbol{\zeta}})$, $\{\boldsymbol{\zeta}(1),\ldots,\boldsymbol{\zeta}(L)\} \sim \pi(\boldsymbol{\zeta}|\tilde{\boldsymbol{y}})$ for any joint distribution $\pi(\boldsymbol{y},\boldsymbol{\zeta})$. Consider any one-dimensional random variable $c : \mathcal{Z} \mapsto \mathbb{R}$, the rank statistic of the prior sample $\tilde{\boldsymbol{\zeta}}$ relative to the posterior sample $\{\boldsymbol{\zeta}(1),\ldots,\boldsymbol{\zeta}(L)\}$ with respect to c is defined as:

$$r\left(\{c\left(\boldsymbol{\zeta}(1)\right),\ldots,c\left(\boldsymbol{\zeta}(L)\right)\},c\left(\tilde{\boldsymbol{\zeta}}\right)\right) = \sum_{l=1}^{L} \mathbb{1}_{\{\boldsymbol{\zeta}(l):c(\boldsymbol{\zeta}(l))<c(\tilde{\boldsymbol{\zeta}})\}}\left[\boldsymbol{\zeta}(l)\right], \tag{3.29}$$

where for a set A,

$$\mathbb{1}_A(a) = \begin{cases} 1, \text{ if } a \in A, \\ 0, \text{ else.} \end{cases}$$

Talts *et al.* (2018) then proved the following theorem:

Theorem 3.1 Given an i.i.d. sample $\{\boldsymbol{\zeta}(1),\ldots,\boldsymbol{\zeta}(L)\}$ from the posterior and any $c : \mathcal{Z} \mapsto \mathbb{R}$, the rank statistic in (3.29) over $\tilde{\boldsymbol{\zeta}} \sim \pi(\boldsymbol{\zeta})$ follows a discrete uniform distribution over $\{0,1,\ldots,L\}$.

Based on the uniformity of the rank statistic, Talts *et al.* (2018) introduced Simulation-based Calibration as a way of exploiting this result to validate the inference process

in practice, by checking that the resulting rank statistic is uniformly distributed. Algorithm 2 is proposed by Talts *et al.* (2018) to obtain a sample of rank statistic.

Algorithm 2 Simulation Based Calibration (SBC)

Require: Data generating model $\pi(\boldsymbol{y}|\boldsymbol{\zeta})$, prior distribution $\pi(\boldsymbol{\zeta})$, function $c : \mathcal{Z} \mapsto \mathbb{R}$, the number of samples, K, for rank statistic; the number of posterior samples, L, used for calculating each rank statistic.

Initialise

for k in $(1 : K)$ **do**

Draw prior sample $\tilde{\boldsymbol{\zeta}}^{(k)} \sim \pi(\boldsymbol{\zeta})$.

Draw a simulated dataset $\tilde{\boldsymbol{y}}^{(k)} \sim \pi\left(\boldsymbol{y}|\tilde{\boldsymbol{\zeta}}^{(k)}\right)$.

Use the Bayesian Inference method of your choice, generate posterior sample $\pi\left(\boldsymbol{\zeta}|\tilde{\boldsymbol{y}}^{(k)}\right)$.

Draw L posterior samples $\left\{\boldsymbol{\zeta}^{(k)}(1), \ldots, \boldsymbol{\zeta}^{(k)}(L)\right\} \sim \pi\left(\boldsymbol{\zeta}|\tilde{\boldsymbol{y}}^{(k)}\right)$.

Compute rank statistic

$$\begin{aligned} r^{(k)} &= r\left[\left\{c\left(\boldsymbol{\zeta}^{(k)}(1)\right), \ldots, c\left(\boldsymbol{\zeta}^{(k)}(L)\right)\right\}, c(\tilde{\boldsymbol{\zeta}}^{(k)})\right] \\ &= \sum_{l=1}^{L} \mathbb{1}_{\left\{\boldsymbol{\zeta}^{(k)}(l) : c\left(\boldsymbol{\zeta}^{(k)}(l)\right) < c\left(\tilde{\boldsymbol{\zeta}}^{(k)}\right)\right\}}\left[\boldsymbol{\zeta}^{(k)}(l)\right]. \end{aligned}$$

end for

Plot the histogram of rank statistic $r^{(k)}$, for $k = 1, \ldots, K$.

Check the uniformity of the histogram of $r^{(k)}$, for $k = 1, \ldots, K$.

We note that Talts *et al.* (2018) recommends visual inspection for SBC results. In a third-party R package `BayesianTools` (Hartig *et al.*, 2019), which includes SBC implementation, Kolmogorov-Smirnov test (Kolmogorov, 1933; Massey Jr, 1951; Smirnov, 1939) is also used to assess the uniformality of SBC rank statistic. In short, Kolmogorov-Smirnov test is based on examining the maximum distance between the CDF curves of the reference distribution and the empirical distribution. In the experiments presented in this chapter, we plot the empirical cumulative density function (ECDF) of the rank statistic and CDFs of samples from Unif$(\{0, 1, \ldots, L\})$ for comparison.

It is noted by Talts *et al.* (2018) that when the Bayesian Inference method applied is MCMC, Algorithm 2 would introduce bias, due to the autocorrelation structure

in the MCMC chain. To mitigate this issue, Talts *et al.* (2018) proposes that for MCMC methods, one can add a step for assessing Effective Sample Size $N_{eff}[c]$ with respect to the measurement of interest $c(\boldsymbol{\zeta})$. If $N_{eff}[c] > L$ then the autocorrelation is negligible, otherwise the MCMC needs to be rerun for an appropriate length of iterations.

Suppose the original posterior MCMC chain length is N_{sample}, the effective MCMC chain length $N_{eff}[c]$ is calculated by

$$N_{eff}[c] = \frac{N_{sample}}{1 + 2\sum_{\tau=1}^{+\infty} R_\tau[c]}, \tag{3.30}$$

where $R_\tau[c]$ is the lag-τ autocorrelation of c:

$$R_\tau[c] = E_t\left[c(\boldsymbol{\zeta}(t))c(\boldsymbol{\zeta}(t+\tau))\right]. \tag{3.31}$$

The lag-τ autocorrelation of c can be estimated numerically from the posterior MCMC chain of length N_{sample}:

$$\begin{aligned} R_\tau^{(k)}[c] = &\frac{1}{(N_{sample}-\tau)\cdot \mathrm{Var}\left[c\left(\boldsymbol{\zeta}^{(k)}\right)\right]} \sum_{t=1}^{N_{sample}-\tau} \left[c\left(\boldsymbol{\zeta}^{(k)}(t)\right) - \overline{c\left(\boldsymbol{\zeta}^{(k)}\right)}\right] \\ &\cdot \left[c\left(\boldsymbol{\zeta}^{(k)}(t+\tau)\right) - \overline{c\left(\boldsymbol{\zeta}^{(k)}\right)}\right], \end{aligned} \tag{3.32}$$

where $\mathrm{Var}\left[c\left(\boldsymbol{\zeta}^{(k)}\right)\right]$ denotes the sample variance of samples $c\left(\boldsymbol{\zeta}^{(k)}(1)\right),\ldots,c\left(\boldsymbol{\zeta}^{(k)}\left(N_{sample}\right)\right)$, and $\overline{c\left(\boldsymbol{\zeta}^{(k)}\right)}$ denotes the sample mean of samples $c\left(\boldsymbol{\zeta}^{(k)}(1)\right),\ldots,c\left(\boldsymbol{\zeta}^{(k)}\left(N_{sample}\right)\right)$. Talts *et al.* (2018) proposed the extended algorithm, Algorithm 3 for validation of Bayesian models inferred by MCMC.

Algorithm 3 Extended SBC for MCMC method

Require: Data generating model $\pi(\boldsymbol{y}|\boldsymbol{\zeta})$, prior distribution $\pi(\boldsymbol{\zeta})$, function $c : \mathcal{Z} \mapsto \mathbb{R}$, the number of samples, K, for rank statistic; the number of posterior samples, L, used for calculating each rank statistic; the number of MCMC iterations, L'; the resulted posterior MCMC chain length, N_{sample}.

Initialise

while k in $(1:K)$ **do**

Draw prior sample $\tilde{\boldsymbol{\zeta}}^{(k)} \sim \pi(\zeta)$.

Draw a simulated data set $\tilde{\boldsymbol{y}}^{(k)} \sim \pi(\boldsymbol{y}|\tilde{\boldsymbol{\zeta}}^{(k)})$.

Run a MCMC for L' iterations to generate the correlated posterior sample chain of length N_{sample} from $\pi(\boldsymbol{\zeta}^{(k)}|\tilde{\boldsymbol{y}}^{(k)})$.

Compute the effective lag-τ autocorrelation for c, when $\tau = 0, \ldots, N_{sample} - 1$:

$$R_{\tau}^{(k)}[c] = \frac{1}{(N_{sample} - \tau)\mathrm{Var}\left(c\left(\boldsymbol{\zeta}^{(k)}\right)\right)} \sum_{t=1}^{N_{sample}-\tau} \left[c\left(\boldsymbol{\zeta}_t^{(k)}\right) - \overline{c\left(\boldsymbol{\zeta}^{(k)}\right)}\right] \cdot \left[c\left(\boldsymbol{\zeta}_{t+\tau}^{(k)}\right) - \overline{c\left(\boldsymbol{\zeta}^{(k)}\right)}\right]. \tag{3.33}$$

Compute the effective sample size

$$N_{eff}^{(k)}[c] = \frac{N_{sample}}{1 + 2\sum_{\tau=1}^{+\infty} R_{\tau}[c]}. \tag{3.34}$$

if $N_{eff} < L$ **then**

Rerun the MCMC for $\frac{L' \cdot L}{N_{eff}[c]}$ iterations.

else

Thin the posterior MCMC chain to L samples $\left\{\boldsymbol{\zeta}^{(k)}(1), \ldots, \boldsymbol{\zeta}^{(k)}(L)\right\}$, and truncate any leftover sample from the k-th run after $\boldsymbol{\zeta}^{(k)}(L)$.

end if

Compute rank statistic

$$r^{(k)} = r\left(\left\{c\left(\boldsymbol{\zeta}^{(k)}(1)\right), \ldots, c\left(\boldsymbol{\zeta}^{(k)}(L)\right)\right\}, c(\tilde{\boldsymbol{\zeta}}^{(k)})\right) \tag{3.35}$$

$$= \sum_{l=1}^{L} \mathbb{1}_{\left\{\boldsymbol{\zeta}^{(k)}(l) : c\left(\boldsymbol{\zeta}^{(k)}(l)\right) < c\left(\tilde{\boldsymbol{\zeta}}^{(k)}\right)\right\}}\left(\boldsymbol{\zeta}^{(k)}(l)\right). \tag{3.36}$$

end while

Plot the histogram of rank statistic $r^{(k)}$, for $k = 1, \ldots, K$.

Check the uniformity of the histogram of $r^{(k)}$, for $k = 1, \ldots, K$.

3.2 Evaluation of a Bayesian Hierarchical Model

In this section, we examine the reliability of the BASiCS (Eling *et al.*, 2018; Vallejos *et al.*, 2015, 2016), as an example of a Bayesian Hierarchical model for scRNAseq data. It is worth to mention that some experiments are already carried out in Eling *et al.* (2018); Vallejos *et al.* (2015, 2016) to check the reliability of BASiCS. Vallejos *et al.* (2016) experiments on synthetic datasets generated with varying cell-specific parameters Φ_j, s_j and global parameter θ, confirming that different values of Φ_j, s_j and θ in data simulation do not affect the posterior inference of gene-specific parameters. Eling *et al.* (2018) carries out some experiments on a real dataset from Zeisel *et al.* (2015), and finds that when the number of cell samples from the real dataset is small, the non-regression BASiCS (Vallejos *et al.*, 2016) tends to underestimate δ_i for lowly expressed genes and to overestimate δ_i for genes with medium and high expression levels, while the regression BASiCS (Eling *et al.*, 2018) produces more robust results regardless of the sample size. Here we examine the reliability of the BASiCS model from other perspectives to get a more comprehensive understanding of it.

For BHMs applied to biological data, it is rare to have the ground truth of the underlying parameters to assess the recovery of parameters of interest. In this section, we simulate the gene expression count matrix from the prior model of non-regression and regression BASiCS (Eling *et al.*, 2018; Vallejos *et al.*, 2015, 2016) respectively, and then we plug in the synthetic data in the corresponding BASiCS MCMC to compare the estimated posteriors with the "true" parameter values used for data generation, using validation procedures such as PPC and SBC as introduced in Subsection 3.1.3 and 3.1.4. A detailed description of the prior models has been discussed in Subsection 3.1.2. Our experiments are performed in `R 4.0.2` (R Core Team, 2013), code available at https://github.com/lilythepooh/BASiCS-Reliability.git.

3.2.1 Uncertainty of the posterior median as a point estimate

In Eling *et al.* (2018); Vallejos *et al.* (2015, 2016), the posterior median is used to estimate the value of $\delta_i, \mu_i, \nu_j, \phi_j, s_j, \theta$ for downstream analysis. Here, we simulate a dataset $\mathbf{X}^{(1)*}$ of 100 genes, 10 spike-in genes, and 50 cells from non-regression BASiCS model (Vallejos *et al.*, 2016), simulating δ_i from a log-Normal distribution as in Equation (3.3).

BASiCS (Eling *et al.*, 2018; Vallejos *et al.*, 2015, 2016) requires data preprocessing, where the genes and cells with too many 0 gene expression counts are filtered out. After the required data-preprocessing procedure, we obtain the filtered dataset of synthetic gene expression with $n = 39$ cells, $q_0 = 100$ biological genes and $q - q_0 = 10$ spike-in genes. Then we plug this synthetic dataset into the MCMC algorithm for non-regression BASiCS (`BASiCS` package, Vallejos *et al.* (2016)), with fixed prior-hyperparameter values as listed in Table 3.2.

hyperparameters	σ_μ^2	σ_δ^2	a_s	b_s	a_θ	b_θ	$\boldsymbol{p}$
prior value	0.8	0.5	1	1	1	1	$(1,\ldots,1)$

Table 3.2: Fixed prior values for hyperparameters in non-regression BASiCS

Similarly, we simulate a dataset $\mathbf{X}^{(2)*}$ of 100 biological genes, 10 spike-in genes and 50 cells from regression BASiCS model (Eling *et al.*, 2018), where δ_i was simulated from a log-Normal distribution as in Equation (3.13). After the required data-preprocessing procedure, we get a synthetic gene expression count dataset with $n = 44$ cells, $q_0 = 98$ biological genes and $q - q_0 = 10$ spike-in genes. Then we plug the fixed synthetic dataset back into the MCMC algorithm for regression BASiCS within the `BASiCS` package (Eling *et al.*, 2018), with fixed prior-hyperparameter values as listed in Table 3.3.

hyperparameters	σ_μ^2	σ_δ^2	a_s	b_s	a_θ	b_θ	$\boldsymbol{p}$	a_σ	b_σ
prior value	0.8	0.5	1	1	1	1	$(1,\ldots,1)$	2	2

Table 3.3: Fixed prior values for hyperparameters in regression BASiCS

When recovering the parameter values used for generating datasets $\mathbf{X}^{(1)*}$ and $\mathbf{X}^{(2)*}$, we replicate 100 MCMCs respectively, resulting in 100 posterior medians for each parameter μ_i, δ_i, ν_j, s_j, Φ_j and θ from each model for $i = 1,\ldots,q_0$, $j = 1,\ldots,n$. Each of the 100 MCMCs was run for 15,000 iterations, 10,000 burns and thinned by 5, resulting in 100 posterior samples of size 1,000 for both models respectively.

To illustrate the recovery of true parameters in each run, we calculated the 89% Highest Density Credible Interval and 50% Highest Density Credible Interval using `bayestestR` package (Makowski *et al.*, 2019). We calculated the 89% Credible Interval

rather than the common 95% because according to Makowski *et al.* (2019), 89% credible intervals are more stable, in the sense that the standard deviation within a 89% credible interval is smaller than within 95% credible interval. We note that by the time of the writing of this thesis, the 2022 updated version of `R` package `bayestestR` (Makowski *et al.*, 2019) is adopting 95% credible interval as default again, yet we decide to present our experiment results as we obtained in the first instance.

Non-regression BASiCS

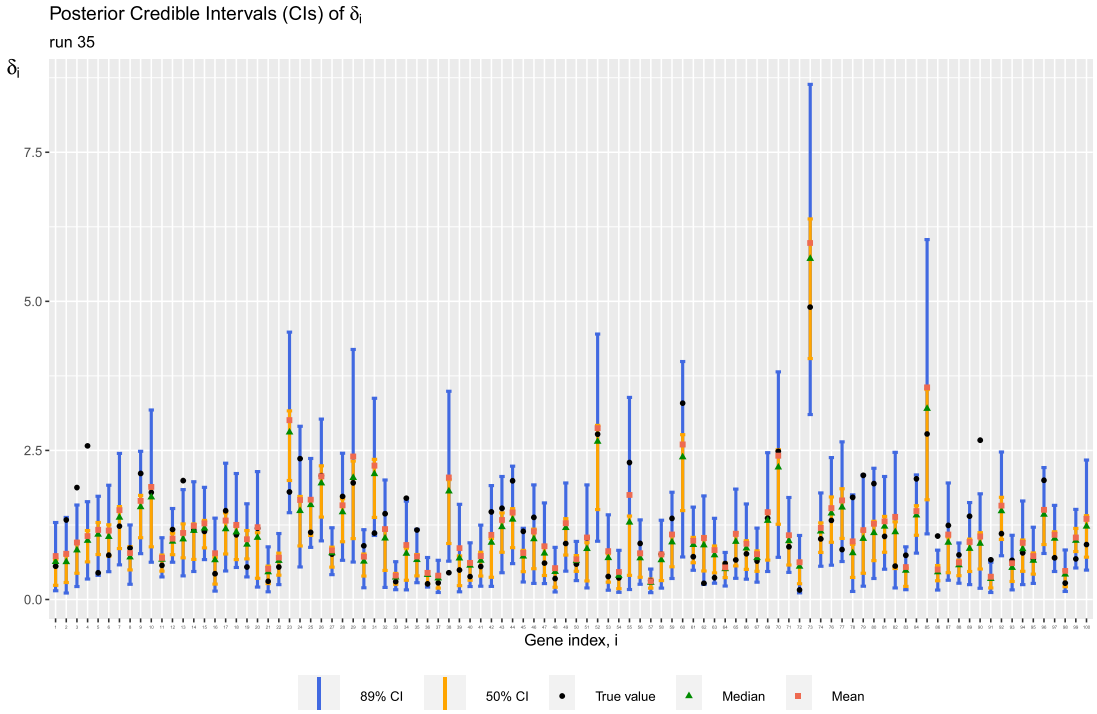

Figure 3.3: True values (δ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for δ_i, for biological genes $i = 1, \ldots, q_0$ ($q_0 = 100$), and for one particular replication (the 35th) of the estimation procedure. Inferred from the non-regression BASiCS model with the fixed dataset $\mathbf{X}^{(1)*}$.

In Figure 3.3, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 100 biological gene-specific variation parameter δ_i, from a single run (replication

number 35) of non-regression BASiCS MCMC on $\mathbf{X}^{(1)*}$. Among 100 biological gene-specific variation parameters δ_i, $i = 1,\ldots,100$, 12 of the true values do not fall into the 89% Highest Density Credible Interval, and 45 of the true values do not fall into the 50% Highest Density Credible Interval. As can be observed, the level of stochasticity in this BHM means that the posterior median as a single estimate of the parameter is not always accurate, since these single estimates may not even properly capture the relative relationship between δ_{i_1} and δ_{i_2}, $i_1 \neq i_2$, , $i_1, i_2 = 1,\ldots,q_0$. For example, δ_{80} and δ_{82} have very similar posterior median and posterior mean (around 1.25), but the true value for gene 82 ($\delta^*_{82} = 0.561$) is much smaller than for gene 80 ($\delta^*_{80} = 1.944$), indicating a lower biological variation factor value for gene 82.

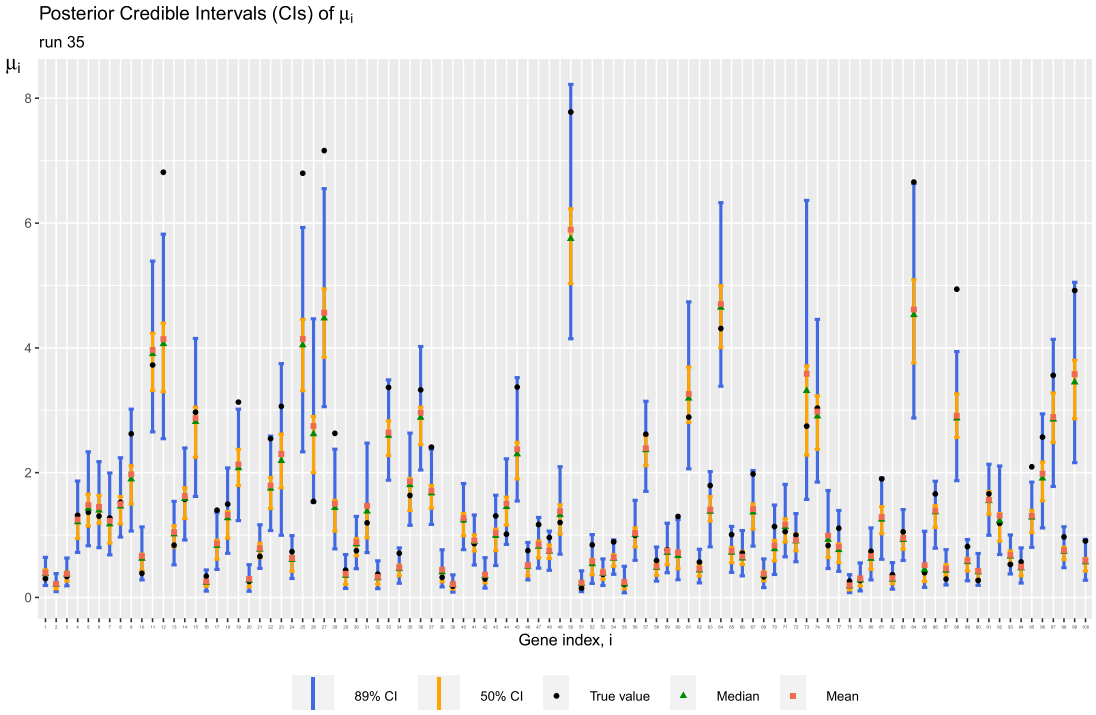

Figure 3.4: True values (μ^*_i) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for μ_i, for biological genes $i = 1,\ldots,q_0$ ($q_0 = 100$), and for one particular replication (the 35th) of the estimation procedure. Inferred from the non-regression BASiCS model with the fixed dataset $\mathbf{X}^{(1)*}$.

In Figure 3.4, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the

100 expected gene-expression level parameter μ_i, from a single run (replication number 35) of non-regression BASiCS MCMC on $\mathbf{X}^{(1)*}$. Among 100 expected gene-expression level parameters μ_i , $i = 1,\ldots,100$ used to simulate $\mathbf{X}^{(1)*}$, 14 of the true values do not fall into the 89% Highest Density Credible Interval, while 55 of the true values do not fall into the 50% Highest Density Credible Interval. Again, it shows that the posterior median does not necessarily reflect even the relative relationship between μ_{i_1} and μ_{i_2}, $i_1 \neq i_2$, $i_1, i_2 = 1,\ldots,q_0$. For example, μ_6 and μ_{60} have very similar true values (around 1.3), but both the posterior mean and posterior median of μ_6 (around 1.5) are much larger than the posterior mean and posterior median of μ_{60} (around 0.8), creating an illusion of higher expected gene expression level for gene 6 compared to gene 60, which is far from the truth.

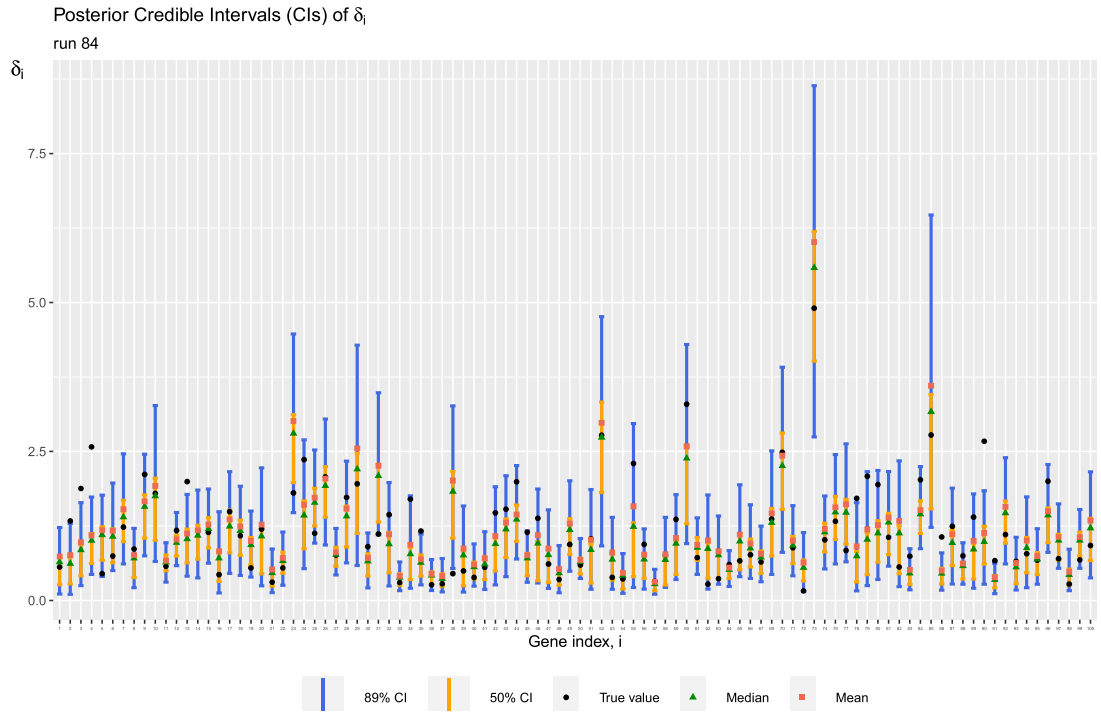

Figure 3.5: True values (δ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for δ_i, for biological genes $i = 1,\ldots,q_0$ ($q_0 = 100$), and for one particular replication (the 84th) of the estimation procedure. Inferred from the non-regression BASiCS model with the fixed dataset $\mathbf{X}^{(1)*}$.

Although our observations correspond to a particular run (35th), they are still

valid when exploring any of the other 99 replications. For example, in Figure 3.5, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 100 biological gene-specific variation parameter δ_i, from a different run (replication number 84) of non-regression BASiCS MCMC on $\mathbf{X}^{(1)*}$. Among 100 biological gene-specific variation parameter δ_i, $i = 1,\ldots,100$, 10 of the true values do not fall into the 89% Highest Density Credible Interval, and 49 of the true values do not fall into the 50% Highest Density Credible Interval. Again, it shows that using the posterior median as an estimation of the true value does not necessarily reflect even the relative relationship between δ_{i_1} and δ_{i_2}, $i_1 \neq i_2$, $i_1, i_2 = 1,\ldots,q_0$. δ_{80} and δ_{82} have very similar posterior median and posterior mean (around 1.25), but the true value of δ_{82} (0.561) is significantly smaller than δ_{80} (1.944), indicating a lower biological variation factor value for gene 82.

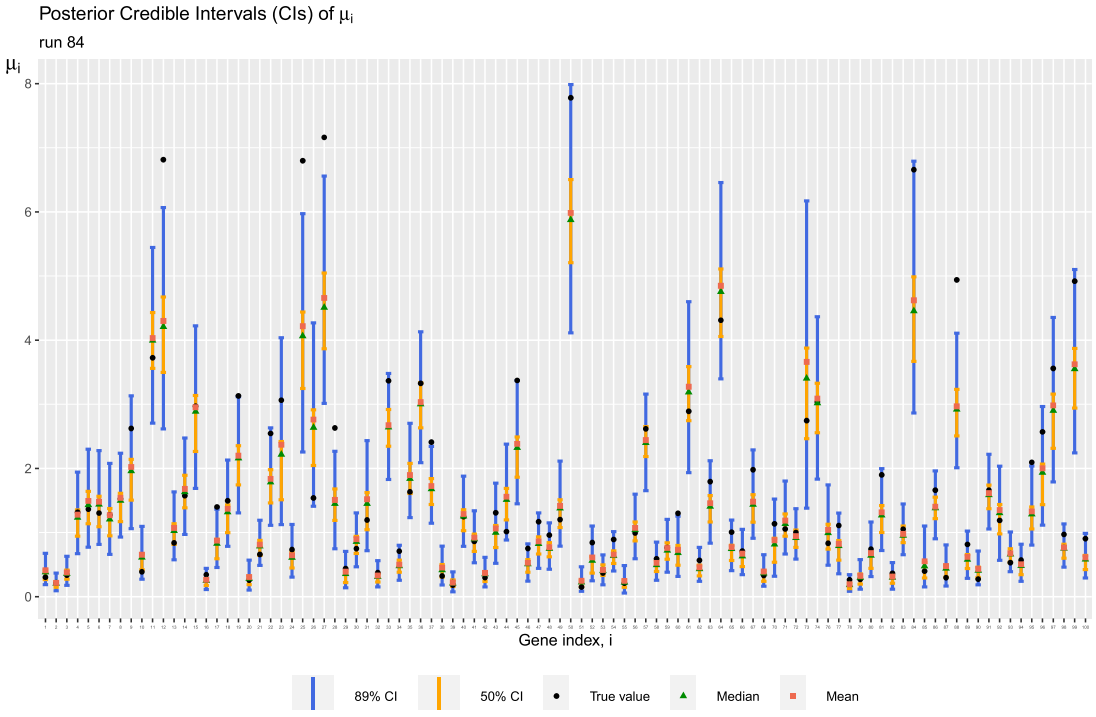

Figure 3.6: True values (μ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for μ_i, for biological genes $i = 1,\ldots,q_0$ ($q_0 = 100$), and for one particular replication (the 84th) of the estimation procedure. Inferred from the non-regression BASiCS model with the fixed dataset $\mathbf{X}^{(1)*}$.

The similarity between the 35th replication and the other 99 replications is also true for the expected gene-expression level parameter μ_i. For example, in Figure 3.6, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 100 biological gene-specific variation parameter μ_i, from a different run (replication number 84) of non-regression BASiCS MCMC on $\mathbf{X}^{(1)*}$. Among 100 expected gene-expression level parameters μ_i , $i = 1,\ldots,100$ used to simulate $\mathbf{X}^{(1)*}$, 19 of the true values do not fall into the 89% Highest Density Credible Interval, and 55 of the true values do not fall into the 50% Highest Density Credible Interval. Again, it shows that the posterior median does not necessarily reflect even the relative relationship between μ_{i_1} and μ_{i_2}, $i_1 \neq i_2$, $i_1, i_2 = 1,\ldots,q_0$. μ_6 and μ_{60} have very similar true values (around 1.3), but both the posterior mean and posterior median of μ_6 (around 1.5) are significantly larger than the posterior mean and posterior median of μ_{60} (around 0.8), creating an illusion of higher expected gene expression level for gene 6 compared to gene 60, which is far from the truth.

To give an overview of the estimation accuracy of posterior medians inferred from the non-regression BASiCS model with $\mathbf{X}^{(1)*}$, we plot all the gene-specific parameters (δ_i and μ_i, $i = 1,\ldots,q_0$) from 200 replication runs with this one fixed dataset $\mathbf{X}^{(1)*}$, as shown in Figure 3.7. Here the x-axis of each point corresponds to the true value of that gene-specific parameter δ_i or μ_i ($i = 1,\ldots,q_0$), which we used to simulate dataset $\mathbf{X}^{(1)*}$. Since all 200 replications are run on the one fixed dataset $\mathbf{X}^{(1)*}$, naturally the plot shows that each true value on x-axis corresponds to 200 posterior medians estimated from 200 replications of posterior inference. If the posterior median as point estimate is accurate, we would expect all the coloured points to fall near the line $y = x$.

Figure 3.7 shows that more posterior medians of μ_i fall inside the 20% relative error range compared to δ_i, but the posterior medians of μ_i vary more compared to δ_i. That is, for a particular gene i, the posterior medians of μ_i from two runs could be more different to the posterior medians of δ_i in the same two runs. The posterior medians of μ_i suffer more from the stochasticity of the MCMC algorithm. Besides, on the one hand, almost all μ_i ($i = 1,\ldots,q_0$) out of the 20% relative error range have large true values and are underestimated. It is possible that the posterior could not recover the true value of μ_i because it is too large. In other words, these

μ_i, $i = 1,\ldots,q_0$ are relatively extreme values in the parameter generating distribution $\mu_i \sim \text{log-Normal}\left(0,\sigma_\mu^2\right)$, and the information given by the generated dataset $\mathbf{X}^{(1)*}$ may not be enough to shift the posterior distribution further from the prior. On the other hand, the δ_i, $i = 1,\ldots,q_0$ out of 20% relative error range seem to be evenly distributed on both sides of the line $y = x$.

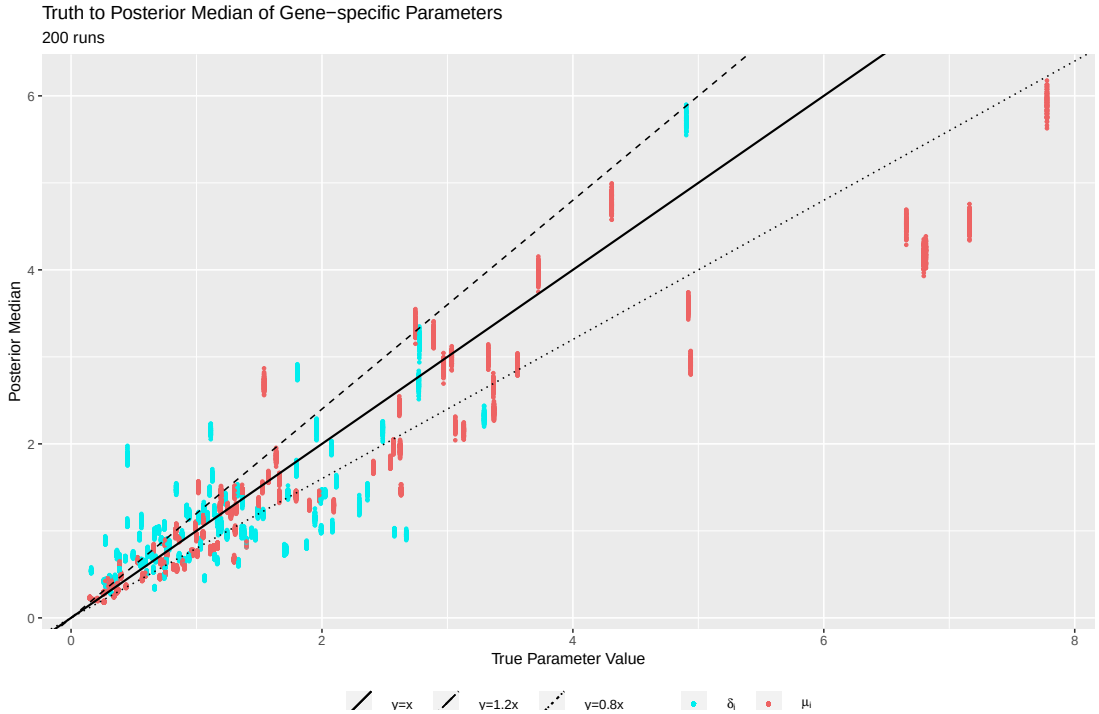

Figure 3.7: True value to posterior median of all gene-specific parameters, 200 replications, inferred from the non-regression BASiCS model with the fixed dataset $\mathbf{X}^{(1)*}$.

Some readers may argue that the above results are not representative as they are only results from repeated experiments on one particular synthetic dataset $\mathbf{X}^{(1)*}$. In order to explore this, we simulate 100 datasets of 100 genes, 10 spike-in genes, and 50 cells from the non-regression BASiCS model (3.1)-(3.10) (Vallejos *et al.*, 2016), simulating δ_i from a log-Normal distribution as in Equation (3.3). After the required data preprocessing, we plug each dataset back into the non-regression BASiCS MCMC for one replication, resulting in 100 posterior samples for δ_i, μ_i, ν_j, Φ_j, s_j, θ, $i = 1,\ldots,q_0$, $j = 1,\ldots,n$. Here we still focus on gene-specific parameters δ_i and μ_i. As examples we focus on the results from two synthetic dataset, the 11th and the 42nd.

Some of the analysis below may seem repetitive, but note that we are only presenting them to demonstrate that the problems exposed in Figure 3.3-3.6 are general rather than a coincidence caused by any particular dataset.

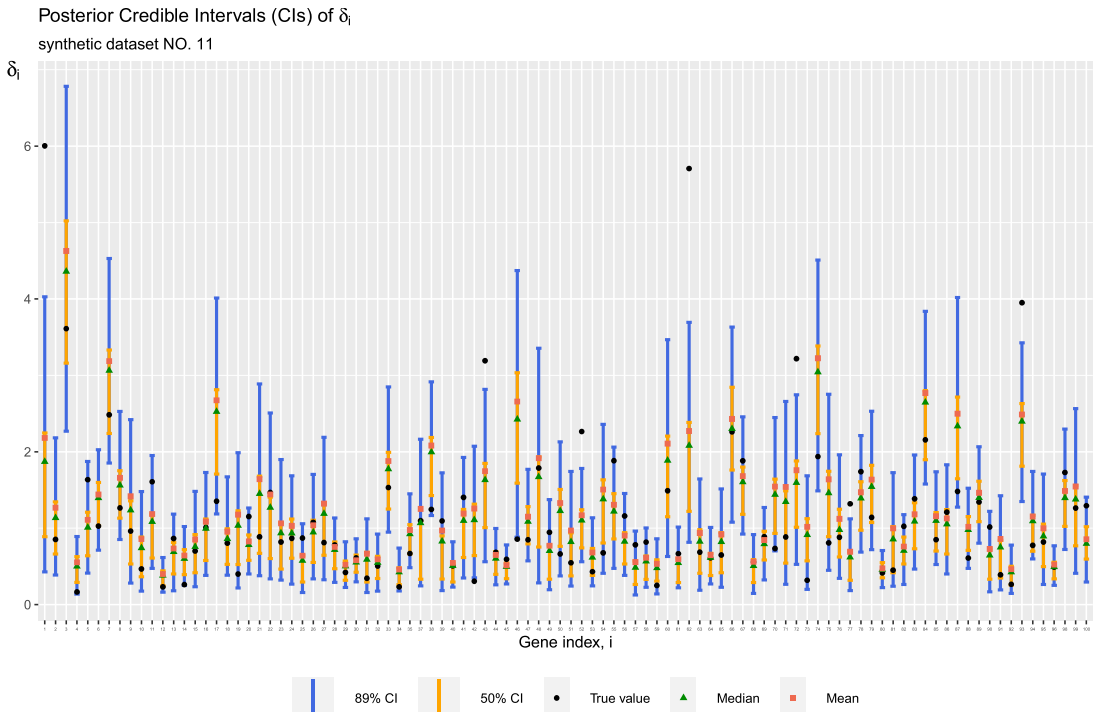

Figure 3.8: True values (δ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for δ_i, for biological genes $i = 1,\ldots,q_0$, and for estimation procedure on the 11th synthetic dataset generated with the non-regression BASiCS model.

In Figure 3.8, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 98 biological gene-specific variation parameter δ_i, from the one run of non-regression BASiCS MCMC on the 11th generated dataset. Among 98 biological gene-specific variation parameters δ_i, 9 of the true values do not fall into the 89% Highest Density Credible Interval, and 48 of the true values do not fall into the 50% Highest Density Credible Interval. Neither the posterior mean nor the posterior median can reflect the relative relationship between δ_{i_1} and δ_{i_2}, for any $i_1 \neq i_2$, $i_1, i_2 = 1,\ldots,q_0$. For example, δ_{46} and δ_{47} have similar true value (around 0.85), but both the posterior

median and the posterior mean of δ_{46} (around 2.6) is much larger than those of δ_{47} (around 1.1).

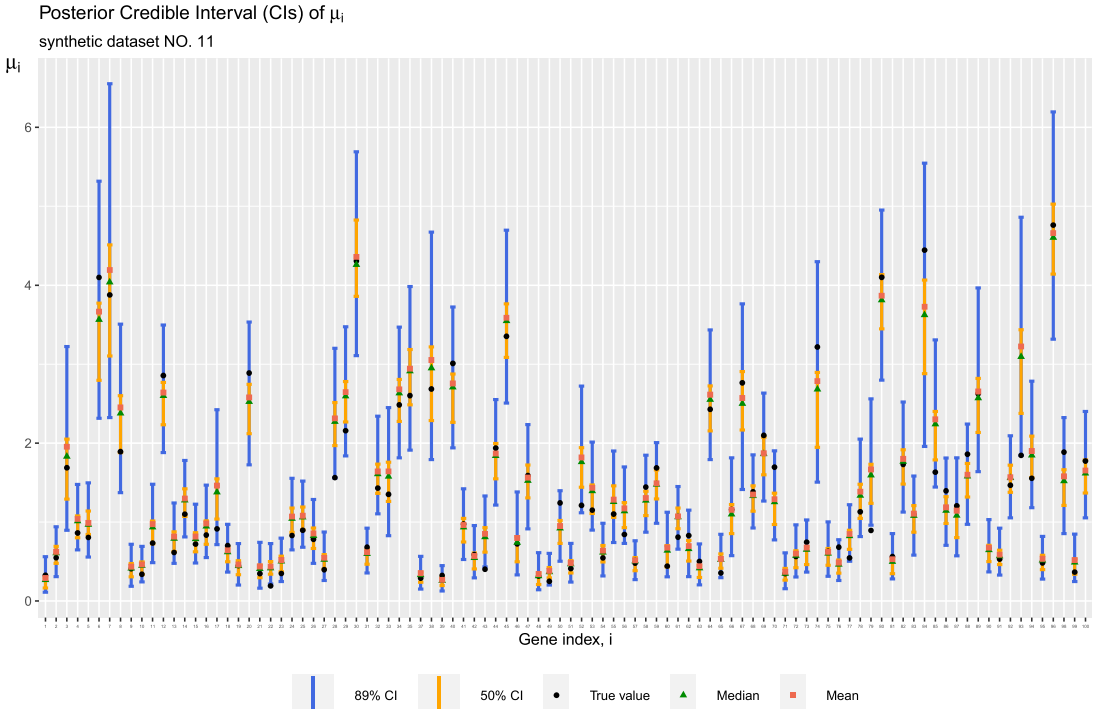

Figure 3.9: True values (μ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for μ_i, for biological genes $i = 1,\ldots,q_0$, and for estimation procedure on the 11th synthetic dataset generated with the non-regression BASiCS model.

In Figure 3.9, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 98 expected gene-expression level parameter μ_i, from the one run of non-regression BASiCS MCMC on the 11th generated dataset. Among 98 expected gene-expression level parameters μ_i, 6 of the true values do not fall into the 89% Highest Density Credible Interval, and 45 of the true values do not fall into the 50% Highest Density Credible Interval. Neither the posterior mean nor the posterior median can reflect the relative relationship between any μ_{i_1} and μ_{i_2}, $i_1 \neq i_2$, $i_1, i_2 = 1,\ldots,q_0$. For example, μ_{88} and μ_{93} have similar true values (around 1.85) , but both the posterior median and the posterior mean of μ_{89} (around 1.7) is much smaller than the posterior median and the posterior mean of μ_{93} (around 3.2).

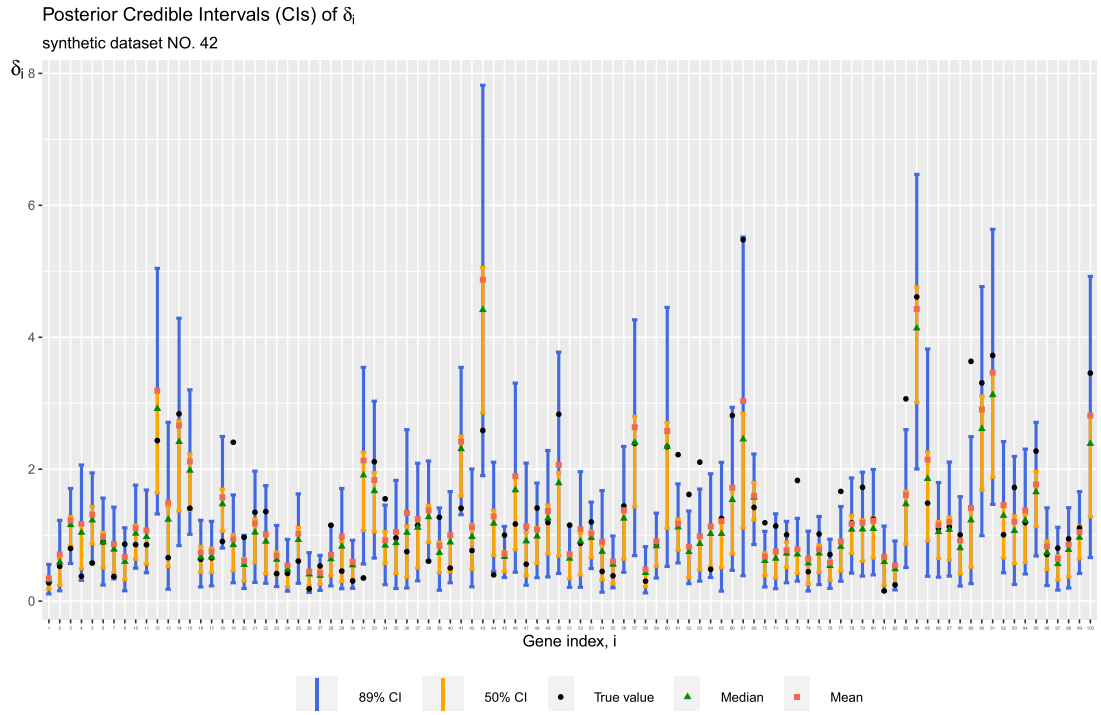

Figure 3.10: True values (δ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for μ_i, for biological genes $i = 1,\ldots,q_0$, and for estimation procedure on the 42nd synthetic dataset generated with the non-regression BASiCS model.

In Figure 3.10, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 97 biological gene-specific variation parameter δ_i, from the one run of the non-regression BASiCS MCMC on the 42nd generated dataset. Among 97 biological gene-specific variation parameters δ_i, 17 of the true values do not fall into the 89% Highest Density Credible Interval, and 57 of the true values do not fall into the 50% Highest Density Credible Interval. Neither the posterior mean nor the posterior median can reflect the relative relationship between any δ_{i_1} and δ_{i_2}, $i_1 \neq i_2$, $i_1, i_2 = 1,\ldots,q_0$. For example, δ_{31} and δ_{33} have similar posterior mean and posterior median level (around 2), but the true value of δ_{31} (0.350) is much smaller than the true value of δ_{33} (2.112).

In Figure 3.11, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 97

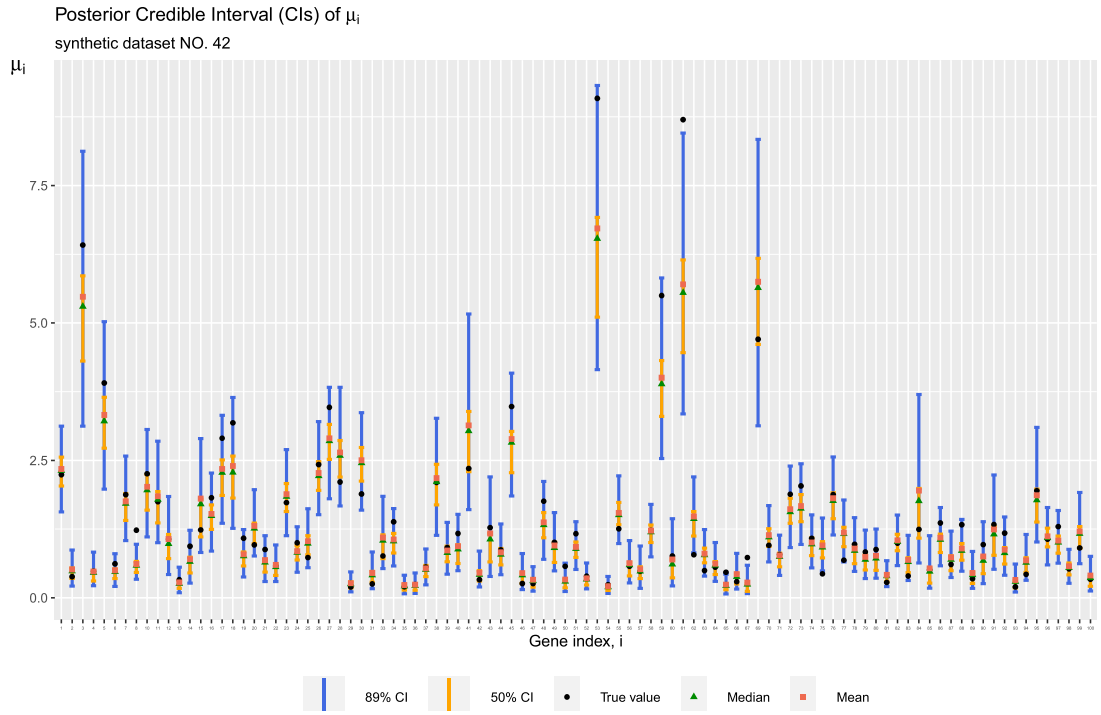

Figure 3.11: True values (μ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for μ_i, for biological genes $i = 1,\ldots,q_0$, and for estimation procedure on the 42nd synthetic dataset generated with the non-regression BASiCS model.

expected gene-expression level parameter μ_i, from the one run of non-regression BASiCS MCMC on the 42nd generated dataset. Among 97 expected gene expression level parameters μ_i, 11 of the true values do not fall into the 89% Highest Density Credible Interval, and 54 of the true values do not fall into the 50% Highest Density Credible Interval. Neither the posterior mean nor the posterior median can reflect the relative relationship between any μ_{i_1} and μ_{i_2}, $i_1 \neq i_2$, $i_1, i_2 = 1,\ldots,q_0$. For example, μ_{61} and μ_{69} have similar posterior median and posterior mean (around 5.4), but the true value of μ_{61} (8.700) is much larger than the true value of μ_{69} (4.704). Figure 3.11 also shows that the larger the true value is, the larger the posterior variance is, indicating that the non-regression BASiCS cannot recover a larger true value precisely.

These results from non-regression BASiCS using different re-generated datasets are consistent with the result from the fixed dataset $\mathbf{X}^{(1)*}$. The recovery of parameters

is not accurate in instances, especially for those larger parameter values. Neither of the point estimates (median or mean) seem to be accurate enough, not being able in some instances even to correctly quantify the relative relationship between two parameter values.

Regression BASiCS

The BASiCS framework has been updated in Eling *et al.* (2018), taking into account the confounding effect between mean and variability. To explore if the resulted regression BASiCS MCMC improves estimation accuracy, we move on to test the regression part of the `BASiCS` package (Eling *et al.*, 2018) with the dataset $\mathbf{X}^{(2)*}$, which is generated with the regression BASiCS model Equation (3.5) - Equation (3.19). As with $\mathbf{X}^{(1)*}$ which we discussed earlier, the one fixed dataset $\mathbf{X}^{(2)*}$ is plugged into regression BASiCS MCMC. We replicate 100 MCMCs, resulting in 100 posterior medians for each parameters $\delta_i, \mu_i, \nu_j, \Phi_j, s_j$ and θ for $i = 1, \ldots, q_0,\ j = 1, \ldots, n$ ($q_0 = 98,\ n = 44$).

In Figure 3.12, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 98 biological gene-specific variation parameter δ_i, from a single run (replication number 29) of regression BASiCS MCMC on $\mathbf{X}^{(2)*}$. Among 98 gene-specific biological variation parameters δ_i, 10 of the true values δ_i^* do not fall inside the estimated 89% Highest Density Credible Interval, and 54 of the true values do not fall inside the 50% Highest Density Credible Interval. We note that those 10 true values outside of the 89% Highest Density Credible Intervals are small values between $(0, 1)$ with very narrow posterior Highest Density Credible Intervals. In this case, the posterior median could still act as a fair single point estimate for them. For most δ_i, the variance of the posteriors looks much smaller compared to Figure 3.3, but such precision only occurs on the posteriors of δ_i with small true values.

In Figure 3.13, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 98 expected gene expression level parameter μ_i, from a single run (replication number 29) of regression BASiCS MCMC on $\mathbf{X}^{(2)*}$. Among 98 expected gene expression level parameter μ_i, 4 of the true values μ_i^* do not fall into the 89% Highest Density Credible

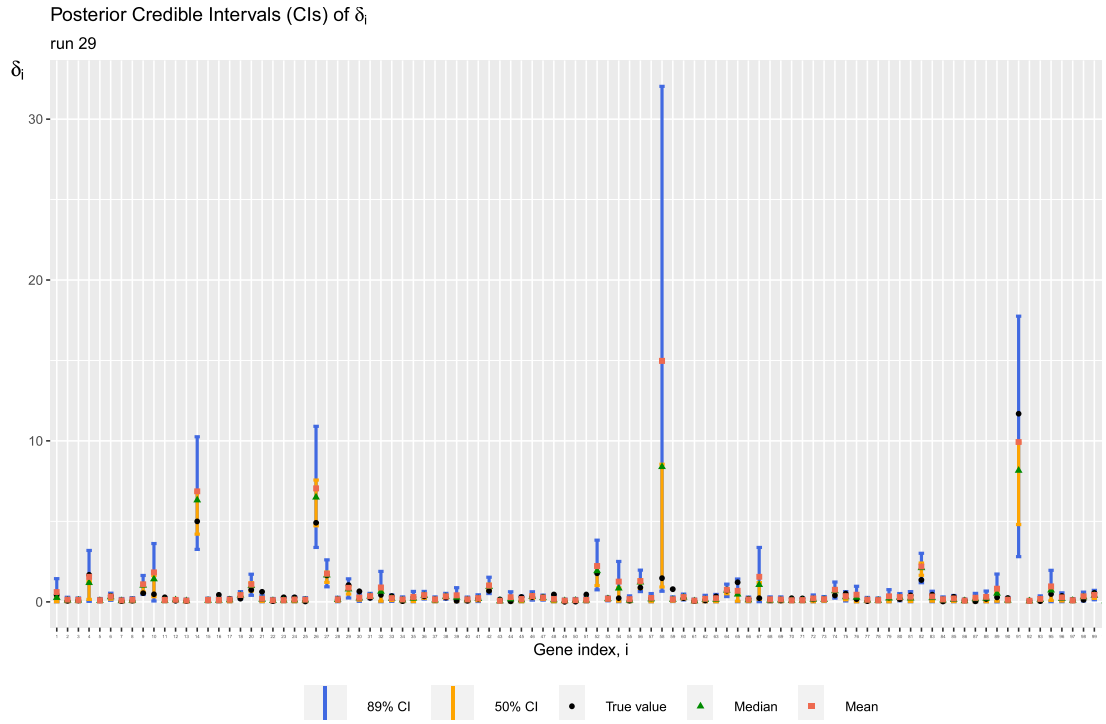

Figure 3.12: True values (δ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for μ_i, for biological genes $i = 1,\ldots,q_0$ ($q_0 = 98$), and for the 29th run of synthetic data generation and estimation procedure. Inferred from the regression BASiCS model with the fixed dataset $\mathbf{X}^{(2)*}$.

Interval, and 55 of the true values do not fall into the 50% Highest Density Credible Interval.

For the 29th estimation procedure on fixed synthetic dataset $\mathbf{X}^{(2)*}$ with regression BASiCS, if we consider the true values of δ_i which fall outside of the corresponding 89% Highest Density Credible Interval, and the true values of μ_i which fall outside of the corresponding 89% Highest Density Credible Interval, we find that they mostly correspond to different genes. In particular, the δ_i^* that do not fall into the 89% Highest Density Credible Interval are δ_{11}^*, δ_{16}^*, δ_{21}^*, δ_{23}^*, δ_{30}^*, δ_{43}^*, δ_{48}^*, δ_{51}^*, δ_{59}^* and δ_{70}^*, while the μ_i^* that do not fall into the 89% Highest Density Credible Interval are μ_{70}^*, μ_{85}^*, μ_{91}^*, μ_{95}^* and μ_{99}^*. Here all the true values of μ_i which are out of the corresponding 89% Highest Density Credible Interval are small values between $(0,2)$. They are outside of the 89% Credible Interval because the posteriors has smaller

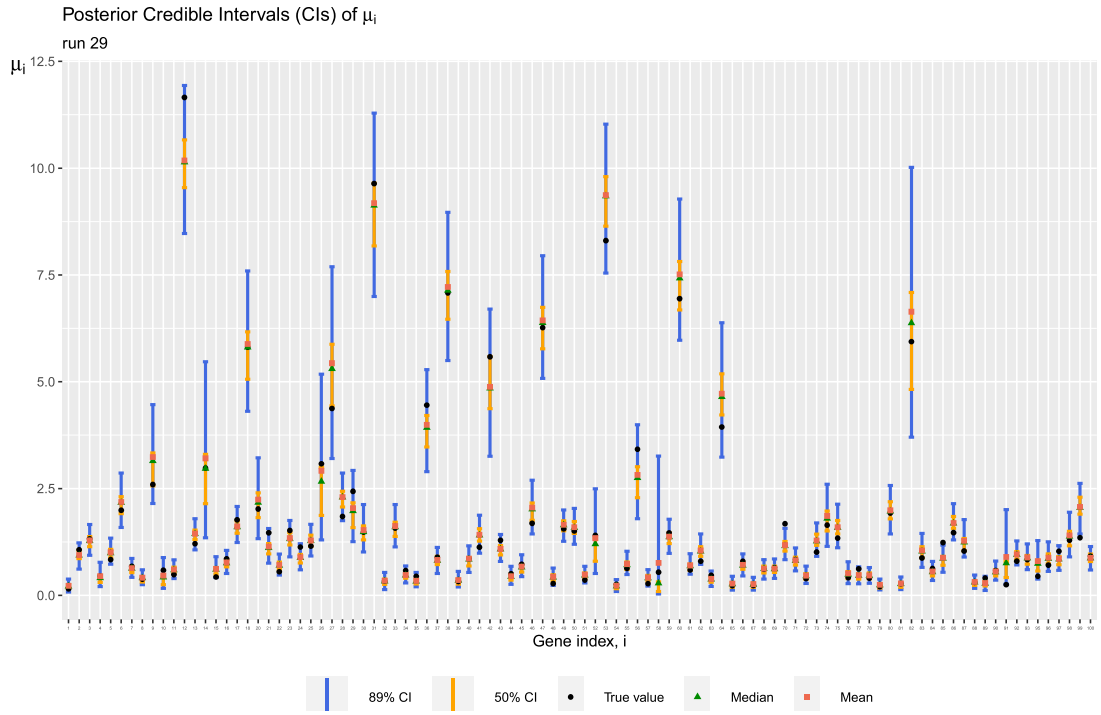

Figure 3.13: True values (μ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for μ_i, for biological genes $i = 1, \ldots, q_0$ ($q_0 = 98$), and for the 29th run of synthetic data generation and estimation procedure. Inferred from the regression BASiCS model with the fixed dataset $\mathbf{X}^{(2)*}$.

variance.

Although our observations correspond to a particular run (29th), they are still valid when exploring any of the other 99 replications. For example, in Figure 3.14, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 98 biological gene-specific variation parameter δ_i, from a different run (replication number 57) of regression BASiCS MCMC on $\mathbf{X}^{(2)*}$. Among 98 biological gene-specific variation parameter δ_i, 10 of the true values δ_i^* do not fall into the 89% Highest Density Credible Interval, 56 of the true value do not fall into the 50% Highest Density Credible Interval. We note that those 10 true values outside of the 89% Highest Density Credible Intervals are small values between $(0, 1)$ with very narrow posterior Highest Density Credible Intervals. In this case, the posterior median can still be considered as a fairly accurate point estimate. For most δ_i, the variance of the posteriors looks

much smaller compared to Figure 3.3, but such precision only occurred on the posteriors of δ_i with small true values.

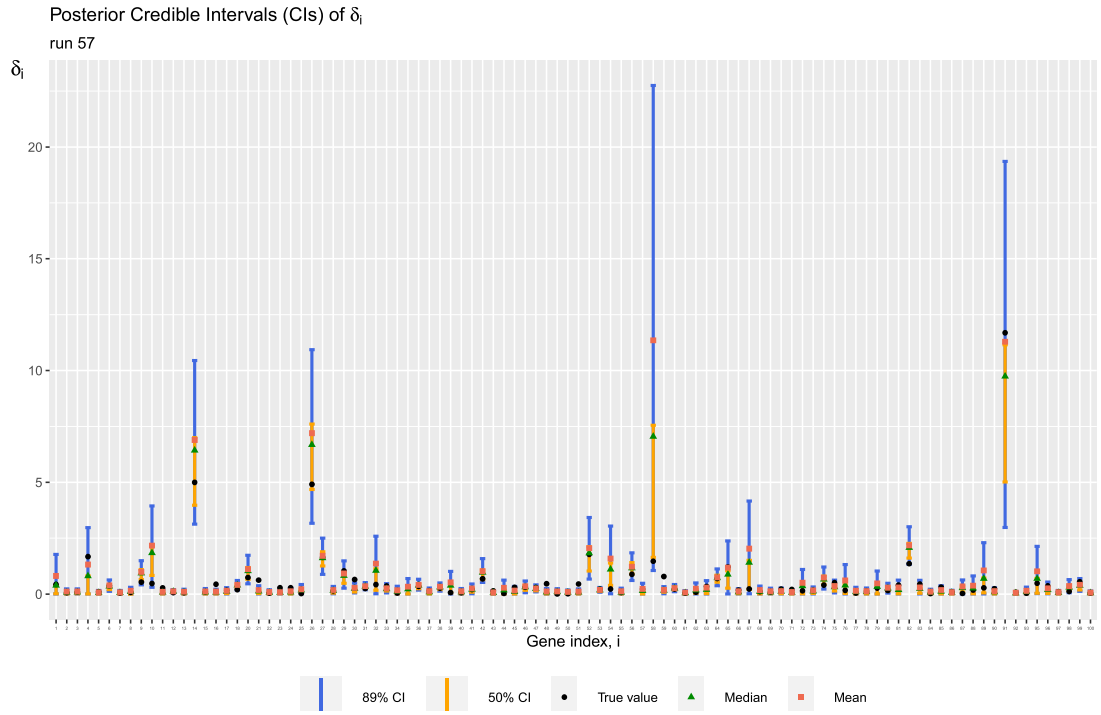

Figure 3.14: True values (δ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for μ_i, for biological genes $i = 1, \ldots, q_0$ ($q_0 = 98$), and for the 57th run of synthetic data generation and estimation procedure. Inferred from the regression BASiCS model with the fixed dataset $\mathbf{X}^{(2)*}$.

The similarity between the 29th replication and the other 99 replications is also true for the expected gene-expression level parameter μ_i. In Figure 3.15, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 98 expected gene expression level parameter μ_i, from a different run (replication number 57) of regression BASiCS MCMC on $\mathbf{X}^{(2)*}$. Among 98 expected gene expression level parameter μ_i, 5 of the true values μ_i^* do not fall into the 89% Highest Density Credible Interval, 56 of the true values do not fall into the 50% Highest Density Credible Interval.

For the 57th estimation procedure on fixed synthetic dataset $\mathbf{X}^{(2)*}$ with regression BASiCS, if we consider the true values of δ_i which fall outside of the corresponding

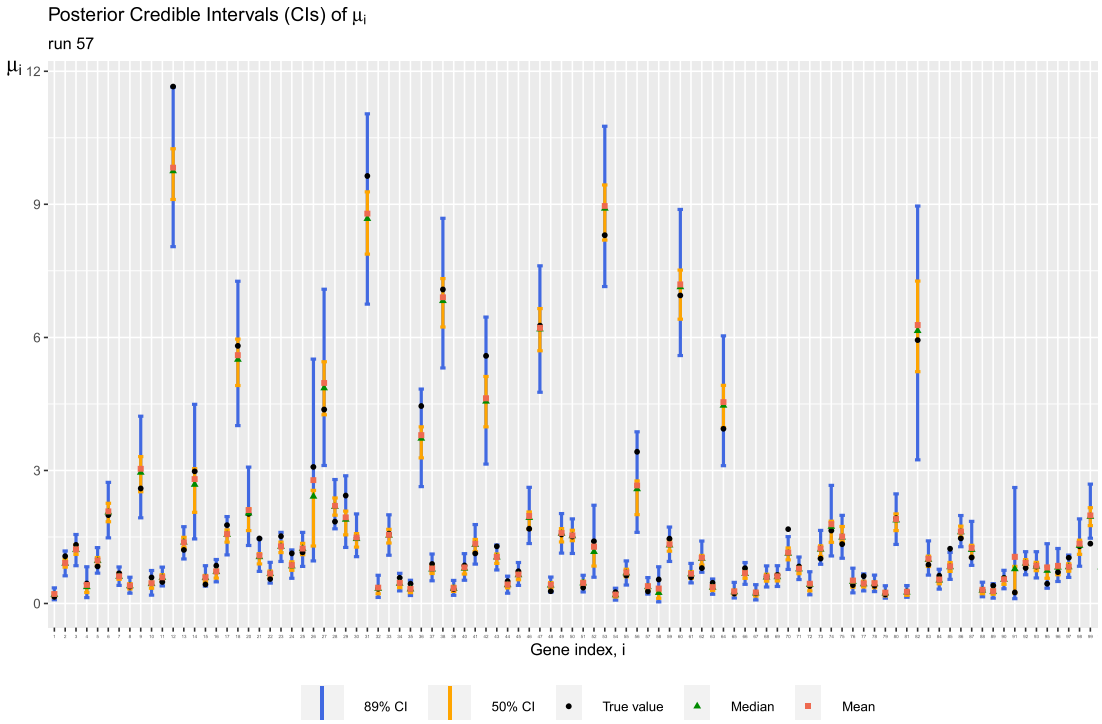

Figure 3.15: True values (δ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for μ_i, for biological genes $i = 1,\ldots,q_0$ ($q_0 = 98$), and for the 57th run of synthetic data generation and estimation procedure. Inferred from the regression BASiCS model with the fixed dataset $\mathbf{X}^{(2)*}$.

89% Highest Density Credible Interval, and the true values of μ_i which fall outside of the corresponding 89% Highest Density Credible Interval, we find that they correspond to different genes. In particular, the δ_i^* that do not fall into the 89% Highest Density Credible Interval are δ_{11}^*, δ_{16}^*, δ_{21}^*, δ_{23}^*, δ_{24}^*, δ_{30}^*, δ_{48}^*, δ_{51}^*, δ_{59}^* and δ_{71}^*, while the μ_i^* that do not fall into the 89% Highest Density Credible Interval are μ_{12}^*, μ_{21}^*, μ_{70}^*, μ_{85}^* and μ_{99}^*. Here only μ_{12} has a true value too high (11.66) that the MCMC could not recover, other true values of μ_i which are out of the corresponding 89% Highest Density Credible Interval are small values between $(0,2)$. They are outside of the 89% Credible Interval because the posteriors has smaller variance.

To give an overview of the estimation accuracy of posterior medians inferred from the regression BASiCS model with $\mathbf{X}^{(2)*}$, we plot all the gene-specific parameters (δ_i and μ_i, $i = 1,\ldots,q_0$) from 100 replication runs with this one fixed dataset $\mathbf{X}^{(2)*}$, as

shown in Figure 3.16. Here the x-axis of each point corresponds to the true value of that gene-specific parameter δ_i or μ_i ($i = 1, \ldots, q_0$), which we used to simulate dataset $\mathbf{X}^{(2)*}$. Since all 100 replications are run on the one fixed dataset $\mathbf{X}^{(2)*}$, naturally the plot shows that each true value on the x-axis corresponds to 100 posterior medians estimated from 100 replications of posterior inference. If the posterior median as point estimate is accurate, we would expect all the coloured points fall on the line $y = x$.

Figure 3.16 shows that more posterior medians of μ_i fall inside the 20% relative error range when compared to δ_i. When compared to Figure 3.7, more posterior medians of μ_i inferred from regression BASiCS falls inside the 20% than the posterior medians of μ_i inferred from non-regression BASiCS. However, compared to the posterior medians of δ_i inferred from non-regression BASiCS, a few δ_i in regression BASiCS model have more varying posterior median value across replication runs, indicating more stochasticity across replication runs.

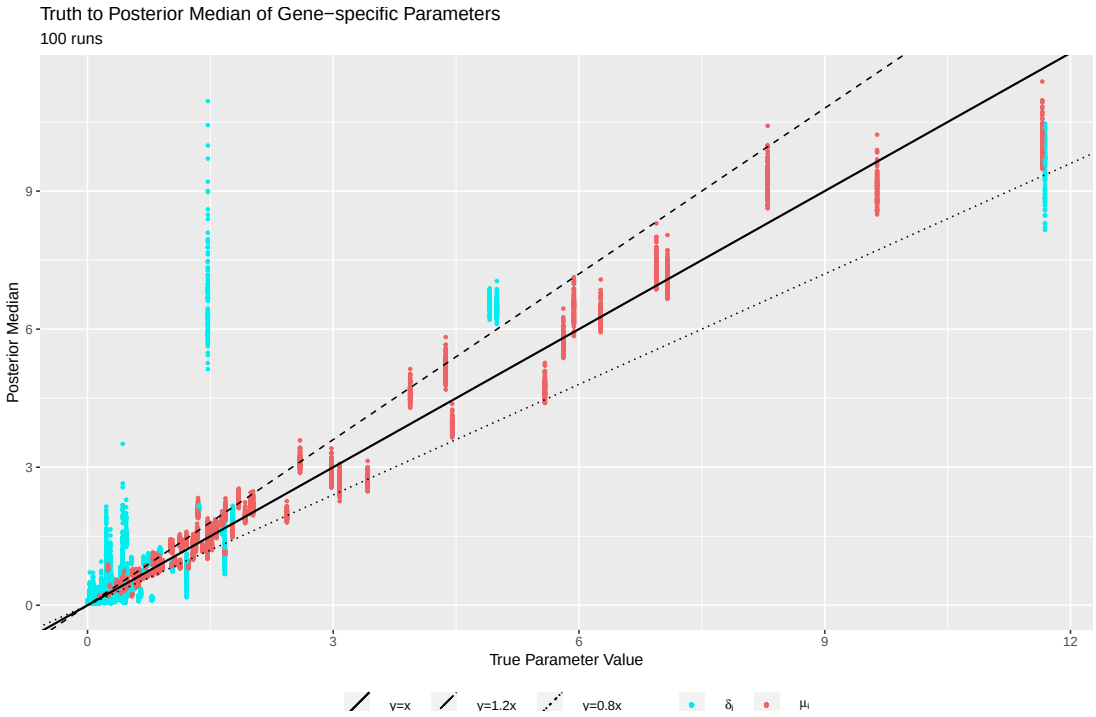

Figure 3.16: True value to posterior median of all gene-specific parameters, 100 replications, inferred from the regression BASiCS model with the fixed dataset $\mathbf{X}^{(1)}$.

This subsection so far shows the recovery of parameters using the regression BASiCS model on one particular synthetic dataset $\mathbf{X}^{(2)*}$. To prove the generality of our analysis above, we simulate 100 datasets of 100 genes, 10 spike-in genes, and 50 cells from regression BASiCS model (3.5)-(3.19) (Eling *et al.*, 2018). After the required data preprocessing, we plug each dataset back into the regression BASiCS MCMC for one replication, resulting in 100 posterior samples for δ_i, μ_i, ν_j, Φ_j, s_j, θ, $i = 1,\ldots,q_0$, $j = 1,\ldots,n$. Here we still focus on gene-specific parameters δ_i and μ_i. As examples we focus on the results from two synthetic dataset, the 5th and the 60th. Some of the analysis below may seem repetitive, but note that we are only presenting them to demonstrate that the problems exposed in Figure 3.12-3.15 are general rather than a coincidence caused by any particular dataset.

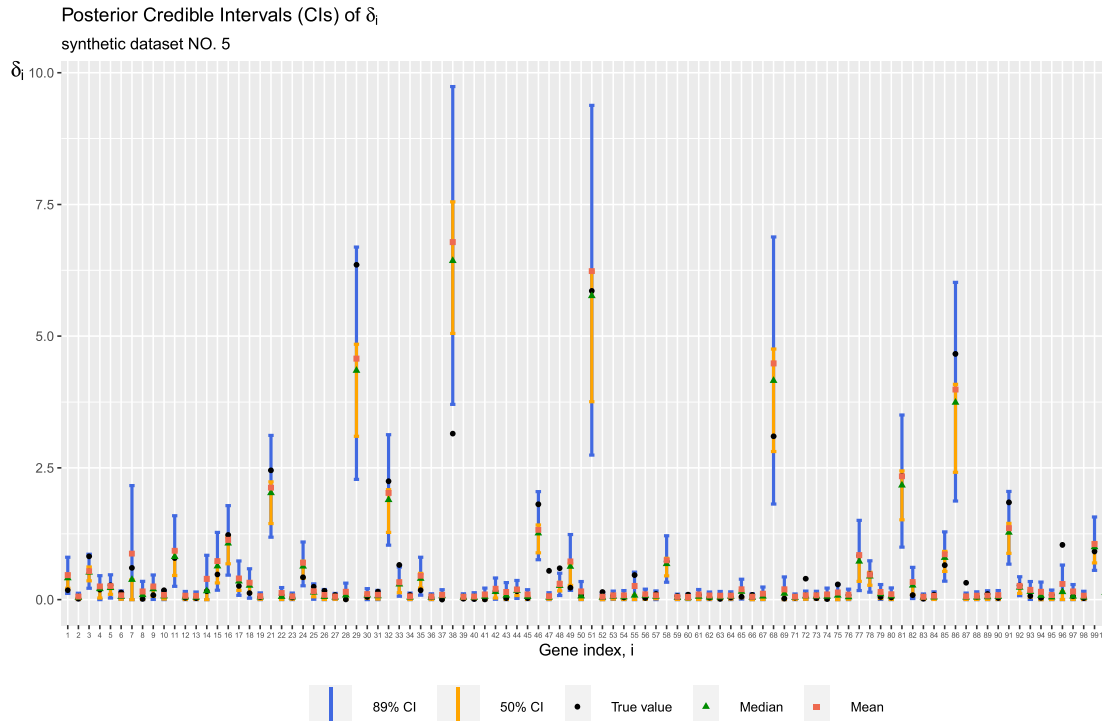

Figure 3.17: True values (δ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for δ_i, for biological genes $i = 1,\ldots,q_0$, and for estimation procedure on the 5th synthetic dataset generated generated with regression BASiCS model.

In Figure 3.17, we plot the 89% Highest Density Intervals, 50% Highest Density

Intervals, posterior medians, posterior means and the ground truth for each of the 98 biological gene-specific variation parameter δ_i, from the one run of regression BASiCS MCMC on the 5th generated dataset. Among 98 biological gene-specific variation parameters δ_i, 10 of the true values do not fall into the 89% Highest Density Credible Interval, and 51 of the true value do not fall into the 50% Highest Density Credible Interval. Apart from δ_{38} which has a relatively high true value (3.15) and was overestimated, the other 9 true δ_i^* outside of 89% Highest Density Interval are small values between $(0, 1.04)$ with very narrow posterior Highest Density Interval. In this case, the posterior median could still act as fair point estimates for them. For most δ_i, the variance of the posteriors looks much smaller compared to Figure 3.8, but such precision only occurrs on the posteriors of δ_i with small true values.

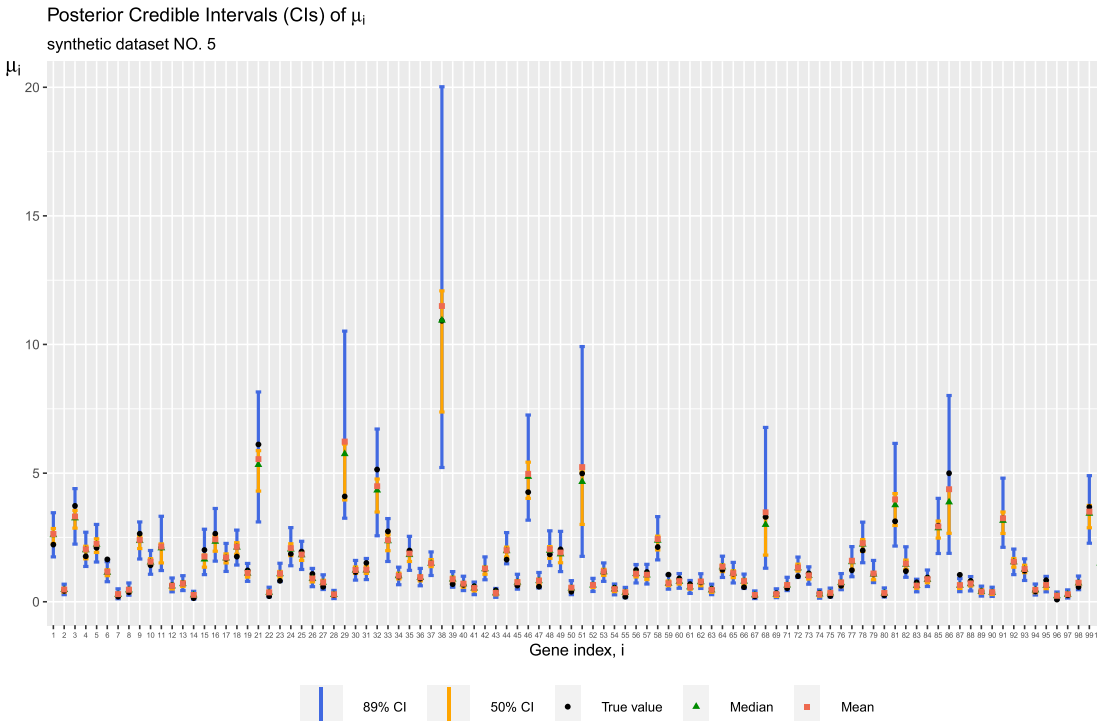

Figure 3.18: True values (μ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for μ_i, for biological genes $i = 1, \ldots, q_0$, and for estimation procedure on the 5th synthetic dataset generated generated with regression BASiCS model.

In Figure 3.18, we plot the 89% Highest Density Intervals, 50% Highest Density

Intervals, posterior medians, posterior means and the ground truth for each of the 98 expected gene-expression level parameter μ_i, from the one run of regression BASiCS MCMC on the 5th generated dataset. Among 98 expected gene expression level parameter μ_i, 4 of the true values do not fall into the 89% Highest Density Credible Interval, 51 of the true values do not fall into the 50% Highest Density Credible Interval.

For the 5th synthetic data generation and estimation procedure with regression BASiCS, if we consider the true values of δ_i which fall outside of the corresponding 89% Highest Density Credible Interval, and the true values of μ_i which fall outside of the corresponding 89% Highest Density Credible Interval, we find that they correspond to different genes. In particular, the δ_i^* that do not fall into the 89% Highest Density Credible Interval are δ_{10}^*, δ_{33}^*, δ_{38}^*, δ_{47}^*, δ_{48}^*, δ_{52}^*, δ_{72}^*, δ_{75}^*, δ_{87}^* and δ_{96}^*, while the μ_i^* that do not fall into the 89% Highest Density Credible Interval are μ_6^*, μ_{59}^*, μ_{87}^* and μ_{96}^*. Here all the true values of μ_i which fall out of the corresponding 89% Highest Density Credible Interval are small values between $(0,2)$. Most of them are outside of the 89% Credible Interval because the posteriors have smaller variance.

In Figure 3.19, we plot 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 97 biological gene-specific variation parameter δ_i, from the one run of regression BASiCS MCMC on the 60th generated dataset. Among 97 biological gene-specific variation parameter δ_i, 14 of the true values do not fall into the 89% Highest Density Credible Interval, 52 of the true value do not fall into the 50% Highest Density Credible Interval. This time 12 true values outside of 89% Highest Density Interval are large values in $(1.7,31)$ with very large posterior Highest Density Interval, indicating the posterior could not capture the precise level of δ_i when the true value is large.

In Figure 3.20, we plot the 89% Highest Density Intervals, 50% Highest Density Intervals, posterior medians, posterior means and the ground truth for each of the 97 expected gene-expression level parameter μ_i, from the one run of regression BASiCS MCMC on the 60th generated dataset. Among 97 expected gene expression level parameter μ_i, 4 of the true values do not fall into the 89% Highest Density Credible Interval, 54 of the true values do not fall into the 50% Highest Density Credible Interval.

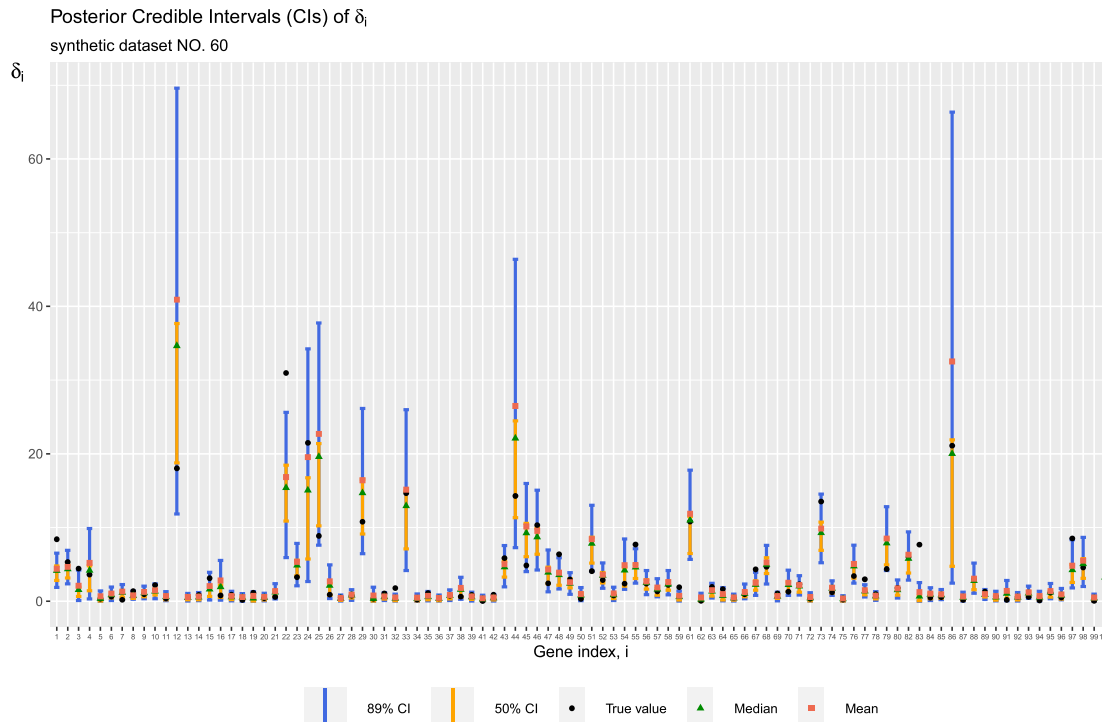

Figure 3.19: True values (δ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for δ_i, for biological genes $i = 1,\ldots,q_0$, and for estimation procedure on the 60th synthetic dataset generated generated with regression BASiCS model.

For the 60th synthetic data generation and estimation procedure with regression BASiCS, if we consider the true values of δ_i which fall outside of the corresponding 89% Highest Density Credible Interval, and the true values of μ_i which fall outside of the corresponding 89% Highest Density Credible Interval, we find that they correspond to different genes. In particular, the δ_i^* that do not fall into the 89% Highest Density Credible Interval are δ_1^*, δ_3^*, δ_7^*, δ_{22}^*, δ_{32}^*, δ_{48}^*, δ_{51}^*, δ_{55}^*, δ_{59}^*, δ_{67}^*, δ_{77}^*, δ_{83}^*, δ_{97}^* and δ_{100}^*, while the μ_i^* that do not fall into the 89% Highest Density Credible Interval are μ_{12}^*, μ_{59}^*, μ_{83}^* and μ_{92}^*. Apart from μ_{12} which has relatively high true value (16.22) and were underestimated, the other 3 true value of μ_i which fall outside of the corresponding 89% Highest Density Interval are small values between $(0,2)$ with very narrow posterior Highest Density Interval. In this case the posterior median could still act as fair point estimates for the true value.

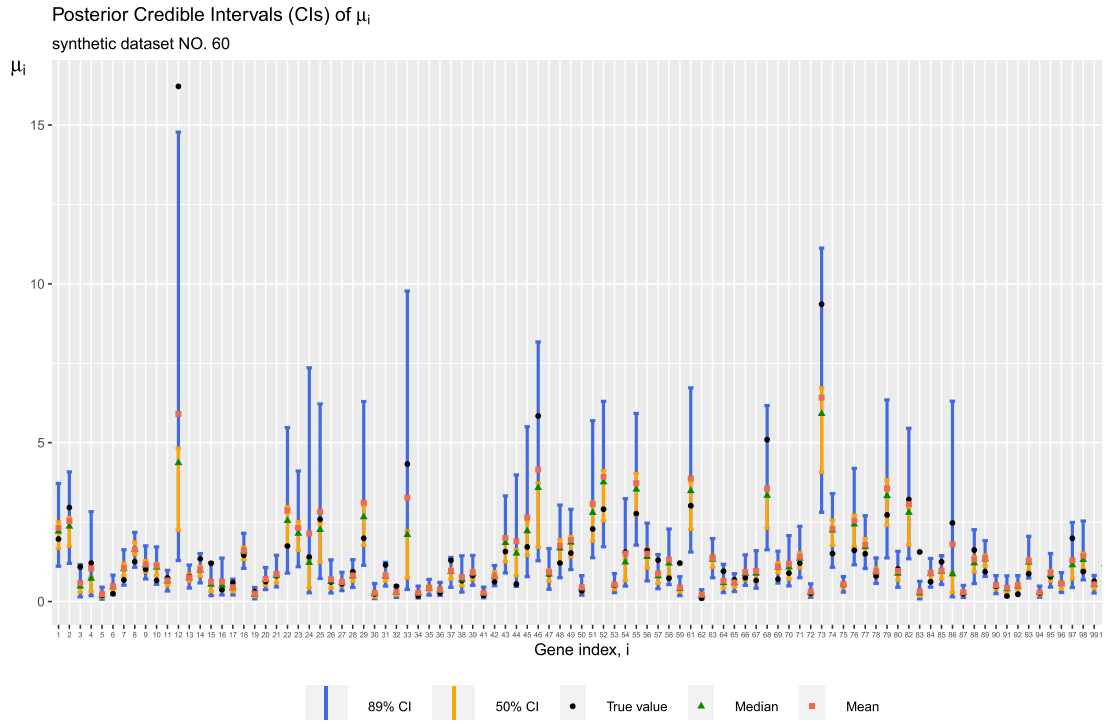

Figure 3.20: True values (μ_i^*) and posterior estimates (median, mean, 89% and 50% Highest Density Credible Intervals) for μ_i, for biological genes $i = 1,\ldots,q_0$, and for estimation procedure on the 60th synthetic dataset generated generated with regression BASiCS model.

3.2.2 Posterior Predictive Check

Following Gelman *et al.* (2013), we use Posterior Predictive Check (PPC) to assess our model fit. For a run in Subsection 3.2.1, from each set of parameters $\{\mu_i, \nu_j, \Phi_j, \delta_i\}$ in 1000 posterior samples, we simulate a posterior predictive value of biological gene expression count $X_{ij}^{(1)}$ for non-regression BASICS and $X_{ij}^{(2)}$ for regression BASiCS, from Equation (3.11), which results in 1000 posterior predictive $X_{ij}^{(1)}$ for non-regression BASiCS and 1000 posterior predictive $X_{ij}^{(2)}$ for regression BASiCS, to compare with the true data $X_{ij}^{(1)*}$ and $X_{ij}^{(2)*}$, respectively.

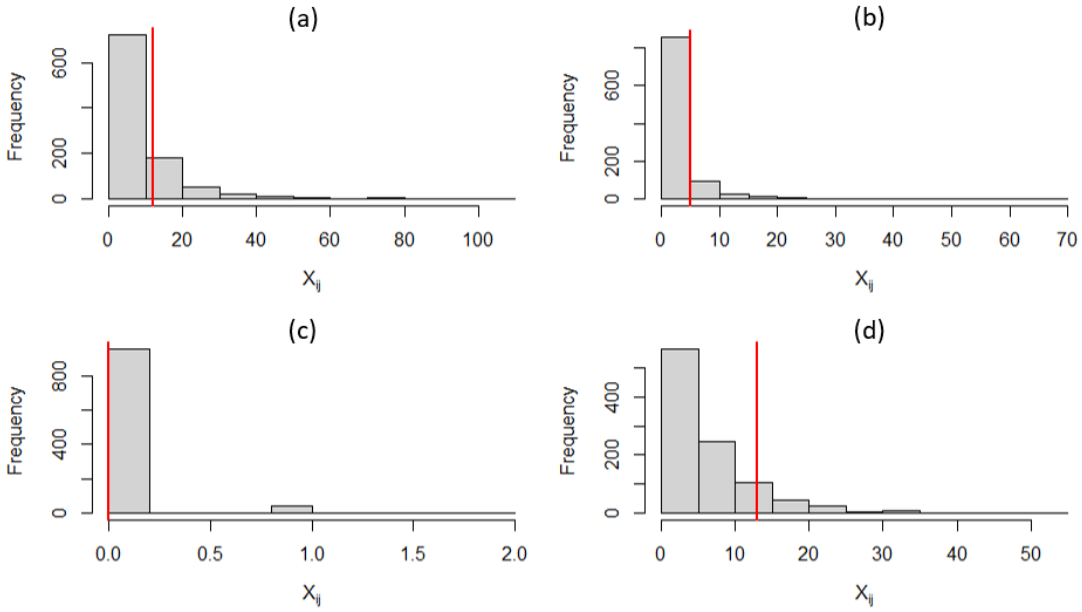

Figure 3.21: **histogram:** posterior predictive distribution of $X_{ij}^{(1)}$, simulated from the posteriors of run 1.
red line: input data $X_{ij}^{(1)*}$.
(a): gene $i = 7$, cell $j = 1$. **(b):** gene $i = 10$, cell $j = 12$.
(c): gene $i = 38$, cell $j = 14$. **(d):** gene $i = 56$, cell $j = 33$.

Figure 3.21 and Figure 3.22 plot the histogram of posterior predictive $X_{ij}^{(1)}$ and $X_{ij}^{(2)}$ and the vertical line of $x = X_{ij}^{(1)*}$ and $x = X_{ij}^{(2)*}$ for non-regression BASiCS model and regression BASiCS model, respectively. We can see that the regression BASiCS model (Eling *et al.*, 2018) performs better compared with the non-regression BASiCS model (Vallejos *et al.*, 2016).

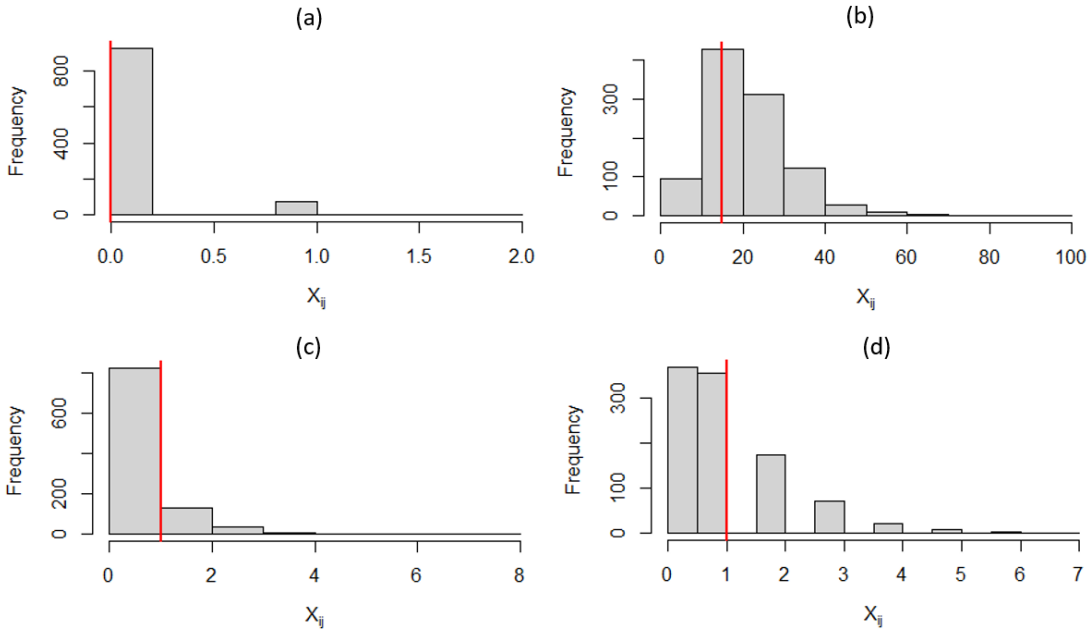

Figure 3.22: **histogram:** posterior predictive distribution of $X_{ij}^{(2)}$, simulated from the posteriors of run 1, regression BASiCS model.
red line: input data $X_{ij}^{(2)*}$.
(a): gene $i = 1$, cell $j = 9$. **(b):** gene $i = 12$, cell $j = 2$.
(c): gene $i = 15$, cell $j = 11$. **(d):** gene $i = 77$, cell $j = 7$.

3.2.3 Sensitivity to contamination on prior

In this subsection, we modified the non-regression part of BASiCS package Vallejos *et al.* (2016) so that we can pass an ε into the prior of δ_i:

$$\delta_i \overset{ind}{\sim} (1-\varepsilon)\cdot \text{log-Normal}\left(0,\sigma_\delta^2\right) + \varepsilon \cdot \text{Gamma}\left(a_\delta, b_\delta\right), \tag{3.37}$$

which gives us a continuous range of choice for the prior distribution of δ_i.

We simulate one dataset from non-regression BASiCS model (3.1)-(3.10) (Vallejos *et al.*, 2016), simulating δ_i from log-Normal distribution, which is equivalent to let $\varepsilon = 0$ in (3.37). Then we plug this one dataset back to the MCMC of Vallejos *et al.* (2016), with fixed prior-hyperparameter values in Table 3.2 and changing $\varepsilon \in \{0, 0.25, 0.5, 0.75, 1\}$.

To investigate the stochastic variation in MCMC result, for each fixed $\varepsilon \in \{0, 0.25, 0.5, 0.75, 1\}$, we replicate the MCMC for 200 times. Each MCMC was run for

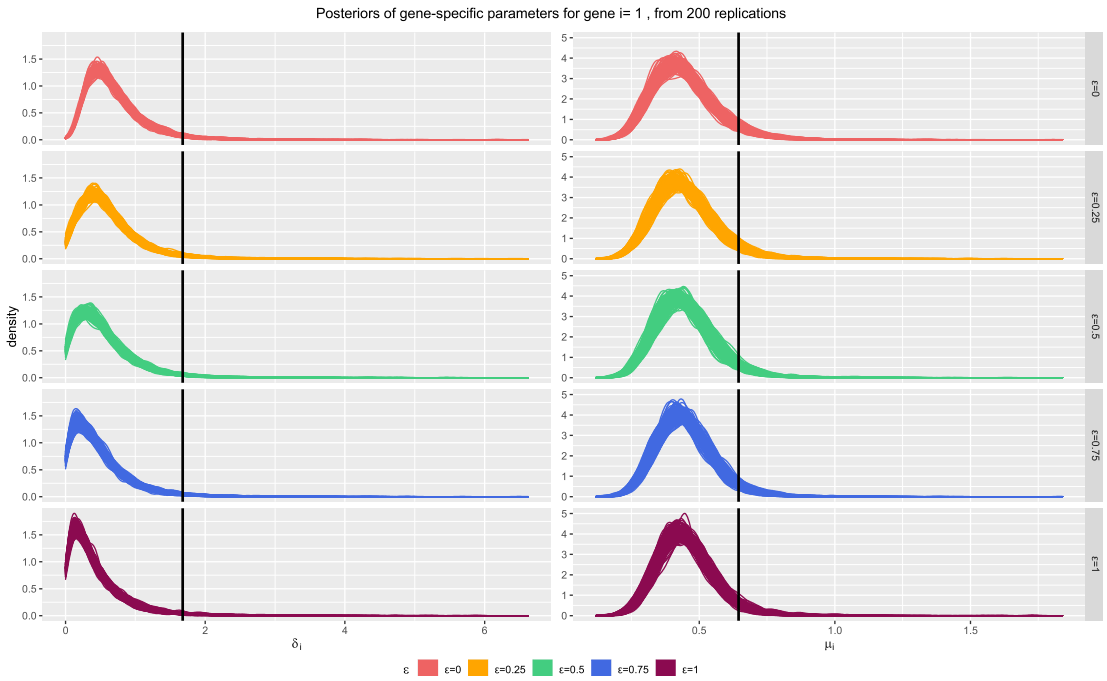

Figure 3.23: Gene-specific posterior results with one fixed synthetic dataset, varying ε, i.e. varying the prior of δ_i, for gene $i = 1, \ldots, q_0$, with 200 replications for each ε.
The curves: Posterior samples of δ_i and μ_i, for gene $i = 1$.
The vertical line: The true value of δ_i and μ_i used in data generation, for gene $i = 1$.

15,000 iterations, 10,000 burns and thinned by 5, resulting in length-1000 posterior samples. We plot the posterior distribution curves of these posterior samples.

Figure 3.23 shows several things. Firstly, focusing on one ε value, the curves imply a degree of stochastic variation of the MCMC posterior samples. Secondly, from the relative position of the vertical line x = ground truth and the curves we can see that, on the one hand, the recovery of δ_1 worsens with larger ε, i.e. a prior distribution,

$$\delta_i \overset{ind}{\sim} (1-\varepsilon) \cdot \text{log-Normal}\left(0, \sigma_\delta^2\right) + \varepsilon \cdot \text{Gamma}\left(a_\delta, b_\delta\right),$$

more different from the true distribution we simulated δ_i from,

$$\delta_i \overset{ind}{\sim} \text{log-Normal}\left(0, \sigma_\delta^2\right).$$

On the other hand, the recovery of μ_i does not suffer a significant change due to the change of ε, implying the change of the prior of δ_i does not affect μ_i much.

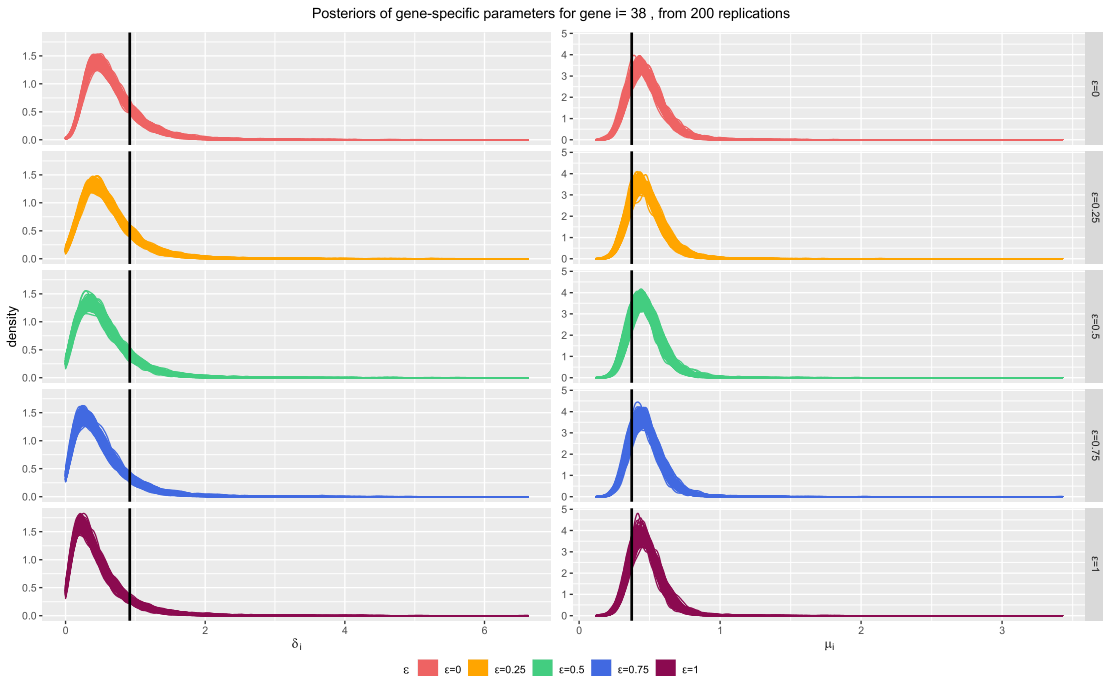

Figure 3.24: Gene-specific posterior results with one fixed synthetic dataset, varying ε, i.e. varying the prior of δ_i, for gene $i = 1, \ldots, q_0$, with 200 replications for each ε.
The curves: Posterior samples of δ_i and μ_i, for gene $i = 38$.
The vertical line: The true value of δ_i and μ_i used in data generation, for gene $i = 38$.

However, it is not always the same case for other genes in this synthetic dataset. Figure 3.24 shows that for gene 38 in this synthetic dataset, the MCMC recovery of δ_{38} is consistently wrong, regardless of the choice of ε, i.e. the prior distribution.

Figure 3.25 shows the posterior results of the cell-specific parameters. We can see that the change of the prior for gene-specific parameter δ_i does not affect the inference of cell-specific parameters. In particular, from the results from replications with fixed ε, the posterior ν_j and ϕ_j has higher stochasticity compared to the posterior of other parameters in the same replications. This result is consistent with the convergence diagnosis showed in the Supplementary material of Vallejos *et al.* (2016), where the trace plots indicate that the Monte Carlo Markov Chain of ν_j and ϕ_j do not converge in this version of BASiCS MCMC.

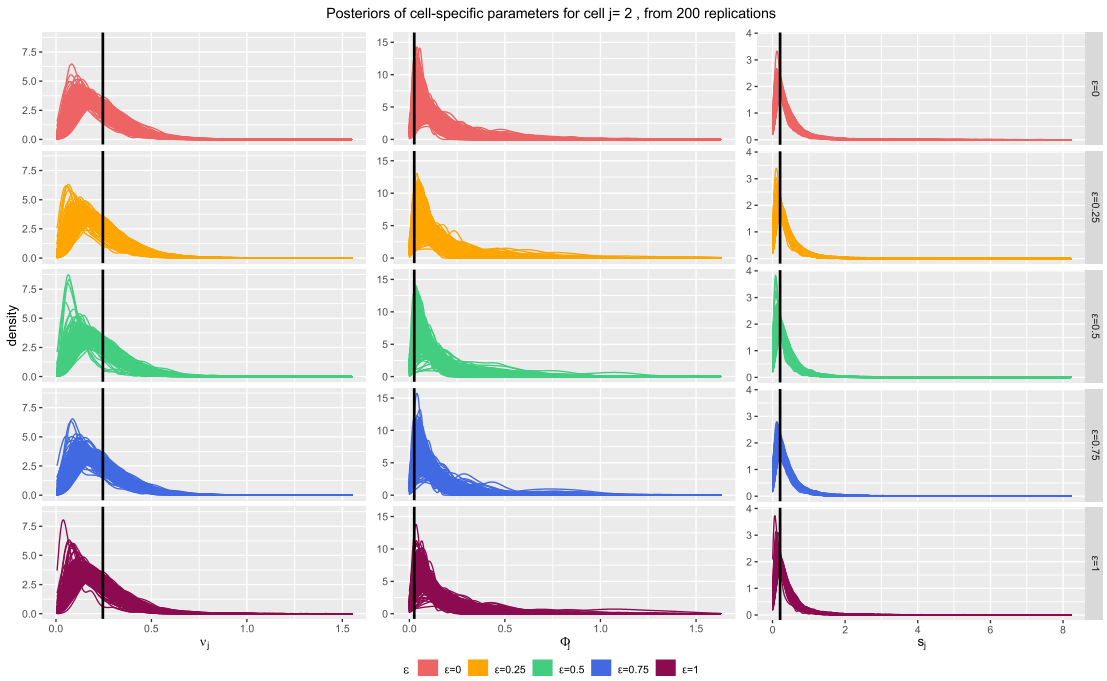

Figure 3.25: Gene-specific posterior results with one fixed synthetic dataset, varying ε, i.e. varying the prior of δ_i, for gene $i = 1,\ldots,q_0$, with 200 replications for each ε.
The curves: Posterior samples of ν_j, ϕ_j and s_j, for cell $j = 2$.
The vertical line: The true value of ν_j, ϕ_j and s_j used in data generation, for cell $j = 2$.

3.2.4 Simulation based calibration adapted for BHM with high-dimensional parameters

BASiCS is implemented via MCMC , therefore it can be assessed with the extended SBC with the Effective Sample Size assessment, as described in Subsection 3.1.4. Notably, in complex real data models like BASiCS, we have multiple measurements of interest $c_1(\boldsymbol{\zeta}),\ldots,c_M(\boldsymbol{\zeta})$. Therefore, similar to Talts *et al.* (2018), we assess the minimal Effective Sample Size with respect to all the measurements of interest, that is, if:

$$\min_{m=1,\ldots,M}\left\{N_{eff}\left[c_m\right]\right\} > L. \tag{3.38}$$

To implement this approach for the BHM in Subsection 3.1.2, we define c as the projection function to each individual parameter in the parameter vector $\boldsymbol{\zeta}$. This is similar to the identity function c proposed in Schad *et al.* (2021) for models with a

single parameter, where the diagnosis consists of checking if the rank statistic for the parameter mirrors a uniform distribution. However, in BHMs like BASiCS, the approach can be adapted for the high-dimensional parameter space (Talts *et al.*, 2018) $\boldsymbol{\zeta} = \left(\delta_1,\ldots,\delta_{q_0},\mu_1,\ldots,\mu_{q_0},\nu_1,\ldots,\nu_n,\phi_1,\ldots,\phi_n,s_1,\ldots,s_n,\theta\right)$. We define $c_{\delta_i}(\boldsymbol{\zeta}) = \delta_i$, $c_{\mu_i}(\boldsymbol{\zeta}) = \mu_i$, $c_{\nu_j}(\boldsymbol{\zeta}) = \nu_j$, $c_{\Phi_j}(\boldsymbol{\zeta}) = \Phi_j$,$c_{s_j}(\boldsymbol{\zeta}) = s_j$,$c_\theta(\boldsymbol{\zeta}) = \theta$ in Equation (3.38), and we assess if:

$$\min_{i,j}\left\{N_{eff}[\delta_i], N_{eff}[\mu_i], N_{eff}[\nu_j], N_{eff}[\Phi_j], N_{eff}[s_j], N_{eff}[\theta]\right\} > L.$$

Algorithm 4 SBC for BASiCS: individual parameters

Require: Data generating model $\pi(X_{ij}|\delta_i,\mu_i,\nu_j,\Phi_j,s_j,\theta)$, prior distribution $\pi(\delta_i),\pi(\mu_i),\pi(\nu_j),\pi(\Phi_j),\pi(s_j),\pi(\theta)$, the number of rank statistic K, , the number of MCMC iterations L', the resulted posterior MCMC chain length N_{sample}, the number of posterior sample used for calculating each rank statistic $L \approx \frac{N_{sample}}{10}$.

Initialise

while k in $(1:K)$ **do**

Draw prior sample for i=1,...,q, j=1,...,n:
$\widetilde{\delta_i^{(k)}} \sim \pi(\delta_i)$, $\widetilde{\mu_i^{(k)}} \sim \pi(\mu_i)$, $\widetilde{\nu_j^{(k)}} \sim \pi(\nu_j)$, $\widetilde{\Phi_j^{(k)}} \sim \pi(\Phi_j)$, $\widetilde{s_j^{(k)}} \sim \pi(s_j)$, $\widetilde{\theta^{(k)}} \sim \pi(\theta)$.

Draw a simulated dataset, for $i = 1,\ldots,q,\ j = 1,\ldots,n$:
$\widetilde{X_{ij}^{(k)}} \sim \pi\left(X_{ij}|\widetilde{\delta_i^{(k)}},\widetilde{\mu_i^{(k)}},\widetilde{\nu_j^{(k)}},\widetilde{\Phi_j^{(k)}},\widetilde{s_j^{(k)}},\widetilde{\theta^{(k)}}\right)$.

Run the corresponding MCMC algorithm with Input dataset $\widetilde{\mathbf{X}^{(k)}} = \left(\widetilde{X_{ij}^{(k)}}\right)$ in `BASiCS` package for L' iterations to generate the correlated posterior sample chain of length N_{sample} from $\pi(\delta_i^{(k)},\mu_i^{(k)},\nu_j^{(k)},\Phi_j^{(k)},s_j^{(k)},\theta^{(k)}|\tilde{y^{(k)}})$:
$\left(\delta_i^{(k)}(t),\mu_i^{(k)}(t),\nu_j^{(k)}(t),\Phi_j^{(k)}(t),s_j^{(k)}(t),\theta^{(k)}(t)\right)$ for $t = 1,\ldots,N_{sample}$, , $i = 1,\ldots,q,\ j = 1,\ldots,n$.

Call `R` function `LaplacesDemon::ESS` (Statisticat & LLC., 2021) to compute the effective sample size for each parameter, $N_{eff}^{(k)}[\delta_i]$, $N_{eff}^{(k)}[\mu_i]$, $N_{eff}^{(k)}[\nu_j]$, $N_{eff}^{(k)}[\Phi_j]$, $N_{eff}^{(k)}[s_j]$, $N_{eff}^{(k)}[\theta]$, for $i = 1,\ldots,q,$, $j = 1,\ldots,n$.

$$N_{eff}^{(k)} = \min\{N_{eff}^{(k)}[\delta_i], N_{eff}^{(k)}[\mu_i], N_{eff}^{(k)}[\nu_j], N_{eff}^{(k)}[\Phi_j], N_{eff}^{(k)}[s_j], N_{eff}^{(k)}[\theta].\}$$

if $N_{eff}^{(k)} < L$ **then**

rerun the MCMC for $\frac{L' \cdot L}{N_{eff}^{(k)}}$ iterations.

else

For each $i = 1, \ldots, q_0$, $j = 1, \ldots, n$, thin the posterior MCMC chain to L samples $\left\{\left(\delta_i^{(k)}(t_l), \mu_i^{(k)}(t_l), \nu_j^{(k)}(t_l), \Phi_j^{(k)}(t_l), s_j^{(k)}(t_l), \theta^{(k)}(t_l)\right)\right\}_{l=1}^{L}$, and truncate any leftover sample from the k-th run after $\left(\delta_i^{(k)}(t_L), \mu_i^{(k)}(t_L), \nu_j^{(k)}(t_L), \Phi_j^{(k)}(t_L), s_j^{(k)}(t_L), \theta^{(k)}(t_L)\right)$.

end if

Compute rank statistic for $i = 1, \ldots, q, j = 1, \ldots, n$:

$$\begin{aligned} r^{(k)}[\delta_i] &= r\left(\left\{\delta_i^{(k)}(t_1), \ldots, \delta_i^{(k)}(t_L)\right\}, \widetilde{\delta_i^{(k)}}\right) \\ &= \sum_{l=1}^{L} \mathbb{1}_{\left\{t_l : \delta_i^{(k)}(t_l) < \widetilde{\delta_i^{(k)}}\right\}}\left(\delta_i^{(k)}(t_l)\right). \end{aligned}$$

Similarly, calculate $r^{(k)}[\mu_i]$, $r^{(k)}[\nu_j]$, $r^{(k)}[\Phi_j]$, $r^{(k)}[s_j]$, $r^{(k)}[\theta]$.

end while

Plot the histogram of rank statistic $r_{ij}^{(k)}$. for $k = 1, \ldots, K$.

Check the uniformity of the histogram of $r_{ij}^{(k)}$. for $k = 1, \ldots, K$.

This leads to Algorithm 4, which we implement and apply to the BASiCS non-regression model. In order to check the deviation of rank statistics from Uniform($\{0, 1, \ldots, L\}$), we plot the empirical cumulative density function (ECDF) and the expected CDF behaviour of Uniform($\{0, 1, \ldots, L\}$).

As Algorithm 4 demonstrates, in $k = 1, \ldots, K$ runs, all the parameters $\widetilde{\delta_i^{(k)}}$, $\widetilde{\mu_i^{(k)}}$, $\widetilde{\nu_j^{(k)}}$, $\widetilde{\Phi_j^{(k)}}$, $\widetilde{s_j^{(k)}}$, $\widetilde{\theta(k)}$ are re-simulated from the corresponding $i.i.d.$ prior distribution for all $i = 1, \ldots, q_0$, $j = 1, \ldots, n$. Therefore, the rank statistic of each δ_i is equivalent to each other, the same applies to $\mu_i, \nu_j, \Phi_j, s_j, \theta$. Without losing generality, in Figure 3.26, we plot the ECDF of $\delta_1, s_1, \theta, \Phi_1, \nu_1, \mu_1$. From Figure 3.26, one can observe that the behaviour of the rank statistics for most of the parameters are close to the uniform distribution. On the other hand, the rank statistics for θ are far from the uniform distribution, suggesting that θ is likely to be underestimated in this model. This illustrates the applicability of the techniques in Talts *et al.* (2018) for diagnosing the estimation of parameters in a BHM such as that in Figure 3.1.

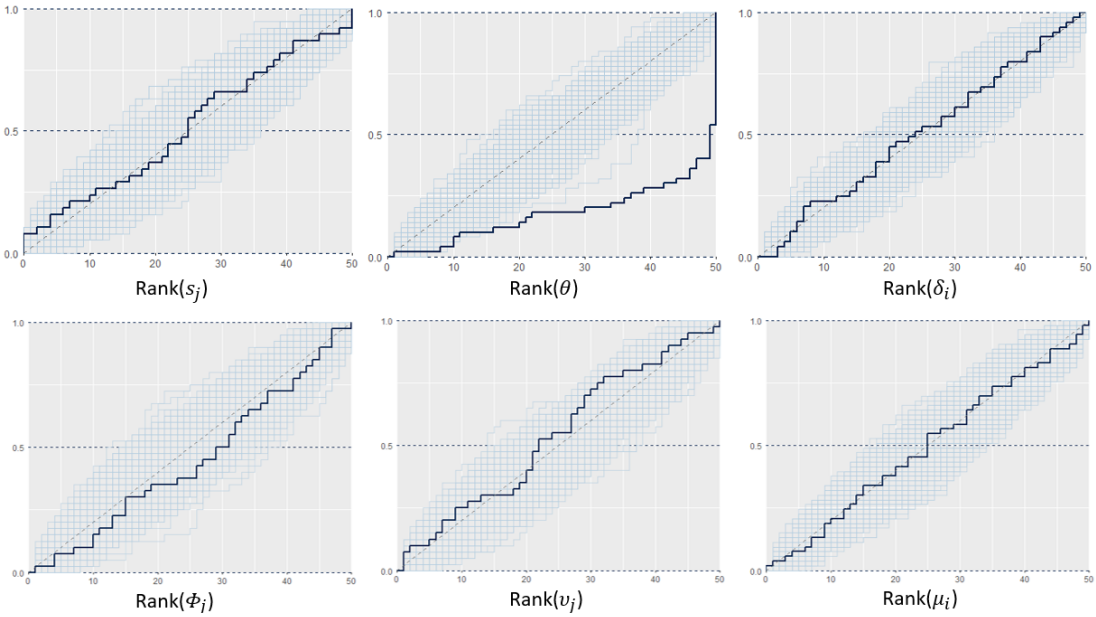

Figure 3.26: SBC results of non-regression BASiCS (Vallejos *et al.*, 2016). For the model parameters s_j, δ_i, θ, Φ_j, ν_j and μ_i, the ECDF of the calculated rank statistic (dark blue) and 500 uniform samples (light blue) are plotted. Without loosing generality, here $i = 1$, $j = 1$.

We also perform the Simulation based calibration procedure described in Algorithm 4 on regression BASiCS model, adapted from Talts *et al.* (2018). Similar to the arguments in the last paragraph, without losing generality, in Figure 3.27 we plot the ECDF for the calculated rank statistics and the uniform distribution for $\delta_1, s_1, \theta, \Phi_1, \nu_1, \mu_1$. In terms of the Simulation-based Calibration results, the behaviours observed for the regression BASiCS model in Figure 3.27 are similar to those observed for the non-regression BASiCS model in Figure 3.26. In particular, the rank statistics for most parameters in Figure 3.27 are close to a uniform distribution. The low ranks of s_j are seen slightly more often in the computed ranks than we would expect from a uniform distribution, and the rank statistic of θ is far from the range of the uniform distribution. Thus, this suggests that θ tends to be underestimated in the regression BASiCS model.

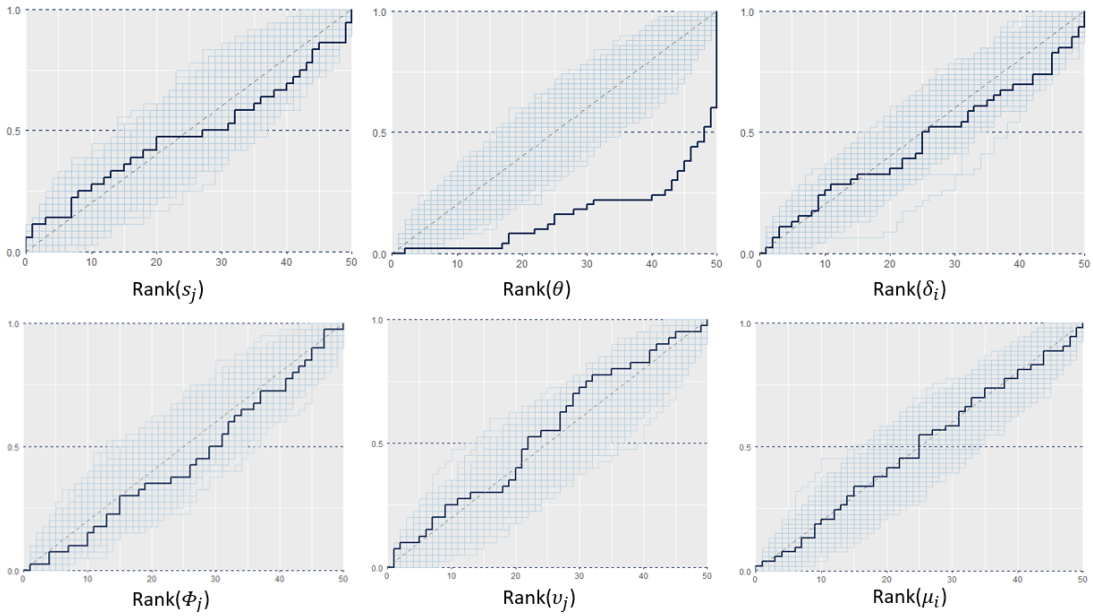

Figure 3.27: SBC results of regression BASiCS (Eling *et al.*, 2018). For the model parameters s_j, δ_i and θ, Φ_j, ν_j, μ_i, the ECDF of the calculated rank statistic (dark blue) and 500 uniform samples (light blue) are plotted. Without loosing generality, here $i = 1$, $j = 1$.

3.3 Conclusion

In summary, we have illustrated a simulation-based evaluation framework for BHMs for scRNAseq. We explored the reliability of a non-Gaussian distribution based BHM inferred via Monte Carlo Markov Chain (MCMC) algorithm, using the BASiCS framework developed by Vallejos *et al.* (2015, 2016) and by Eling *et al.* (2018) as an example. From our experiments, both posterior median and posterior mean are revealed to be inaccurate point estimates at times, showing the limitations of considering point estimates from posterior distributions for downstream analysis, when considering BHMs for scRNAseq. In Subsection 3.2.2, we also show that for a fixed given model, the effect of a contaminated prior on the posteriors varies. For the purpose of this experiment on contaminated prior distribution, we modified the `BASiCS` package from Vallejos *et al.* (2015, 2016) and Eling *et al.* (2018), providing the choice of a mixed

prior on a spectrum. We also implemented two validation methods for Bayesian models, namely the Posterior Predictive Check (Gelman *et al.*, 1996; Rubin, 1984) and Simulation based calibration (Talts *et al.*, 2018), specifically for BASiCS framework in R (R Core Team, 2013). Our analysis in Subsection 3.2.1 and 3.2.3 shows that regression BASiCS achieves some improvement over non-regression BASiCS on the posterior estimation accuracy in terms of the length of 89% credible interval and posterior predictive distribution. The Simulation based calibration method implemented in Subsection 3.2.4 returns similar results for the two models. This is because the Simulation based calibration approach implemented here relies on checking if the true value of the parameter used to generate the corresponding synthetic dataset falls inside the posterior credible interval estimated under the assumed model (Talts *et al.*, 2018). From our experiments, we identified that one of the parameters, namely the global technical noise parameter θ in BASiCS framework, is consistently underestimated, thereby suggesting the future direction for the improvement of the BASiCS framework. Since the ground truth is typically unknown in BHM, we would like to emphasise that a simulation based reliability analysis is important in validating BHM and its implementation.

Chapter 4

Scalable Bigraphical Lasso

4.1 Introduction

In this chapter we present an eigen-decomposition based two-way network inference approach for count data. The main motivation of this research is that real world problems often come with correlations between several dimensions. For example, in biology the gene expression data encodes the relationship between genes and the relationship between cells or tissues (Teng & Huang, 2009). Another example is the multi-channel electroencephalography (EEG) data from brain imaging research (Bijma *et al.*, 2005), where the data in matrix form encodes the temporal trajectory and the relationship between different channels. Recently, Gaussian graphical models have been developed for two-way network inference on matrix data, and these models are even extended to multi-network inference on tensor data. For example, Tsiligkaridis & Hero (2013) and Zhou (2014) study a matrix normal distribution where the precision matrix corresponds to the Kronecker product between the row-specific and the column-specific precision matrices. Kalaitzis *et al.* (2013) introduces Bigraphical Lasso, and Greenewald *et al.* (2019) introduces TeraLasso, both studying a multivariate normal distribution where the precision matrix corresponds to a Kronecker sum instead of a Kronecker product of matrices.

However, a multivariate normal distribution can only be applied to Gaussian data. Many datasets in different application fields come with count data, such as the scRNAseq data mentioned in Chapter 3, for which Gaussian based models are not applicable. Some methods use other distributions to infer networks from the

data. Jia *et al.* (2017) infers the gene regulation networks with a Poisson-Gamma based Bayesian Hierarchical Model, borrowing information across cells. McDavid *et al.* (2019) infers the gene regulation networks with a multivariate Hurdle model (zero-inflated mixed Gaussian). Several approaches have extended the use of Gaussian models to an appropriate continuous transformation of count data. Liu *et al.* (2009) and Liu *et al.* (2012) proposes a semiparametric approach, and Roy & Dunson (2020) proposes a nonparametric approach, while Chiquet *et al.* (2019) considered Bayesian Hierarchical Models. However, all these methods only produce a one-way network inference. Bartlett *et al.* (2021) proposes a Bayesian model with a prior having decoupled two-way sparsity to infer a dynamic network structure through time. However, the method still depends on a pre-inferred or known ordering of time.

Our method extends the semiparametric approach to enable a two-way network inference on non-Gaussian data, where the structure in both dimensions is to be inferred simultaneously. Firstly, we present a Scalable Bigraphical Lasso algorithm, reducing both the space complexity and the computational complexity of the inference, with respect to the Bigraphical Lasso algorithm originally developed by Kalaitzis *et al.* (2013). Secondly, we extend the Bigraphical model to count data by means of a semiparametric approach. Our proposed methodology not only accounts for the dependencies across both instances and features, but also reduces the computational complexity for high dimensional data.

This chapter is structured as follows: In Section 4.2 we present a detailed review on this topic; In Section 4.3 we present our Scalable Bigraphical Lasso algorithm for Gaussian data; In Section 4.4 we propose a semiparametric extension to the Bigraphical model for count data; In Section 4.5 we showcase the performance of our method on both synthetic and real datasets.

4.2 Background

4.2.1 From the matrix normal model to the Kronecker sum structure

As we have introduced in Chapter 2, for a Gaussian density, the precision matrix defines an undirected Gaussian Markov random field graph (Lauritzen, 1996), encoding

conditional independence between variables in the Gaussian model. Therefore we can induce the network structure from the support of the precision matrix. A matrix normal model with the Kronecker sum structure is proposed in Kalaitzis *et al.* (2013). If a $p \times n$ random matrix $\mathbf{Y}$ follows a matrix normal distribution,

$$\mathbf{Y} \sim \mathbf{MN}_{p\times n}\left(\mathbf{M}_{p\times n}; \mathbf{\Psi}^{-1}_{n\times n}, \mathbf{\Theta}^{-1}_{p\times p}\right),$$

with the mean matrix $\mathbf{M}_{p\times n}$, and with precision matrix $\mathbf{\Psi}_{n\times n}$ indicating the dependency structure in rows, and precision matrix $\mathbf{\Theta}_{p\times p}$ indicating the dependency structure in columns. The model can be reparametrized and shifted such that the vectorised random matrix follows an np-dimensional multivariate normal distribution (denoted as **mN**):

$$\text{vec}\,(\mathbf{Y}) \sim \mathbf{mN}_{np}\left(\mathbf{0}_{np}, \left(\mathbf{\Psi}_{n\times n} \otimes \mathbf{\Theta}_{p\times p}\right)^{-1}\right),$$

where $\otimes$ denotes the Kronecker product (*KP*), $\mathbf{\Psi}_{n\times n} \otimes \mathbf{\Theta}_{p\times p}$ is the overall precision matrix, and $\mathbf{0}_{np}$ is a column vector of zeros of length np. Kalaitzis *et al.* (2013) proposes to use the Kronecker sum (*KS*) $\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p} = \mathbf{\Psi}_{n\times n} \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \mathbf{\Theta}_{p\times p}$ to structure the overall precision matrix instead.

In a KS-structured matrix normal distribution, for a $p \times n$ random matrix $\mathbf{Y}$, we write

$$\text{vec}\,(\mathbf{Y}) \sim \mathbf{mN}_{np}\left(\mathbf{0}_{np}, \left(\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}\right)^{-1}\right).$$

The KS-structure has several advantages. Firstly, in algebraic graph theory, the Kronecker sum corresponds to the Cartesian product of graphs (Sabidussi, 1959b). A KS-structured model therefore provides intuitive and interpretable results. Secondly, for high-dimensional data, the KS-structure enhances the sparsity of the network, reducing the computation complexity and memory requirements.

4.2.2 Rank-based estimation in a Gaussian graphical model

Both KP-structured and KS-structured matrix normal distributions can only be applied on Gaussian data. To model count data or other non-Gaussian data via a Gaussian graphical model, the Gaussian copula can be applied to transfer these data into a latent Gaussian variable. Liu *et al.* (2012) proposes a semiparametric Gaussian copula for one-way network inference. For a $p \times n$ matrix $\mathbf{Y}$, Liu *et al.* (2012) considers

it as n samples of a $p-$dimensional vector $\left(Y_{1j},\ldots,Y_{pj}\right)$. Liu *et al.* (2012) assumes that there exist functions $f=\{f_i\}_{i=1}^{p}$ such that for $j=1,\ldots,n$:

$$\left(f_1\left(Y_{1j}\right),\ldots,f_p\left(Y_{pj}\right)\right)\sim \mathbf{mN}_p\left(\mathbf{0}_p,\mathbf{\Theta}_{p\times p}^{-1}\right),$$

where $\mathbf{\Theta}_{p\times p}$ is an unknown precision matrix. In this case $\mathbf{Y}_j=\left(Y_{1j},\ldots,Y_{pj}\right)$ is said to follow a nonparanormal multivariate normal distribution, $Y_j\sim \mathbf{NPN}\left(\mathbf{0}_p,\mathbf{\Theta}_{p\times p}^{-1},f\right)$. Then they inferred the precision matrix $\mathbf{\Theta}_{p\times p}$ with the following objective function from *graphical lasso* (Friedman *et al.*, 2008):

$$\min_{\mathbf{\Theta}_{p\times p}}\left\{\operatorname{tr}\left(\mathbf{\Theta}_{p\times p}\mathbf{S}\right)-\ln\left|\mathbf{\Theta}_{p\times p}\right|+\beta\sum_{i_1,i_2}\mathbf{\Theta}_{i_1i_2}\right\},$$

where $\mathbf{S}$ is the empirical covariance matrix of $\left(f_1\left(Y_{1j}\right),\ldots,f_p\left(Y_{pj}\right)\right),\ j=1,\ldots,n$ in *graphical lasso*, $\ln\left|\mathbf{\Theta}_{p\times p}\right|$ is the natural logarithm of the determinant of $\Theta_{p\times p}$, and β is the regularization parameter controlling sparsity. Liu *et al.* (2012) uses the estimated correlation matrix $\hat{\mathbf{S}}$ instead of $\mathbf{S}$, estimated using Kendall's tau or Spearman's rho. In particular, one defines $\Delta_i(j,j')=Y_{ij}-Y_{ij'}$, so that

(Kendall's tau)

$$\hat{\tau}_{i_1i_2}=\frac{2}{n\left(n-1\right)}\sum_{j<j'}\operatorname{sign}\left(\Delta_{i_1}(j,j')\Delta_{i_2}(j,j')\right),$$

(Spearman's rho)

$$\hat{\rho}_{i_1i_2}=\frac{\sum_{j=1}^{n}\left(r_{i_1j}^{(c)}-\bar{r}_j^{(c)}\right)\left(r_{i_2j}^{(c)}-\bar{r}_j^{(c)}\right)}{\sqrt{\sum_{j=1}^{n}\left(r_{i_1j}^{(c)}-\bar{r}_j^{(c)}\right)^2\left(r_{i_2j}^{(c)}-\bar{r}_j^{(c)}\right)^2}},$$

where $r_{ij}^{(c)}$ is the rank of Y_{ij} among $Y_{1j},\ldots,Y_{pj}$ and $\bar{r}_j^{(c)}=\frac{1}{p}\sum_{i=1}^{p}r_{ij}^{(c)}=\frac{1+p}{2}$. Correspondingly,

$$\hat{\mathbf{S}}_{i_1i_2}=\begin{cases}\sin\left(\frac{\pi}{2}\hat{\tau}_{i_1i_2}^{(c)}\right), & i_1\neq i_2,\\ 1, & i_1=i_2.\end{cases}$$

$$\hat{\mathbf{S}}_{i_1i_2}=\begin{cases}2\sin\left(\frac{\pi}{6}\hat{\rho}_{i_1i_2}^{(c)}\right), & i_1\neq i_2,\\ 1, & i_1=i_2.\end{cases}$$

Ning & Liu (2013) extends the matrix-normal distribution with Kronecker product structure to non-Gaussian data with a similar semiparametric approach applied on both the row vectors and the column vectors of $\mathbf{Y}$.

4.2.3 Background on Bigraphical lasso

In this subsection, we introduce some more details on the Bigraphical Lasso method (Kalaitzis *et al.*, 2013) which works on the multivariate model with KS structure.

Let $\mathbf{Y} \in \mathbb{R}^{p\times n}$ be a random matrix. Assume each row of $\mathbf{Y}$ are generated as i.i.d. samples from $\mathbf{mN}\left(\mathbf{0}_n, \mathbf{\Psi}_{n\times n}^{-1}\right)$, then each row of $\mathbf{Y}$ is a Gaussian Markov random field, and the dependency structure in the rows of $\mathbf{Y}$ is a Gaussian Markov random field graph associated with the precision matrix $\mathbf{\Psi}_{n\times n}$. At the same time, assume each column of $\mathbf{Y}$ are generated as i.i.d. samples from $\mathbf{mN}\left(\mathbf{0}_p, \mathbf{\Theta}_{p\times p}^{-1}\right)$, then each column of $\mathbf{Y}$ is a Gaussian Markov random field, and the dependency structure in the columns of $\mathbf{Y}$ is a Gaussian Markov random field graph associated with the precision matrix $\Theta_{p\times p}$. Consider the overall structure in $\mathbf{Y}$ being the Cartesian product of the structure in rows and in columns, then from the discussion in Chapter 2, we have $\text{vec}(\mathbf{Y}) \sim \mathbf{mN}\left(0, \left(\mathbf{\Psi}_{n\times n} \otimes \mathbf{\Theta}_{p\times p}\right)^{-1}\right)$, and the probability density of $\mathbf{Y}$ as

$$\pi(\mathbf{Y}) \propto \exp\left\{-\text{tr}\left(\mathbf{\Psi}_{n\times n}\mathbf{Y}\mathbf{Y}^\top\right) - \text{tr}\left(\mathbf{\Theta}_{p\times p}\mathbf{Y}^\top\mathbf{Y}\right)\right\},$$

with a precision matrix given by the KS:

$$\mathbf{\Omega} = \mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p} = \mathbf{\Psi}_{n\times n} \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \mathbf{\Theta}_{p\times p}.$$

Through this representation we obtain a parameter vector of size $\left(n^2 + p^2\right)$ instead of the usual $\left(n^2 p^2\right)$.

Given data in the form of some design matrix $\mathbf{Y}$, the Bigraphical Lasso model proposed in Kalaitzis *et al.* (2013) estimates the sparse KS-structured inverse covariance of a matrix normal by minimising the ℓ_1-penalized negative likelihood function of $(\mathbf{\Psi}_{n\times n}, \mathbf{\Theta}_{p\times p})$:

$$\min_{\mathbf{\Theta}_{p\times p}, \mathbf{\Psi}_{n\times n}} \left\{ n\text{tr}\left(\mathbf{\Theta}_{p\times p}\mathbf{S}\right) + p\text{tr}\left(\mathbf{\Psi}_{n\times n}\mathbf{T}\right) - \ln|\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}| + \beta_1||\mathbf{\Psi}_{n\times n}||_1 + \beta_2||\mathbf{\Theta}_{p\times p}||_1 \right\}, \tag{4.1}$$

where $\mathbf{S} = \frac{1}{n}\mathbf{Y}^\top\mathbf{Y}$ and $\mathbf{T} = \frac{1}{p}\mathbf{Y}\mathbf{Y}^\top$ are empirical covariance matrices across the samples and features respectively, and β_1 and β_2 are regularization parameters.

From Equation (4.1), we need a method to estimate two graphs simultaneously – one over the columns of $\mathbf{Y}$, corresponding to the sparsity pattern of $\mathbf{\Theta}_{p\times p}$, and another over the rows of $\mathbf{Y}$, corresponding to the sparsity pattern of $\mathbf{\Psi}_{n\times n}$. The original paper

of Kalaitzis *et al.* (2013) proposes a *flip-flop* approach first optimizing over $\mathbf{\Psi}_{n\times n}$, while holding $\mathbf{\Theta}_{p\times p}$ fixed, and then optimizing over $\mathbf{\Theta}_{p\times p}$ while holding $\mathbf{\Psi}_{n\times n}$ fixed. They show that in case of no regularization, the first step of the optimization problem is reduced to

$$\min_{\mathbf{\Psi}_{n\times n}}\left\{p\mathrm{tr}\left(\mathbf{\Psi}_{n\times n}\mathbf{T}\right)-\ln\left|\mathbf{\Psi}_{n\times n}\oplus\mathbf{\Theta}_{p\times p}\right|\right\}.$$

Obtaining the stationary point:

$$\mathbf{T}-\tfrac{1}{2p}\mathbf{T}\circ\mathbf{I}=\tfrac{1}{p}\mathrm{tr}_p\left(\mathbf{W}\right)-\tfrac{1}{2p}\mathrm{tr}_p\left(\mathbf{W}\right)\circ\mathbf{I}\,, \tag{4.2}$$

where $\circ$ is the Hadamard product and we define $\mathbf{W}=\left(\mathbf{\Psi}_{n\times n}\oplus\mathbf{\Theta}_{p\times p}\right)^{-1}$. The block-wise trace $\mathrm{tr}_p\left(\cdot\right)$ is an operator as defined in Definition 4.1.

Definition 4.1 (Kalaitzis *et al.*, 2013) If $\mathbf{M}$ is a $np\times np$ matrix written in terms of n^2 $p\times p$ blocks, as

$$\mathbf{M}=\begin{bmatrix}\mathbf{M}_{11} & \dots & \mathbf{M}_{1n}\\ \vdots & & \vdots\\ \mathbf{M}_{n1} & \dots & \mathbf{M}_{nn}\end{bmatrix},$$

then $\mathrm{tr}_p\left(\mathbf{M}\right)$ is the $n\times n$ matrix of traces of these $p\times p$ blocks:

$$\mathrm{tr}_p\left(\mathbf{M}\right)=\begin{bmatrix}\mathrm{tr}\left(\mathbf{M}_{11}\right) & \dots & \mathrm{tr}\left(\mathbf{M}_{1n}\right)\\ \vdots & & \vdots\\ \mathrm{tr}\left(\mathbf{M}_{n1}\right) & \dots & \mathrm{tr}\left(\mathbf{M}_{\mathbf{nn}}\right)\end{bmatrix}.$$

While the approach proposed in Kalaitzis *et al.* (2013) dramatically reduces the computational complexity from $O(np)$ of naive GLasso (Friedman *et al.*, 2008) to $O(n+p)$, its memory requirements (i.e. space complexity) are prohibitive for problems involving large n or p. Our contribution in Section 4.3 is to give a more efficient solution in terms of computational and space complexity.

4.3 Scalable Bigraphical Lasso Algorithm

Consider the eigen-decomposition of the two precision matrices $\mathbf{\Psi}_{n\times n}=\mathbf{U}\mathbf{\Lambda}_1\mathbf{U}^\top$ and $\mathbf{\Theta}_{p\times p}=\mathbf{V}\mathbf{\Lambda}_2\mathbf{V}^\top$, where $\mathbf{\Lambda}_1\in\mathbb{R}^{n\times n}$ and $\mathbf{\Lambda}_2\in\mathbb{R}^{p\times p}$ are diagonal matrices of eigenvalues and $\mathbf{U}=\left(u_{ij}\right)\in\mathbb{R}^{n\times n}$ and $\mathbf{V}=\left(v_{ij}\right)\in\mathbb{R}^{p\times p}$ are orthogonal eigenvectors matrices

associated with $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$, respectively. It follows that Equation (4.2.3) can be rewritten as

$$\mathbf{\Omega} = (\mathbf{U}\otimes\mathbf{V})\left[\mathbf{\Lambda}_1\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\mathbf{\Lambda}_2\right]\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right). \tag{4.3}$$

Proof of Equation (4.3). Using the Bilinearity and Mixed Product properties of Kronecker product, also noting that $\mathbf{U}\mathbf{U}^\top = \mathbf{I}_n$, $\mathbf{V}\mathbf{V}^\top = \mathbf{I}_p$, we have

$$\begin{aligned}\mathbf{\Omega} &= \mathbf{\Psi}_{n\times n}\oplus\mathbf{\Theta}_{p\times p}\\ &= \mathbf{U}\mathbf{\Lambda}_1\mathbf{U}^\top\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\mathbf{V}\mathbf{\Lambda}_2\mathbf{V}^\top\\ &= (\mathbf{U}\otimes\mathbf{V})\left[\mathbf{\Lambda}_1\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\mathbf{\Lambda}_2\right]\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right).\end{aligned}$$

□

We note that the inverse of a symmetric matrix for which an eigenvalue decomposition is provided is obtained by inverting the eigenvalues,

$$\mathbf{W} = \mathbf{\Omega}^{-1} = (\mathbf{U}\otimes\mathbf{V})\left[\mathbf{\Lambda}_1\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\mathbf{\Lambda}_2\right]^{-1}\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right).$$

Taking

$$\left(\mathbf{I}_n\otimes\mathbf{V}^\top\right)\left(\mathbf{I}_n\otimes\mathbf{I}_p\right) = \mathbf{I}_n\otimes\mathbf{V}^\top,$$

then

$$\mathbf{W}\mathbf{\Omega} = \mathbf{I}_n\otimes\mathbf{I}_p \tag{4.4}$$

can be premultiplied by $\mathbf{I}_n\otimes\mathbf{V}^\top$ to provide

$$\left(\mathbf{I}_n\otimes\mathbf{V}^\top\right)\mathbf{W}\mathbf{\Omega} = \left(\mathbf{U}\otimes\mathbf{I}_p\right)\mathbf{D}\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right)\mathbf{\Omega}, \tag{4.5}$$

where $\mathbf{D} = \left[\mathbf{\Lambda}_1\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\mathbf{\Lambda}_2\right]^{-1}$ is a diagonal matrix.

Proof of Equation (4.5).

$$\begin{aligned}\left(\mathbf{U}\otimes\mathbf{I}_p\right)\mathbf{D}\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right)\mathbf{\Omega} &= \left(\mathbf{U}\otimes\mathbf{I}_p\right)\mathbf{D}\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right)\left(\mathbf{\Psi}_{n\times n}\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\mathbf{V}\mathbf{\Lambda}_2\mathbf{V}^\top\right)\\ &= \left(\mathbf{U}\otimes\mathbf{I}_p\right)\mathbf{D}\left(\mathbf{U}^\top\otimes\mathbf{I}_p\right)\left(\mathbf{I}_n\otimes\mathbf{V}^\top\right)\left(\mathbf{\Psi}_{n\times n}\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\mathbf{V}\mathbf{\Lambda}_2\mathbf{V}^\top\right)\\ &= \left(\mathbf{U}\otimes\mathbf{I}_p\right)\mathbf{D}\left(\mathbf{U}^\top\otimes\mathbf{I}_p\right)\left(\mathbf{\Psi}_{n\times n}\otimes\mathbf{V}^\top + \mathbf{I}_n\otimes\mathbf{\Lambda}_2\mathbf{V}^\top\right)\\ &= \mathbf{I}_n\otimes\mathbf{V}^\top\\ &= \mathbf{I}_n\otimes\mathbf{V}^\top\mathbf{W}\mathbf{\Omega}.\end{aligned}$$

□

If we multiply both sides of Equation (4.5) by $\mathbf{I}_n \otimes \mathbf{V}$, we have

$$\mathbf{I}_n \otimes \mathbf{I}_p = \left(\mathbf{U} \otimes \mathbf{I}_p\right) \mathbf{D} \left(\mathbf{U}^\top \otimes \mathbf{I}_p\right) \left(\mathbf{\Psi}_{n\times n} \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \mathbf{\Lambda}_2\right), \tag{4.6}$$

Proof of Equation (4.6). Note that $\mathbf{W\Omega} = \mathbf{I}_{np}$, therefore we can write Equation (4.5) as:

$$\left(\mathbf{I}_n \otimes \mathbf{V}^\top\right) \mathbf{W\Omega} = \left(\mathbf{U} \otimes \mathbf{I}_p\right) \mathbf{D} \left(\mathbf{U}^\top \otimes \mathbf{V}^\top\right) \mathbf{\Omega}.$$

If we multiply both sides of the equation above by $\mathbf{I}_n \otimes \mathbf{V}$:

$$\left(\mathbf{I}_n \otimes \mathbf{V}^\top\right) \mathbf{W\Omega} \left(\mathbf{I}_n \otimes \mathbf{V}\right) = \left(\mathbf{U} \otimes \mathbf{I}_p\right) \mathbf{D} \left(\mathbf{U}^\top \otimes \mathbf{V}^\top\right) \mathbf{\Omega} \left(\mathbf{I}_n \otimes \mathbf{V}\right).$$

From the right-hand side, we get $\left(\mathbf{U} \otimes \mathbf{I}_p\right) \mathbf{D} \left(\mathbf{U}^\top \otimes \mathbf{I}_p\right) \left(\mathbf{\Psi}_{n\times n} \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \mathbf{\Lambda}_2\right)$. On the left-hand side, remember that $\mathbf{W\Omega} = \mathbf{I}_{np}$, so

$$\left(\mathbf{I}_n \otimes \mathbf{V}^\top\right) \mathbf{W\Omega} \left(\mathbf{I}_n \otimes \mathbf{V}\right) = \mathbf{I}_n \otimes \mathbf{I}_p.$$

□

Equation (4.6) can be rewritten in a similar form as Equation (4.4)

$$\hat{\mathbf{W}}\hat{\mathbf{\Omega}} = \mathbf{I}_n \otimes \mathbf{I}_p,$$

where

$$\hat{\mathbf{W}} = \left[\mathbf{U} \otimes \mathbf{I}_p\right] \mathbf{D} \left[\mathbf{U}^\top \otimes \mathbf{I}_p\right]$$

and

$$\hat{\mathbf{\Omega}} = \mathbf{\Psi}_{n\times n} \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \mathbf{\Lambda}_2.$$

We partition $\hat{\mathbf{W}}$ and $\hat{\mathbf{\Omega}}$ into blocks

$$\hat{\mathbf{W}} = \begin{bmatrix} \hat{\mathbf{W}}_{11} & \hat{\mathbf{W}}_{1\setminus 1} \\ \hat{\mathbf{W}}_{\setminus 11} & \hat{\mathbf{W}}_{\setminus 1\setminus 1} \end{bmatrix},$$

$$\hat{\mathbf{\Omega}} = \begin{bmatrix} \hat{\mathbf{\Omega}}_{11} & \hat{\mathbf{\Omega}}_{1\setminus 1} \\ \hat{\mathbf{\Omega}}_{\setminus 11} & \hat{\mathbf{\Omega}}_{\setminus 1\setminus 1} \end{bmatrix},$$

where $\hat{\mathbf{W}}_{11}$ and $\hat{\mathbf{\Omega}}_{11}$ are $p \times p$ matrices and $\hat{\mathbf{W}}_{\setminus 11}$ and $\hat{\mathbf{\Omega}}_{\setminus 11}$ are $p\,(n-1) \times p$ matrices. Then from the bottom-left block of

$$\hat{\mathbf{W}}\hat{\mathbf{\Omega}} = \hat{\mathbf{W}}\left(\mathbf{\Psi}_{n\times n} \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \mathbf{\Lambda}_2\right) = \mathbf{I}_n \otimes \mathbf{I}_p, \tag{4.7}$$

we get

$$\hat{\mathbf{W}}_{\backslash 11}\left(\psi_{11}\mathbf{I}_p+\mathbf{\Lambda}_2\right)+\hat{\mathbf{W}}_{\backslash 1\backslash 1}\left(\boldsymbol{\psi}_{\backslash 11}\otimes\mathbf{I}_p\right)=\mathbf{0}_{n-1}\otimes\mathbf{I}_p,$$

where we use the notation $\mathbf{\Psi}_{n\times n}=\left(\psi_{ij}\right)_{i,j=1,\ldots,n}$ and $\boldsymbol{\psi}_{\backslash 11}$ representing the corresponding sub-block. Post multiplying both sides of the last equation by $\left(\psi_{11}\mathbf{I}_p+\mathbf{\Lambda}_2\right)^{-1}$ we have

$$\hat{\mathbf{W}}_{\backslash 11}+\hat{\mathbf{W}}_{\backslash 1\backslash 1}\begin{bmatrix}\left(\psi_{11}\mathbf{I}_p+\mathbf{\Lambda}_2\right)^{-1}\psi_{21}\\ \vdots\\ \left(\psi_{11}\mathbf{I}_p+\mathbf{\Lambda}_2\right)^{-1}\psi_{n1}\end{bmatrix}=\mathbf{0}_{n-1}\otimes\mathbf{I}_p. \tag{4.8}$$

Proof of Equation (4.8). In order to prove Equation (4.8), we first note that, from the bottom-left block of

$$\hat{\mathbf{W}}\hat{\mathbf{\Omega}}=\begin{bmatrix}\hat{\mathbf{W}}_{11} & \hat{\mathbf{W}}_{1\backslash 1}\\ \hat{\mathbf{W}}_{\backslash 11} & \hat{\mathbf{W}}_{\backslash 1\backslash 1}\end{bmatrix}\begin{bmatrix}\psi_{11}\mathbf{I}_p+\mathbf{\Lambda}_2 & \ldots & \psi_{1n}\mathbf{I}_p\\ \vdots & \ddots & \vdots\\ \psi_{n1}\mathbf{I}_p & \ldots & \psi_{nn}\mathbf{I}_p+\mathbf{\Lambda}_2\end{bmatrix}=\mathbf{I}_n\otimes\mathbf{I}_p$$

we get

$$\hat{\mathbf{W}}_{\backslash 11}\hat{\mathbf{\Omega}}_{11}+\hat{\mathbf{W}}_{\backslash 1\backslash 1}\hat{\mathbf{\Omega}}_{\backslash 11}=\hat{\mathbf{W}}_{\backslash 11}\left(\psi_{11}\mathbf{I}_p+\mathbf{\Lambda}_2\right)+\hat{\mathbf{W}}_{\backslash 1\backslash 1}\left(\boldsymbol{\psi}_{\backslash 11}\otimes\mathbf{I}_p\right)=\mathbf{0}_{n-1}\otimes\mathbf{I}_p.$$

Thus, multiplying both sides of the last equation by $\left(\psi_{11}\mathbf{I}_p+\mathbf{\Lambda}_2\right)^{-1}$, one has

$$\hat{\mathbf{W}}_{\backslash 11}+\hat{\mathbf{W}}_{\backslash 1\backslash 1}\begin{bmatrix}\left(\psi_{11}\mathbf{I}_p+\mathbf{\Lambda}_2\right)^{-1}\psi_{21}\\ \vdots\\ \left(\psi_{11}\mathbf{I}_p+\mathbf{\Lambda}_2\right)^{-1}\psi_{n1}\end{bmatrix}=\mathbf{0}_{n-1}\otimes\mathbf{I}_p.$$

□

Decomposing $\hat{\mathbf{W}}_{\backslash 1\backslash 1}$ in $(n-1)$ adjacent blocks $\hat{\mathbf{W}}_{\backslash 1k}\in\mathbb{R}^{(n-1)p\times p}$, $\forall k\in\{2,\ldots,n\}$, then Equation (4.8) can be rewritten as

$$\hat{\mathbf{W}}_{\backslash 11}+\hat{\mathbf{W}}_{\backslash 12}\left(\psi_{11}\mathbf{I}_p+\mathbf{\Lambda}_2\right)^{-1}\psi_{21}+\cdots+\hat{\mathbf{W}}_{\backslash 1n}\left(\psi_{11}\mathbf{I}_p+\mathbf{\Lambda}_2\right)^{-1}\psi_{n1}=\mathbf{0}_{n-1}\otimes\mathbf{I}_p.$$

Proposition 4.1 Following the assumptions and calculations above we have

$$\mathrm{tr}_p\left(\mathbf{W}\right)=\mathrm{tr}_p\left(\hat{\mathbf{W}}\right).$$

Proof of Proposition 4.1. Proposition 4.1 follows from the fact that

$$\left[\mathbf{I}_n \otimes \mathbf{V}^\top\right] \mathbf{W} \left[\mathbf{I}_n \otimes \mathbf{V}\right] = \left[\mathbf{U} \otimes \mathbf{I}_p\right] \mathbf{D} \left[\mathbf{U}^\top \otimes \mathbf{I}_p\right] = \hat{\mathbf{W}}.$$

Then, the $p \times p$ blocks of $\mathbf{W}$ and $\hat{\mathbf{W}}$ hold a similarity relation:

$$\hat{\mathbf{W}}_{ij} = \mathbf{V}^\top \mathbf{W}_{ij} \mathbf{V},$$

and hence $\mathrm{tr}_p(\mathbf{W}) = \mathrm{tr}_p\left(\hat{\mathbf{W}}\right)$. □

Proposition 4.1 enables us to make use of the stationary point given in Equation (4.2). As described in Kalaitzis *et al.* (2013), we can partition the empirical covariance $\mathbf{T}$ as

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_{11} & \mathbf{t}_{1\setminus 1} \\ \mathbf{t}_{\setminus 11} & \mathbf{T}_{\setminus 1\setminus 1} \end{bmatrix},$$

where $\mathbf{t}_{\setminus 11} \in \mathbb{R}^{n-1}$ and $\mathbf{T}_{\setminus 1\setminus 1} \in \mathbb{R}^{(n-1)\times(n-1)}$. In particular, from the lower left block of (4.2) we get

$$\mathbf{t}_{\setminus 11} = \frac{1}{p} \mathrm{tr}_p\left(\mathbf{W}_{\setminus 11}\right).$$

Taking the block-wise trace $\mathrm{tr}_p(\cdot)$ of both sides of (4.8), gives

$$p\mathbf{t}_{\setminus 11} + \mathbf{A}_{\setminus 1\setminus 1} \boldsymbol{\psi}_{\setminus 11} = \mathbf{0}_{n-1}, \tag{4.9}$$

where $\mathbf{A}^\top_{\setminus 1\setminus 1} \in \mathbb{R}^{(n-1)\times(n-1)}$ is:

$$\mathbf{A}^\top_{\setminus 1\setminus 1} \triangleq \begin{bmatrix} \mathrm{tr}_p\left\{\hat{\mathbf{W}}_{\setminus 12}\left(\psi_{11}\mathbf{I}_p + \boldsymbol{\Lambda}_2\right)^{-1}\right\}^\top \\ \vdots \\ \mathrm{tr}_p\left\{\hat{\mathbf{W}}_{\setminus 1n}\left(\psi_{11}\mathbf{I}_p + \boldsymbol{\Lambda}_2\right)^{-1}\right\}^\top \end{bmatrix}. \tag{4.10}$$

The problem posed in Equation (4.9) is addressed via a lasso regression. In Proposition 4.2, we use some of the previous decomposition in order to reduce the computational complexity of the problem.

Proposition 4.2 Following the assumptions and calculations above we have

$$\mathrm{tr}_p\left\{\hat{\mathbf{W}}_{\setminus 1k}\left(\psi_{11}\mathbf{I}_p + \boldsymbol{\Lambda}_2\right)^{-1}\right\} = \sum_{j=1}^{p} \frac{1}{\psi_{11} + \lambda_{2j}} \begin{bmatrix} \sum_{i=1}^{n} \frac{u_{2i}u_{ki}}{\lambda_{1i}+\lambda_{21}} \\ \vdots \\ \sum_{i=1}^{n} \frac{u_{ni}u_{ki}}{\lambda_{1i}+\lambda_{2p}} \end{bmatrix},$$

where $\lambda_{11} \ldots \lambda_{1n}$ and $\lambda_{21} \ldots \lambda_{2p}$ are the diagonal values of $\boldsymbol{\Lambda}_1 \in \mathbb{R}^{n\times n}$ and $\boldsymbol{\Lambda}_2 \in \mathbb{R}^{p\times p}$, respectively.

Proof of Proposition 4.2. To prove Proposition 4.2, we note that

$$\hat{\mathbf{W}}_{\setminus 1\setminus 1} = \left[\mathbf{U}_{\setminus 1} \otimes \mathbf{I}_p\right] \mathbf{D} \left[\mathbf{U}_{\setminus 1}^\top \otimes \mathbf{I}_p\right] = \begin{bmatrix} u_{21}\mathbf{I}_p & \dots & u_{2n}\mathbf{I}_p \\ \vdots & \ddots & \vdots \\ u_{n1}\mathbf{I}_p & \dots & u_{nn}\mathbf{I}_p \end{bmatrix} \mathbf{D} \begin{bmatrix} u_{21}\mathbf{I}_p & \dots & u_{n1}\mathbf{I}_p \\ \vdots & \ddots & \vdots \\ u_{2n}\mathbf{I}_p & \dots & u_{nn}\mathbf{I}_p \end{bmatrix},$$

where $\mathbf{U}_{\setminus 1} \in \mathbb{R}^{(n-1)\times n}$ is the matrix formed by the last $n-1$ rows of $\mathbf{U}$. Then, we can decompose $\hat{\mathbf{W}}_{\setminus 1\setminus 1}$ in $(n-1)\times(n-1)$ blocks $[\hat{\mathbf{W}}_{\setminus 1\setminus 1}]_{\ell,k} \in \mathbb{R}^{p\times p}$, with

$$[\hat{\mathbf{W}}_{\setminus 1\setminus 1}]_{\ell,k} = \begin{bmatrix} \sum_{i=1}^n \frac{u_{\ell i}u_{ki}}{\lambda_{1i}+\lambda_{21}} & \dots & 0 \\ 0 & \dots & \sum_{i=1}^n \frac{u_{\ell i}u_{ki}}{\lambda_{1i}+\lambda_{2p}} \end{bmatrix}, \quad \ell, k \in \{2,\dots,n\}.$$

This formulation allows us to write each trace term of Equation (4.9) as

$$\operatorname{tr}_p\left\{\hat{\mathbf{W}}_{\setminus 1k}\left(\psi_{11}\mathbf{I}_p + \mathbf{\Lambda}_2\right)^{-1}\right\} = \begin{bmatrix} \operatorname{tr}\left\{\hat{\mathbf{W}}_{\setminus 1\setminus 1}\right\}_{1,k}\left(\psi_{11}\mathbf{I}_p + \mathbf{\Lambda}_2\right)^{-1} \\ \vdots \\ \operatorname{tr}\left\{\hat{\mathbf{W}}_{\setminus 1\setminus 1}\right\}_{(n-1),k}\left(\psi_{11}\mathbf{I}_p + \mathbf{\Lambda}_2\right)^{-1} \end{bmatrix}, \quad k \in \{1,\dots,n-1\},$$

More explicitly,

$$\begin{aligned} \operatorname{tr}_p\left\{\hat{\mathbf{W}}_{\setminus 1k}\left(\psi_{11}\mathbf{I}_p + \mathbf{\Lambda}_2\right)^{-1}\right\} &= \begin{bmatrix} \sum_{j=1}^p \sum_{i=1}^n \frac{1}{\psi_{11}+\lambda_{2j}} \frac{u_{2i}u_{ki}}{\lambda_{1i}+\lambda_{2j}} \\ \vdots \\ \sum_{j=1}^p \sum_{i=1}^n \frac{1}{\psi_{11}+\lambda_{2j}} \frac{u_{ni}u_{ki}}{\lambda_{1i}+\lambda_{2j}} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^p \frac{1}{\psi_{11}+\lambda_{2j}} \sum_{i=1}^n \frac{u_{2i}u_{ki}}{\lambda_{1i}+\lambda_{2j}} \\ \vdots \\ \sum_{j=1}^p \frac{1}{\psi_{11}+\lambda_{2j}} \sum_{i=1}^n \frac{u_{ni}u_{ki}}{\lambda_{1i}+\lambda_{2j}} \end{bmatrix} \\ &= \sum_{j=1}^p \frac{1}{\psi_{11}+\lambda_{2j}} \begin{bmatrix} \sum_{i=1}^n \frac{u_{2i}u_{ki}}{\lambda_{1i}+\lambda_{2j}} \\ \vdots \\ \sum_{i=1}^n \frac{u_{ni}u_{ki}}{\lambda_{1i}+\lambda_{2p}} \end{bmatrix}. \end{aligned}$$

□

We note that by imposing an ℓ_1 penalty on $\boldsymbol{\psi}_{\setminus 11}$, the problem posed in (4.9) reduces to a lasso regression involving now only the matrix $\mathbf{U}$, the diagonal of $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$, and ψ_{11}. This decomposition frees the prohibitive amount of memory needed to store the matrix $\hat{\mathbf{W}}$, which is of size n^2p^2.

The lasso regression will provide an estimation on the first column of $\mathbf{\Psi}_{n\times n}$. For the update of all the other columns $\boldsymbol{\psi}_{\setminus ii}$ we need to reiterate the same approach. Indeed we partition $\mathbf{\Psi}_{n\times n}$ into $\psi_{ii}, \boldsymbol{\psi}_{\setminus ii}, \boldsymbol{\psi}_{i\setminus i}$ and $\mathbf{\Psi}_{\setminus i\setminus i}$ for $i = 1,\dots,n$. We then find a sparse solution of $p\mathbf{t}_{\setminus ii} + \mathbf{A}_{\setminus i\setminus i}\boldsymbol{\psi}_{\setminus ii} = \mathbf{0}_{n-1}$ with lasso regression. Given the new

value $\boldsymbol{\psi}_{\setminus ii}$, we then compute the eigenvalues matrix $\boldsymbol{\Lambda}_1$ and eigenvectors matrix $\mathbf{U}$ of $\boldsymbol{\Psi}_{n\times n}$. This will provide the updated values to be used in Proposition 4.2. Hence, after n steps, the columns of $\boldsymbol{\Psi}_{n\times n}$ are estimated. Similarly the estimation of $\boldsymbol{\Theta}_{p\times p}$, conditionally on fixed $\boldsymbol{\Psi}_{n\times n}$, becomes directly analogous to the above simply by *transposing* the design matrix (samples become features and vice-versa) and is obtained in p steps.

Our approach is summarised in Algorithm 5 for Gaussian data. We point out that the convergence of Algorithm 5 could also be directly verified on the value of the objective function (4.1) at each step, but, due to the computation of $\left|\boldsymbol{\Psi}_{n\times n}\oplus\boldsymbol{\Theta}_{p\times p}\right|$, when $p,n >> 100$ this becomes unfeasible. Indeed, the space complexity can be reduced from $O\left(n^2p^2\right)$ to $O\left(n^2+p^2\right)$ by means of Proposition 4.3.

Algorithm 5 scBiGLasso

Input: Maximum iteration number N, tolerance ε, M many observations of $p\times n$ matrices $\mathbf{Y}_m$, $m=1,\ldots,M$. β_1,β_2, initial estimates of $\boldsymbol{\Psi}_{n\times n}$ and $\boldsymbol{\Theta}_{p\times p}$, $\boldsymbol{\Psi}^{(0)}_{n\times n}$ and $\boldsymbol{\Theta}^{(0)}_{p\times p}$.

For each $\mathbf{Y}_m$, $\mathbf{T}^{(m)} \leftarrow p^{-1}\mathbf{Y}_m\mathbf{Y}_m^\top$.

$\mathbf{T} \leftarrow \frac{1}{M}\sum_{m=1}^{M}\mathbf{T}^{(m)}$.

repeat

 # Estimate $\boldsymbol{\Psi}_{n\times n}$:

for iteration $\tau = 1,\ldots,N$ **do**

Decompose $\boldsymbol{\Psi}^{(\tau-1)}_{n\times n} = \mathbf{U}^{(\tau-1)}\boldsymbol{\Lambda}_1^{(\tau-1)}\mathbf{U}^{(\tau-1)\top}$ and $\boldsymbol{\Theta}^{(\tau-1)}_{p\times p} = \mathbf{V}^{(\tau-1)}\boldsymbol{\Lambda}_2^{(\tau-1)}\mathbf{V}^{(\tau-1)\top}$.

for $i = 1,\ldots,n$ **do**

Partition $\boldsymbol{\Psi}^{(\tau-1)}_{n\times n}$ into $\psi^{(\tau-1)}_{ii}$, $\boldsymbol{\psi}^{(\tau-1)}_{i\setminus i}$, $\boldsymbol{\psi}^{(\tau-1)}_{\setminus ii}$ and $\boldsymbol{\Psi}^{(\tau-1)}_{\setminus i\setminus i}$.

Calculate $\mathbf{A}^{(\tau-1)}_{\setminus i\setminus i}$ similar to Equation (4.10) and Proposition 4.2

with $\psi^{(\tau-1)}_{ii}$, $\mathbf{U}^{(\tau-1)}$, $\boldsymbol{\Lambda}_1^{(\tau-1)}$ and $\boldsymbol{\Lambda}_1^{(\tau-1)}$.

With Lasso regression (Friedman *et al.*, 2008), find a sparse solution, $\boldsymbol{\psi}^{*}_{i\setminus i}$, for

$$p\mathbf{t}_{i\setminus i} + \mathbf{A}^{(\tau-1)}_{\setminus i\setminus i}\boldsymbol{\psi}^{(\tau)}_{i\setminus i} = \mathbf{0}_{n-1}.$$

Calculate the direction vector from
$\boldsymbol{\psi}_{i\setminus i}^{(\tau-1)}$ to $\boldsymbol{\psi}_{i\setminus i}^{*}$: $\Delta\boldsymbol{\psi}_{i\setminus i}^{(\tau)} = \boldsymbol{\psi}_{i\setminus i}^{*} - \boldsymbol{\psi}_{i\setminus i}^{(\tau-1)}$.
Since the objective of solving $p\mathbf{t}_{i\setminus i} + \mathbf{A}_{\setminus i\setminus i}^{(\tau-1)}\boldsymbol{\psi}_{i\setminus i}^{(\tau)} = \mathbf{0}_{n-1}$
can be written as $f\left(\boldsymbol{\psi}_{i\setminus i}^{(\tau)}\right) = \left\|\mathbf{t}_{i\setminus i} + \frac{1}{p}\mathbf{A}_{\setminus i\setminus i}^{(\tau-1)}\boldsymbol{\psi}_{i\setminus i}^{(\tau)}\right\|_F^2$,
Calculate $\nabla f\left(\boldsymbol{\psi}_{i\setminus i}\right) = 2\frac{1}{p^2}\mathbf{A}_{\setminus i\setminus i}^{(\tau-1)\top}\mathbf{A}_{\setminus i\setminus i}^{(\tau-1)}\boldsymbol{\psi}_{i\setminus i}^{(\tau)} + 2\frac{1}{p}\mathbf{A}_{\setminus i\setminus i}^{(\tau-1)\top}\mathbf{t}_{i\setminus i}$.
Take $\zeta = \min\{\lambda_{11},\ldots,\lambda_{1n}\}$.
 # Implement FISTA (Beck & Teboulle, 2009) with backtracking line search.
Calculate $Q = f\left(\boldsymbol{\psi}_{i\setminus i}^{(\tau-1)}\right) + \nabla f\left(\boldsymbol{\psi}_{i\setminus i}^{(\tau-1)}\right)^{\top}\Delta\boldsymbol{\psi}_{i\setminus i}^{(\tau)} + \frac{1}{2\zeta}\left\|\Delta\boldsymbol{\psi}_{i\setminus i}^{(\tau)}\right\|_F^2$ and $f(\boldsymbol{\psi}_{i\setminus i}^{*})$.
$t_0 = 1, a = 0$.
while $f\left(\boldsymbol{\psi}_{i\setminus i}^{*}\right) > Q$ **do**

$$a = a+1,\ t_{a+1} = \frac{1+\sqrt{1+4t_a^2}}{2}.$$

$$\boldsymbol{\psi}_{i\setminus i}^{*} = \boldsymbol{\psi}_{i\setminus i}^{(\tau-1)} + \frac{t_a - 1}{t_{a+1}}\Delta\boldsymbol{\psi}_{i\setminus i}^{(\tau)}.$$

$$\Delta\boldsymbol{\psi}_{i\setminus i}^{(\tau)} = \boldsymbol{\psi}_{i\setminus i}^{*} - \boldsymbol{\psi}_{i\setminus i}^{(\tau-1)}.$$

Calculate $Q = f\left(\boldsymbol{\psi}_{i\setminus i}^{(\tau-1)}\right) + \nabla f\left(\boldsymbol{\psi}_{i\setminus i}^{(\tau-1)}\right)^{\top}\Delta\boldsymbol{\psi}_{i\setminus i}^{(\tau)} + \frac{1}{2\zeta}\left\|\Delta\boldsymbol{\psi}_{i\setminus i}^{(\tau)}\right\|_F^2$ and $f(\boldsymbol{\psi}_{i\setminus i}^{*})$.
end while
Update the non-diagonal column $\boldsymbol{\psi}_{i\setminus i}^{(\tau)} = \boldsymbol{\psi}_{i\setminus i}^{*}$.
end for
 # Estimate $\boldsymbol{\Theta}_{p\times p}$:
Proceed as if estimating $\boldsymbol{\Psi}_{n\times n}$ with input $\mathbf{Y}^{\top}, \beta_1, \beta_2$.
Calculate the change in the estimated matrices from each iteration

$$\Delta\boldsymbol{\Psi}^{(\tau)} = \|\boldsymbol{\Psi}_{n\times n}^{(\tau)} - \boldsymbol{\Psi}_{n\times n}^{(\tau-1)}\|_F^2,$$

$$\Delta\boldsymbol{\Theta}^{(\tau)} = \|\boldsymbol{\Theta}_{p\times p}^{(\tau)} - \boldsymbol{\Theta}_{p\times p}^{(\tau-1)}\|_F^2.$$

end for
until Maximum iteration number reached, or
$\max_{\tau^*=\tau-2,\tau-1,\tau}\left\{(\Delta\boldsymbol{\Psi}^{(\tau^*)} + \Delta\boldsymbol{\Theta}^{(\tau^*)})\right\} < \varepsilon$, for $\tau \geq 3$.

Proposition 4.3 Following the assumptions and calculations above we have

$$|\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}| = \prod_{i=1}^{n}\prod_{j=1}^{p}\left(\lambda_{1i}+\lambda_{2j}\right).$$

Proof of Proposition 4.3. Proposition 4.3 follows from the fact that

$$\mathbf{W} = \mathbf{\Omega}^{-1} = (\mathbf{U}\otimes\mathbf{V})\left[\mathbf{\Lambda}_1\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\mathbf{\Lambda}_2\right]^{-1}\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right),$$

and

$$\mathbf{D} = \begin{bmatrix} \frac{1}{\lambda_{11}+\lambda_{21}} & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & \frac{1}{\lambda_{11}+\lambda_{2p}} & \dots & 0 & \dots & 0 \\ \vdots & \dots & \vdots & \ddots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & \dots & \frac{1}{\lambda_{1n}+\lambda_{21}} & \dots & 0 \\ \vdots & \dots & \vdots & \dots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & \frac{1}{\lambda_{1n}+\lambda_{2p}} \end{bmatrix},$$

where $\lambda_{11}\dots\lambda_{1n}$ are the diagonal values of $\mathbf{\Lambda}_1 \in \mathbb{R}^{n\times n}$ and $\lambda_{21}\dots\lambda_{2p}$ are the diagonal values of $\mathbf{\Lambda}_2 \in \mathbb{R}^{p\times p}$. Then, we can write

$$\begin{aligned}|\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}| &= |\,(\mathbf{U}\otimes\mathbf{V})\,\mathbf{D}^{-1}\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right)| = |\mathbf{U}\otimes\mathbf{V}|^2|\mathbf{D}^{-1}| = |\mathbf{U}|^{2p}|\mathbf{V}|^{2n}\prod_{i=1}^{n}\prod_{j=1}^{p}\left(\lambda_{1i}+\lambda_{2j}\right)\\ &= \prod_{i=1}^{n}\prod_{j=1}^{p}\left(\lambda_{1i}+\lambda_{2j}\right).\end{aligned}$$

□

It follows that:

$$\ln|\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}| = \sum_{i=1}^{n}\sum_{j=1}^{p}\ln|\lambda_{1i}+\lambda_{2j}| = C.$$

Hence we can write the objective function as

$$\min_{\mathbf{\Theta}_{p\times p},\mathbf{\Psi}_{n\times n}}\left\{n\mathrm{tr}\left(\mathbf{\Theta}_{p\times p}\mathbf{S}\right) + p\mathrm{tr}\left(\mathbf{\Psi}_{n\times n}\mathbf{T}\right) - C + \beta_1||\mathbf{\Psi}_{n\times n}||_1 + \beta_2||\mathbf{\Theta}_{p\times p}||_1\right\}.$$

Note that this scalable version of the Bigraphical Lasso is able to deal with higher dimensional problems. This is mainly due to the fact that in our implementation

there is no need to directly evaluate the matrix $\mathbf{W}$. Instead we just need the eigen-decomposition of the two precision matrices $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$. In the original paper Kalaitzis *et al.* (2013) at each step i the blocks of $\mathbf{W}$ are explicitly updated and of course were involved in the next step of the estimation. In particular $\mathbf{W}_{\setminus ii}$ is computed via backward-substitution in Equation (4.8) and W_{11} via backward-substitution in Equation (4.7).

In summary, as we are not interested in the estimation of the overall $\hat{\mathbf{W}}$ nor $\mathbf{\Omega}$, we will never explicitly update them, but we will rather focus on the estimation of $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$. This leads to a space complexity reduction from $O(n^2p^2)$ to $O(n^2+p^2)$ by means of Proposition 4.2 and Proposition 4.3.

Our Scalable Bigraphical Lasso algorithm (ScB) benefits from the same statistical convergence properties embedded in the original Bigraphical Lasso model (Kalaitzis *et al.*, 2013). Greenewald *et al.* (2019) gives the statistical convergence rates (Greenewald *et al.*, 2019, Theorems 1-3) (Greenewald *et al.*, 2019, Lemma 19, Supplementary Material) of the Bigraphical Lasso model and its generalisation for K-way tensor-valued data.

4.4 Nonparanormal Bigraphical Lasso Model

The method in Section 4.3 only deals with Gaussian data, while in real world many data come in the form of count data. In this section, we introduce a Gaussian copula based method to adapt Algorithm 5 for count data. We start by introducing the definition of the matrix nonparanormal distribution with a Kronecker sum structure.

Definition 4.2 Consider a $p\times n$ non-Gaussian data matrix $\mathbf{Y}$. $\mathbf{Y}$ follows a matrix nonparanormal distribution with a Kronecker sum structure $\mathbf{MNPN}_{KS}\left(\mathbf{M};\mathbf{\Psi}_{n\times n}^{-1},\mathbf{\Theta}_{p\times p}^{-1};f\right)$, with mean matrix $\mathbf{M}$, and where $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$ are the row-specific and the column-specific precision matrices, if and only if there exists a set of monotonic transformations $f=\{f_{ij}\}_{i=1,\dots,p}^{j=1,\dots,n}$ such that

$$\operatorname{vec}\big(f(\mathbf{Y})\big)\sim\mathbf{mN}\Big(\operatorname{vec}(\mathbf{M}),\big(\mathbf{\Psi}_{n\times n}\oplus\mathbf{\Theta}_{p\times p}\big)^{-1}\Big).$$

In this chapter, we only consider the model after centering, i.e $\operatorname{vec}(\mathbf{M})=\mathbf{0}_{np}$. The choices $f_{ij}\left(Y_{ij}\right)=Y_{ij}$ and $f_{ij}\left(Y_{ij}\right)=\ln Y_{ij}$ give us multivariate Normal distribution

and multivariate log-Normal distribution respectively. Since we only require f to be monotone, this model provides us with a wider family of distributions to work on, thus extends the Bigraphical model to non-Gaussian data. We note that the model in Definition 4.2 can be viewed as a latent model, with latent variable $\mathbf{Z} = f(\mathbf{Y})$ and $\text{vec}(\mathbf{Z}) \sim \mathbf{mN}\left(\mathbf{0}_{np}, \left(\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}\right)^{-1}\right)$.

Following the arguments in Kalaitzis *et al.* (2013) and Greenewald *et al.* (2019), the supports of $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$ encode the dependence structure of the row variables and the column variables, respectively. Following the discussion in Section 2.2, $\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}$ represents the Cartesian product of the Gaussian Markov random field graphs corresponding to the rows and the columns. In the next section, we introduce a method to infer the nonparanormal distribution without explicitly defining f.

4.4.1 Estimation of the precision matrices

We now consider the estimation of the precision matrices $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$. Like the lasso methods applied in one-way network inference and in Gaussian Bigraphical models, we enforce sparsity on $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$ by regularization on the negative log-likelihood, which gives us the objective function:

$$\min_{\mathbf{\Psi}_{n\times n}, \mathbf{\Theta}_{p\times p}} \left\{ -\ln|\,\mathbf{\Omega}\,| + p\,\text{tr}\left(\mathbf{\Psi}_{n\times n}\mathbf{T}\right) + n\,\text{tr}\left(\mathbf{\Theta}_{p\times p}\mathbf{S}\right) + \beta_1 \|\mathbf{\Psi}_{n\times n}\|_1 + \beta_2 \|\mathbf{\Theta}_{p\times p}\|_1 \right\},$$

where $\mathbf{T} = \frac{1}{p}\left(\mathbf{Z}\mathbf{Z}^\top\right)$ is the empirical covariance matrix along the rows, and $\mathbf{S} = \frac{1}{n}\left(\mathbf{Z}^\top\mathbf{Z}\right)$ is the empirical covariance matrix along the columns. The only problem that remains now is to estimate the empirical covariance matrices $\mathbf{T}$ and $\mathbf{S}$. When estimating one-way network, Liu *et al.* (2012) proposed the nonparanormal skeptic, exploiting Kendall's tau or Spearman's rho, without explicitly calculating the marginal transforming function f. Similarly, we define Kendall's tau and Spearman's rho along rows and columns. More specifically, let $r_{ij}^{(c)}$ be the rank of Y_{ij} among $Y_{1j}, \ldots, Y_{pj}$ and $\bar{r}_j^{(c)} = \frac{1}{p}\sum_{i=1}^{p} r_{ij} = \frac{p+1}{2}$. Define $\Delta_i(j, j') = Y_{ij} - Y_{ij'}$. We consider the following statistics:

(Column-wise Kendall's tau)

$$\hat{\tau}_{i_1 i_2}^{(c)} = \frac{2}{n(n-1)} \sum_{j<j'} \text{sign}\left(\Delta_{i_1}(j, j')\Delta_{i_2}(j, j')\right)$$

(Column-wise Spearman's rho)

$$\hat{\rho}^{(c)}_{i_1 i_2} = \frac{\sum_{j=1}^{n}\left(r^{(c)}_{i_1 j} - \bar{r}^{(c)}_{j}\right)\left(r^{(c)}_{i_2 j} - \bar{r}^{(c)}_{j}\right)}{\sqrt{\sum_{j=1}^{n}\left(r^{(c)}_{i_1 j} - \bar{r}^{(c)}_{j}\right)^2\left(r^{(c)}_{i_2 j} - \bar{r}^{(c)}_{j}\right)^2}}.$$

Similarly, let $r^{(r)}_{ij}$ be the rank of Y_{ij} among $Y_{i1},\ldots,Y_{in}$ and $\bar{r}^{(r)}_{i} = \frac{1}{n}\sum_{j=1}^{n} r_{ij} = \frac{n+1}{2}$. Define $\Delta_j(i,i') = Y_{ij} - Y_{i'j}$. We consider the following statistics:

(Row-wise Kendall's tau)

$$\hat{\tau}^{(r)}_{j_1 j_2} = \frac{2}{p(p-1)}\sum_{i<i'}\operatorname{sign}\left(\Delta_{j_1}(i,i')\Delta_{j_2}(i,i')\right),$$

(Row-wise Spearman's rho)

$$\hat{\rho}^{(r)}_{j_1 j_2} = \frac{\sum_{i=1}^{p}\left(r^{(r)}_{i j_1} - \bar{r}^{(r)}_{i}\right)\left(r^{(r)}_{i j_2} - \bar{r}^{(r)}_{i}\right)}{\sqrt{\sum_{i=1}^{p}\left(r^{(r)}_{i j_1} - \bar{r}^{(r)}_{i}\right)^2\left(r^{(r)}_{i j_2} - \bar{r}^{(r)}_{i}\right)^2}}.$$

And the following estimated covariance matrices using Kendall's tau and Spearman's rho:

$$\hat{\mathbf{T}}_{j_1 j_2} = \begin{cases} \sin\left(\frac{\pi}{2}\hat{\tau}^{(r)}_{j_1 j_2}\right), & j_1 \neq j_2, \\ 1, & j_1 = j_2. \end{cases} \tag{4.11}$$

$$\hat{\mathbf{T}}_{j_1 j_2} = \begin{cases} 2\sin\left(\frac{\pi}{6}\hat{\rho}^{(r)}_{j_1 j_2}\right), & j_1 \neq j_2, \\ 1, & j_1 = j_2. \end{cases} \tag{4.12}$$

$$\hat{\mathbf{S}}_{i_1 i_2} = \begin{cases} \sin\left(\frac{\pi}{2}\hat{\tau}^{(c)}_{i_1 i_2}\right), & i_1 \neq i_2, \\ 1, & i_1 = i_2. \end{cases}$$

$$\hat{\mathbf{S}}_{i_1 i_2} = \begin{cases} 2\sin\left(\frac{\pi}{6}\hat{\rho}^{(c)}_{i_1 i_2}\right), & i_1 \neq i_2, \\ 1, & i_1 = i_2. \end{cases}$$

In Algorithm 6 we summarise the Nonparanormal Scalable Bigraphical Lasso approach for count data.

Algorithm 6 Nonparanormal scBiGLasso

Input: Maximum iteration number N, tolerance ε, M many observations of $p \times n$ count matrices $\mathbf{Y}_m$, $m = 1, \ldots, M$. β_1, β_2 and initial estimates of $\boldsymbol{\Psi}_{n\times n}$ and $\boldsymbol{\Theta}_{p\times p}$, $\boldsymbol{\Psi}^{(0)}_{n\times n}$ and $\boldsymbol{\Theta}^{(0)}_{p\times p}$.

For each $\mathbf{Y}_m$, calculate $\hat{\mathbf{T}}^{(m)}$ according to Equation (4.11) or (4.12).

$\mathbf{T} \leftarrow \frac{1}{M}\sum_{m=1}^{M} \hat{\mathbf{T}}^{(m)}$

repeat

 # Estimate $\boldsymbol{\Psi}_{n\times n}$:

for iteration $\tau = 1, \ldots, N$ **do**

Decompose $\boldsymbol{\Psi}^{(\tau-1)}_{n\times n} = \mathbf{U}^{(\tau-1)}\boldsymbol{\Lambda}_1^{(\tau-1)}\mathbf{U}^{(\tau-1)\top}$ and $\boldsymbol{\Theta}^{(\tau-1)}_{p\times p} = \mathbf{V}^{(\tau-1)}\boldsymbol{\Lambda}_2^{(\tau-1)}\mathbf{V}^{(\tau-1)\top}$.

for $i = 1, \ldots, n$ **do**

Partition $\boldsymbol{\Psi}^{(\tau-1)}_{n\times n}$ into $\psi^{(\tau-1)}_{ii}$, $\boldsymbol{\psi}^{(\tau-1)}_{i\setminus i}$, $\boldsymbol{\psi}^{(\tau-1)}_{\setminus ii}$ and $\boldsymbol{\Psi}^{(\tau-1)}_{\setminus i\setminus i}$.

Calculate $\mathbf{A}^{(\tau-1)}_{\setminus i\setminus i}$ as in Equation (4.10) and Proposition 4.2

with $\psi^{(\tau-1)}_{ii}$, $\mathbf{U}^{(\tau-1)}$, $\boldsymbol{\Lambda}_1^{(\tau-1)}$ and $\boldsymbol{\Lambda}_1^{(\tau-1)}$.

With *Lasso* regression, find a sparse solution, $\boldsymbol{\psi}^{*}_{i\setminus i}$,

for $p\mathbf{t}_{i\setminus i} + \mathbf{A}^{(\tau-1)}_{\setminus i\setminus i}\boldsymbol{\psi}^{(\tau)}_{i\setminus i} = \mathbf{0}_{n-1}$.

Calculate the direction vector from

$\boldsymbol{\psi}^{(\tau-1)}_{i\setminus i}$ to $\boldsymbol{\psi}^{*}_{i\setminus i}$: $\Delta\boldsymbol{\psi}^{(\tau)}_{i\setminus i} = \boldsymbol{\psi}^{*}_{i\setminus i} - \boldsymbol{\psi}^{(\tau-1)}_{i\setminus i}$.

Since the objective of solving $p\mathbf{t}_{i\setminus i} + \mathbf{A}^{(\tau-1)}_{\setminus i\setminus i}\boldsymbol{\psi}^{(\tau)}_{i\setminus i} = \mathbf{0}_{n-1}$

can be written as $f\left(\boldsymbol{\psi}_{i\setminus i}\right) = \left\|\mathbf{t}_{i\setminus i} + \frac{1}{p}\mathbf{A}^{(\tau-1)}_{\setminus i\setminus i}\boldsymbol{\psi}^{(\tau)}_{i\setminus i}\right\|_F^2$,

Calculate $\nabla f\left(\boldsymbol{\psi}_{i\setminus i}\right) = 2\frac{1}{p^2}\mathbf{A}^{(\tau-1)\top}_{\setminus i\setminus i}\mathbf{A}^{(\tau-1)}_{\setminus i\setminus i}\boldsymbol{\psi}^{(\tau)}_{i\setminus i} + 2\frac{1}{p}\mathbf{A}^{(\tau-1)\top}_{\setminus i\setminus i}\mathbf{t}_{i\setminus i}$.

Take $\zeta = \min\{\lambda_{11}, \ldots, \lambda_{1n}\}$.

 # Implement FISTA (Beck & Teboulle, 2009) with backtracking line search.

Calculate $Q = f\left(\boldsymbol{\psi}^{(\tau-1)}_{i\setminus i}\right) + \nabla f\left(\boldsymbol{\psi}^{(\tau-1)}_{i\setminus i}\right)^{\top}\Delta\boldsymbol{\psi}^{(\tau)}_{i\setminus i} + \frac{1}{2\zeta}\left\|\Delta\boldsymbol{\psi}^{(\tau)}_{i\setminus i}\right\|_F^2$ and $f(\boldsymbol{\psi}^{*}_{i\setminus i})$.

$t_0 = 1, a = 0$.

while $f\left(\boldsymbol{\psi}^{*}_{i\setminus i}\right) > Q$ **do**

$$a = a + 1, t_{a+1} = \frac{1 + \sqrt{1 + 4t_a^2}}{2}.$$

$$\boldsymbol{\psi}^{*}_{i\setminus i} = \boldsymbol{\psi}^{(\tau-1)}_{i\setminus i} + \frac{t_a - 1}{t_{a+1}}\Delta\boldsymbol{\psi}^{(\tau)}_{i\setminus i}.$$

$$\Delta\boldsymbol{\psi}^{(\tau)}_{i\setminus i} = \boldsymbol{\psi}^{*}_{i\setminus i} - \boldsymbol{\psi}^{(\tau-1)}_{i\setminus i}.$$

Calculate $Q = f\left(\boldsymbol{\psi}^{(\tau-1)}_{i\setminus i}\right) + \nabla f\left(\boldsymbol{\psi}^{(\tau-1)}_{i\setminus i}\right)^{\top}\Delta\boldsymbol{\psi}^{(\tau)}_{i\setminus i} + \frac{1}{2\zeta}\left\|\Delta\boldsymbol{\psi}^{(\tau)}_{i\setminus i}\right\|_F^2$ and $f(\boldsymbol{\psi}^{*}_{i\setminus i})$.

end while

Update the non-diagonal column $\boldsymbol{\psi}_{i\setminus i}^{(\tau)} = \boldsymbol{\psi}_{i\setminus i}^{*}$.

end for

 # Estimate $\boldsymbol{\Theta}_{p\times p}$:

Proceed as if estimating $\boldsymbol{\Psi}_{n\times n}$ with input $\mathbf{Y}^{\top}, \beta_1, \beta_2$.

Calculate the change in the estimated matrices from each iteration

$$\Delta\boldsymbol{\Psi}^{(\tau)} = \|\boldsymbol{\Psi}_{n\times n}^{(\tau)} - \boldsymbol{\Psi}_{n\times n}^{(\tau-1)}\|_F^2,$$

$$\Delta\boldsymbol{\Theta}^{(\tau)} = \|\boldsymbol{\Theta}_{p\times p}^{(\tau)} - \boldsymbol{\Theta}_{p\times p}^{(\tau-1)}\|_F^2,$$

end for

until Maximum iteration number reached, or $\max_{\tau^*=\tau-2,\tau-1,\tau} \left\{(\Delta\boldsymbol{\Psi}^{(\tau^*)} + \Delta\boldsymbol{\Theta}^{(\tau^*)})\right\} < \varepsilon$, for $\tau \geq 3$.

4.5 Numerical Results

In this section, we implement our Scalable Bigraphical Lasso algorithm in `MATLAB` (MATLAB, 2020). After precision matrices $\boldsymbol{\Psi}_{n\times n}$ and $\boldsymbol{\Theta}_{p\times p}$ are inferred, they are transformed into binary matrices to reveal the network structures, where any negative value in the precision matrices become 1 and any non-negative value become 0. We illustrate applications of our overall approach on both synthetic and real datasets as described in the following subsections. The code to reproduce our results is available on GitHub.

4.5.1 Synthetic Gaussian Data

To demonstrate the efficiency of our Scalable Bigraphical Lasso algorithm (Algorithm 5), we generate sparse positive definite matrices $\boldsymbol{\Psi}_{n\times n}$ and $\boldsymbol{\Theta}_{p\times p}$, then simulate M many $p \times n$ Gaussian data $Y_G^{(m)}$, $m = 1,\ldots,M$ from $\mathbf{mN}\left(\mathbf{0}, \left(\boldsymbol{\Psi}_{n\times n} \oplus \boldsymbol{\Theta}_{p\times p}\right)^{-1}\right)$. We plug $Y_G^{(m)}$, $m = 1,\ldots,M$ into our implemented Algorithm 5, Bigraphical Lasso from Kalaitzis *et al.* (2013) and TeraLasso from Greenewald *et al.* (2019). Figure 4.1 shows a comparison between the convergence times of Algorithm 1 and Bigraphical Lasso for increasing problem dimensions $n = p$. We can observe that, as expected, Algorithm 5 converges in significantly faster times, allowing one to tackle higher dimensional

problems in practice. Table 4.1 shows the network recovery when $n = p = 100$. We can see that our method provides high Accuracy while improving greatly on speed; see Section 4.5.2 for the definition of Accuracy.

Table 4.1: Comparison between computational convergence times, Accuracy of $\boldsymbol{\Psi}$ and of $\boldsymbol{\Theta}$ for Bigraphical Lasso (Kalaitzis *et al.*, 2013), TeraLasso (Greenewald *et al.*, 2019) and Algorithm 5, on a synthetic Gaussian dataset with dimensions $n = p = 100$.

Method	**Accuracy**$_\Psi$	**Accuracy**$_\Theta$	**Time(s)**
BigLasso	0.9032	0.9028	852.56
ScBigLasso	0.9032	0.9028	2.88
TeraLasso	0.7948	0.8460	0.42

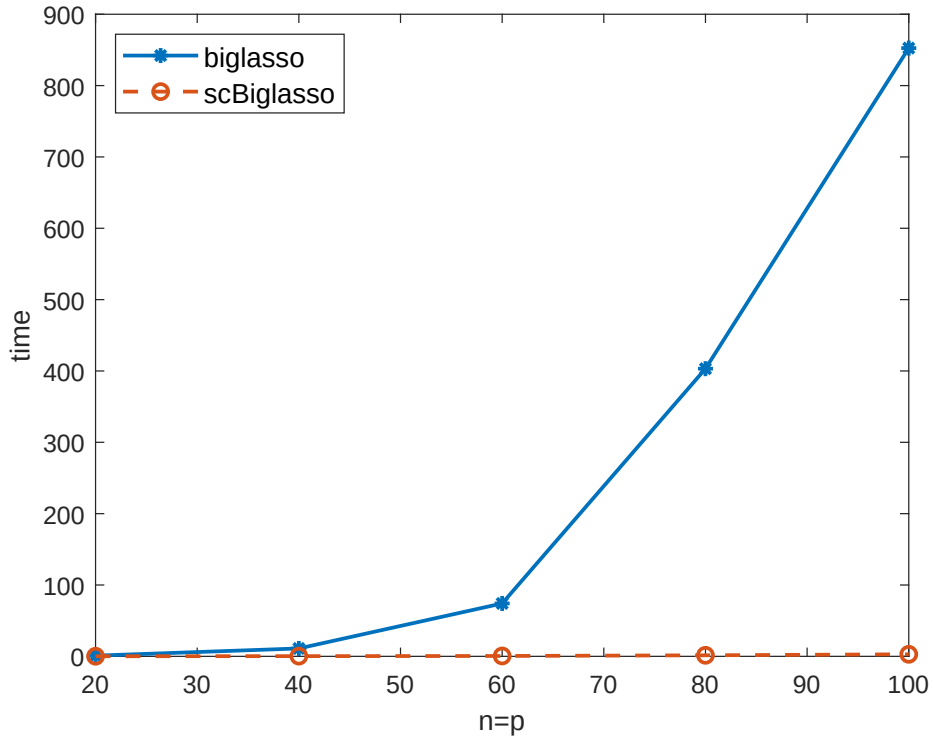

Figure 4.1: Computational convergence time (seconds) comparison between Bigraphical Lasso (Kalaitzis *et al.*, 2013) and Algorithm 5, for increasing values of the dataset dimensions $n = p$.

4.5.2 Synthetic count data

We generate and process Gaussian Copula-based count data through the following steps:

1. Generate sparse positive definite matrix $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$. Calculate the Kronecker sum of $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$.Generate M multivariate-normal vectors of length $p\times n$ from $\mathbf{mN}\left(\mathbf{0},\mathbf{\Omega}^{-1}\right)$, where $\mathbf{\Omega}=\mathbf{\Psi}_{n\times n}\otimes\mathbf{I}_p+\mathbf{I}_n\otimes\mathbf{\Theta}_{p\times p}$.

2. Centre each of the M multivariate-normal vectors around their mean, and reshape the vectors into $p\times n$ matrices $\mathbf{X}^{(1)},\ldots,\mathbf{X}^{(M)}$.

3. For each $\mathbf{X}^{(m)}$, $m=1,\ldots,M$, calculate the matrix $P^{(m)}$ such that $P_{ij}^{(m)}=\Phi\left(X_{ij}^{(m)}\right)$, where $\Phi(\cdot)$ is the cumulative density function of the standard normal distribution.

4. For each $m=1,\ldots,M$, produce the negative binomial variable

$$Y_{ij}^{(m)}=QNB\left(P_{ij}^{(m)},r,p\right),$$

where $QNB\left(\cdot,r,p\right)$ is the quantile function of Negative-Binomial$\left(r,p\right)$, with r the number of success to be observed and p the success rate, resulting in M matrices of count data $\mathbf{Y}_m$.

Below we describe some of the criteria we use to assess the recovery of the synthetic network. Denote TP as the number of *True Positives* in the network recovery, TN as the number of *True Negatives* in the network recovery, FP as the number of *False Positives* in the network recovery, and FN the number of *False Negatives* in the network recovery, then we can define

$$\begin{aligned}
&Precision=\frac{TP}{TP+FP},\quad Recall=\frac{TP}{TP+FN},\\
&Accuracy=\frac{TP+TN}{TP+TN+FP+FN},\\
&TPR=\frac{TP}{TP+FN},\quad FPR=\frac{FP}{TN+FP}.
\end{aligned}$$

Figure 4.2 shows some results from synthetic data. Figure 4.2 (a) is the Precision-Recall of the recovery of $\mathbf{\Psi}_{n\times n}$ with changing β_1 (different points on the graph) and

β_2 (different colours on the graph). Two arbitrary values of β_2 have been chosen to illustrate how the results do not depend on β_2. This is expected as β_1 is the regularization parameter for $\mathbf{\Psi}_{n\times n}$, while β_2 corresponds to $\mathbf{\Theta}_{p\times p}$. A similar result is shown in Figure 4.2 (b), where the Precision-Recall of the recovery of $\mathbf{\Theta}_{p\times p}$ heavily depends on the choice of β_2, regardless of the β_1 value. Figure 4.2 (c)(d) show that high values of TPR and Accuracy, with low values of FPR, can be achieved for appropriate choices of β_1 and β_2 in the range $[0.005, 0.016]$.

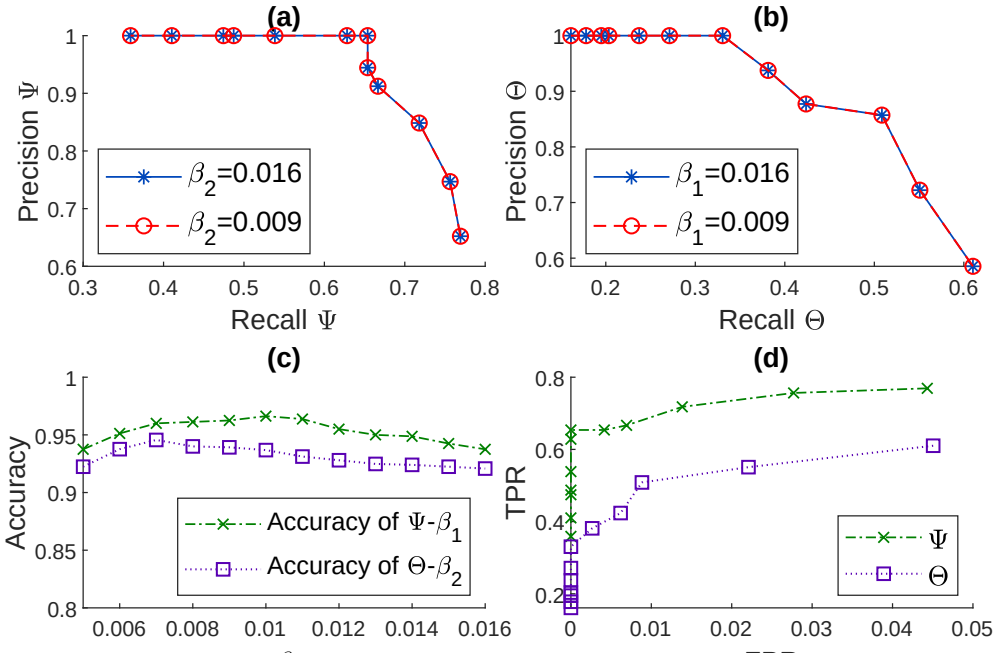

Figure 4.2: Synthetic network recovery results. **(a)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Psi}_{n\times n}$; **(b)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Theta}_{p\times p}$; **(c)** Accuracy vs corresponding regularization parameter β_1 (β_2) of the network recovery relating to the support of $\mathbf{\Psi}_{n\times n}$ ($\mathbf{\Theta}_{p\times p}$); **(d)** TPR-FPR of the network recovery relating to the support of $\mathbf{\Psi}_{n\times n}$ ($\mathbf{\Theta}_{p\times p}$), where the corresponding regularization parameter β_1 (β_2) $\in \{0.005 : 0.001 : 0.0016\}$.

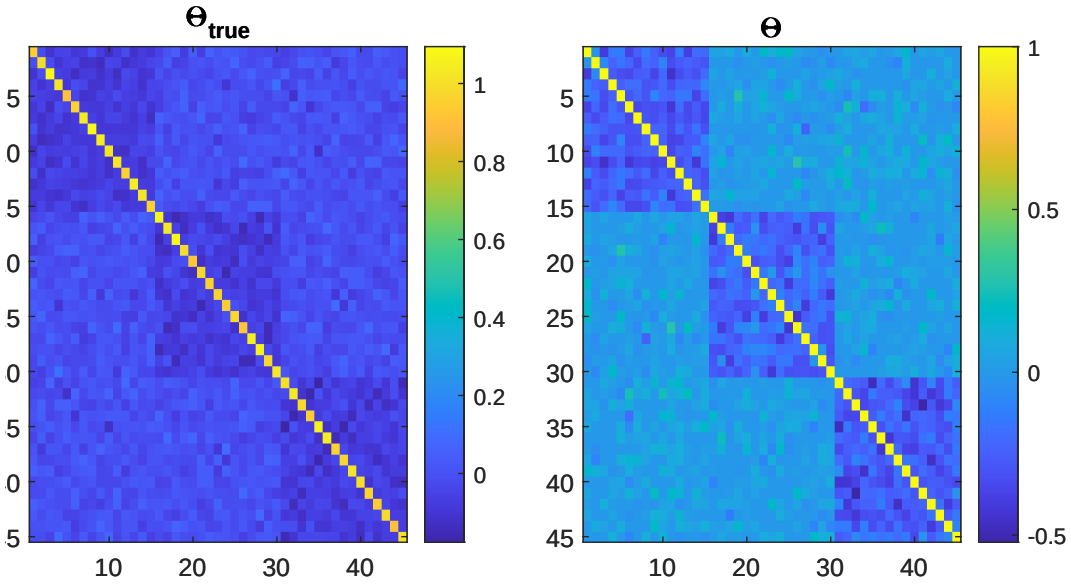

Figure 4.3: Synthetic network recovery. We generated synthetic data as described in Section 5.2 using a block-diagonal precision matrix for $\mathbf{\Theta}_0$ plus Gaussian noise (Left plot). On the right we plot the estimated $\mathbf{\Theta}$ via our method. In this example, we used $\beta_2 = 0.002$.

Figure 4.3 shows network recovery for another synthetic count dataset, where the original precision matrix $\mathbf{\Theta}_0$ was generated with block diagonals and Gaussian noise throughout the matrix. We observe that our method leads to good recovery of the corresponding blocks. Further discussion on the choice of optimal regularization parameters $\beta = (\beta_1, \beta_2)$ is in Subsection 4.5.5.

4.5.3 An example from the COIL-20 Dataset

We use frames of several rotating objects from the COIL-20[1] dataset for data analysis. Each frame is a grey-scaled picture, as shown in Figure 4.4.

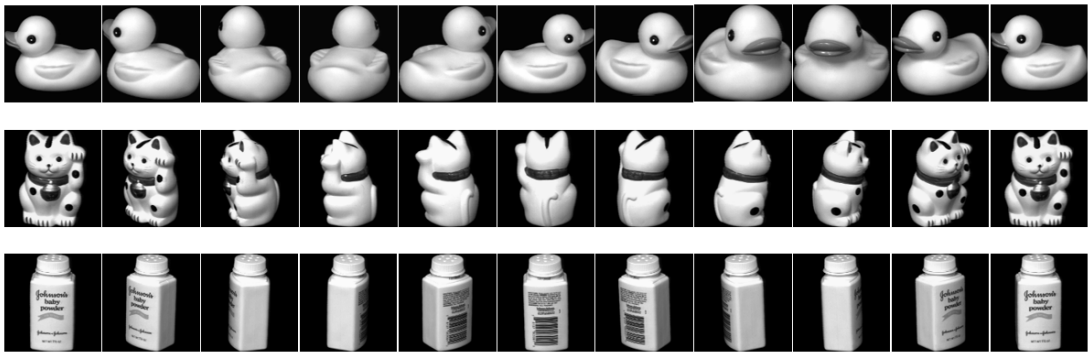

Figure 4.4: First line: frames of a rotating rubber duck from COIL-20 dataset.
Second line: frames of a rotating toy cat from COIL-20 dataset.
Third line: frames of a ratating baby powder bottle from COIL-20 dataset.
Each original frame contained 128 pixels.

We reduced the resolution of each frame from 128×128 to 16×16, and subsampled 11 frames evenly from total 72 frames for each object, respectively. After vectorising each frames (stacking their pixels into 256×1 vectors), we arrange them into a design matrix $\mathbf{Y}$. Here we wish to test if our model is able to cluster frames of the same object together, so arranged the vectorised frames of all three objects into the same matrix. This leads to a data matrix 256×33, where each column vector of length 256 is from a vectorised frame. The data were plugged into our Algorithm 6 of Scalable Bigraphical Lasso. After inferring the matrix $\boldsymbol{\Psi}$ (33×33) and Θ (256×256), we use a binary transformation where only the negative values are considered as an edge in the network.

[1] https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

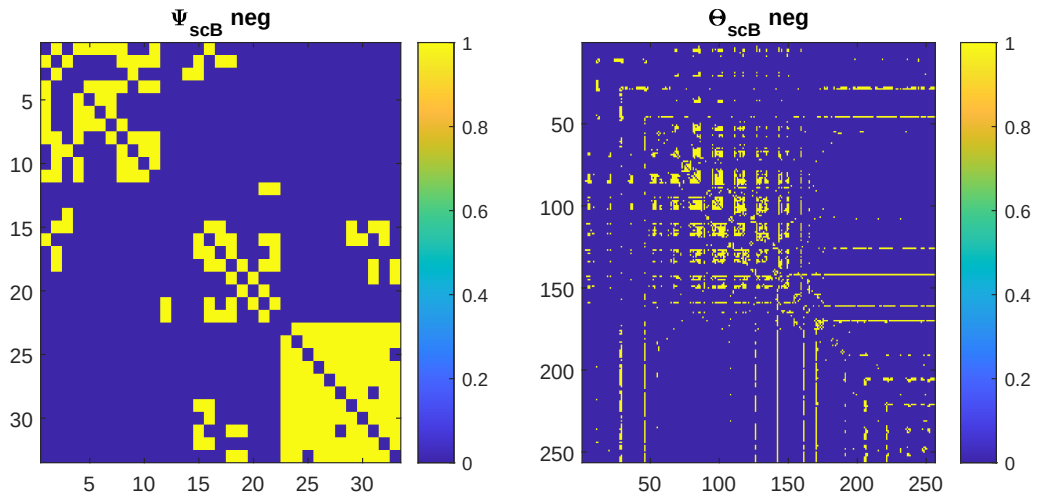

Figure 4.5: Recovered networks of both relationship between frames and between pixels in frames. $\mathbf{\Psi}$ represents the relationship between frames (33 frames, 11 frames for each of the three objects, respectively), while Θ represents the structure in pixels. In this example, we used $(\beta_1, \beta_2) = (0.008, 0.007)$.

Figure 4.5 shows the results inferred from the COIL-20 data. The network of frames ($\mathbf{\Psi}$) shows roughly three distinct clusters of frames every 11 frames, indeed, when we arrange $\mathbf{Y}$, the 1st-11th columns are the vectorised matrices from the frames of rotating rubber ducks, the 12th-22nd columns are the vectorised matrices from the frames of rotating toy cats, and the 23rd-33rd columns are the vectorised from the frames of rotating baby powder bottles. The network of pixels ($\mathbf{\Theta}$) showed strong dependencies between the 49th-160th pixels in intervals of roughly 16, where 16 is the number of pixels we considered in each column of a frame, and the 49th-160th pixels in the subsample of a frame roughly corresponds to where most of the white pixels, i.e. the object itself are. This result shows that our method, the scalable Bigraphical Lasso, is able to cluster image frames conditional on the dependencies between pixels.

We acknowledge that the image clustering result presented still has space for improvement. We shall improve this method for image clustering and discuss it further in Chapter 5.

4.5.4 mESC scRNA-seq data

We use a single cell gene expression dataset from mouse embryonic stem cells (mESC) available in Buettner *et al.* (2015). The data consist of measurements of gene counts in 182 single cells at different stages of the cell cycle. We will refer to the three phases as G1, S and G2M, as shown in Figure 4.6. According to the label provided by Buettner *et al.* (2015), in our dataset there are 65 cells in the G2M phase.

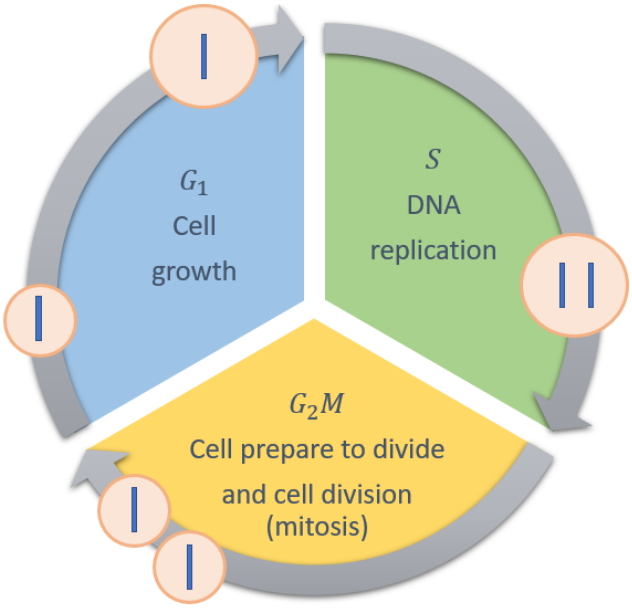

Figure 4.6: Cell cycle stages drawn according to the description in Humphrey & Brooks (2008).

After analysing the list of gene names in dataset through `DAVID` tool (Dennis *et al.*, 2003), 700 genes are annotated as cell cycle related. Of these, we considered 167 genes more active during mitosis as labelled by `DAVID`, the cell division phase and last part of the cell cycle (G2M).

In Figures 4.7 and 4.8, we show how our model allows the identification of the sub-population of cells that correspond to the G2M stage. In Figure 4.7 we show the estimated precision matrices for the cells (left) and the genes (right). We use a binary transformation where only the negative values are considered an edge in the network. In Figure 4.8 we plot the corresponding networks, over imposing the clusters found with the label propagation approach developed by Raghavan *et al.*

(2007) (implemented in R package `igraph` (Csardi *et al.*, 2006)). We note that ~ 92% of the G2M cells are clustered in a densely connected module ($\boldsymbol{\Psi}$ network plot in Figure 4.8), while no connection is measured between cells in different phases of the cell cycle. As expected, on the other hand, the mitosis genes are all densely connected in a single cluster ($\boldsymbol{\Theta}$ network plot in Figure 4.8).

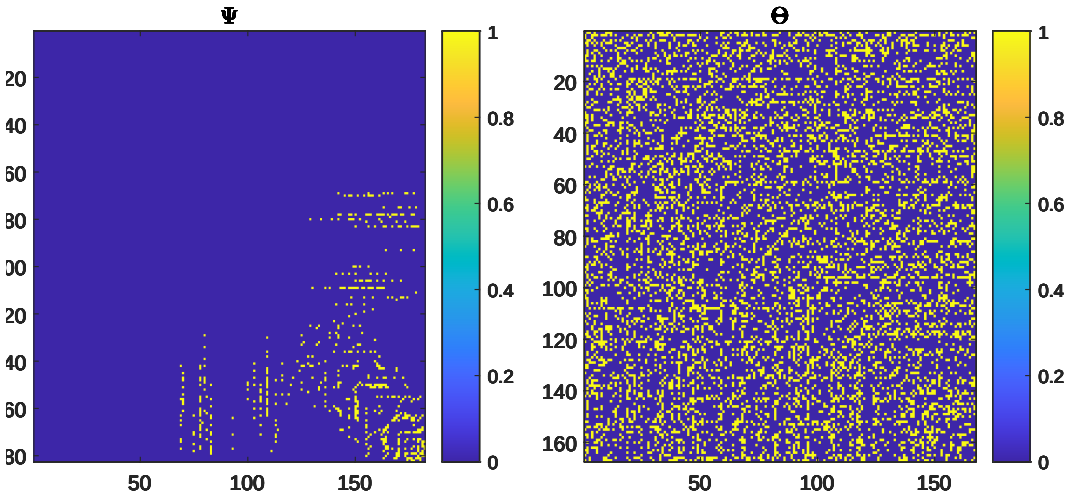

Figure 4.7: Networks recovered by our proposed Scalable Bigraphical Lasso algorithm combined with the nonparanormal transformation as described in Section 4.2, $(\beta_1, \beta_2) = (0.014, 0.001)$.

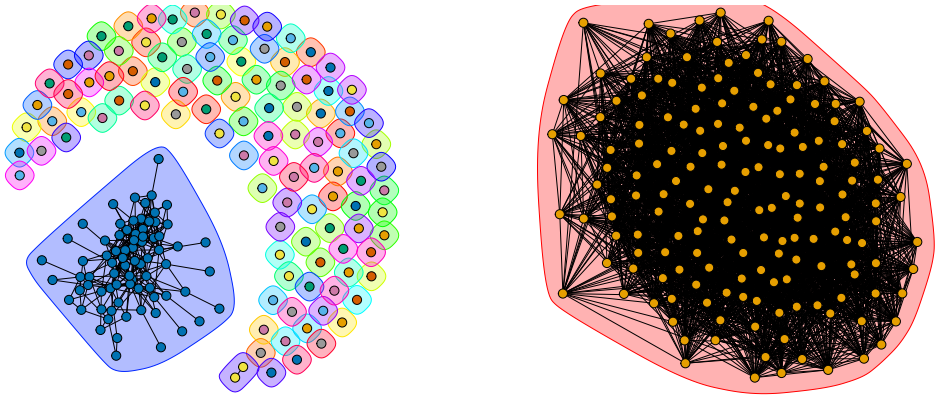

Figure 4.8: $\boldsymbol{\Psi}$ (left) and $\boldsymbol{\Theta}$ (right) induced networks and communities. Each coloured outer circles corresponds to a cluster. Different outer circles with similar colours corresponds to different clusters. On the right, all the genes are identified to be in the same cluster.

4.5.5 The effect of regularization parameters

Our algorithms depend on the regularization parameters β_1 and β_2. Figure 4.10 below illustrates the effect of these parameters on the performance of our algorithms. We generated two random sparse positive-definite matrices with a sparsity of 0.1 and non-zero entries normally distributed with mean 1 and variance 2. These were used as precision matrices $\mathbf{\Psi}_0$ and $\mathbf{\Theta}_0$ to create the Kronecker product matrix Ω_0 as plotted in Figure 4.9. This synthetic dataset corresponds to the experiment plotted in Figure 4.2.

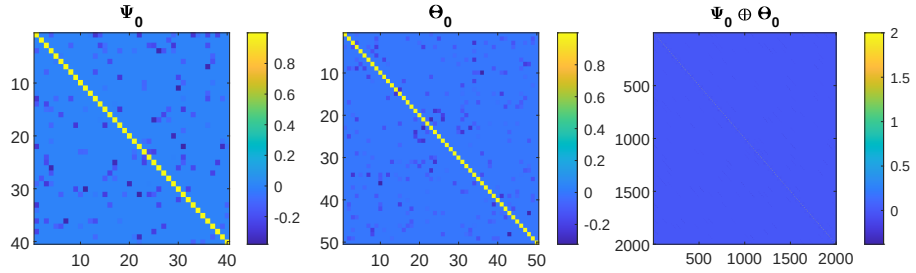

Figure 4.9: Precision matrix $\mathbf{\Psi}_0$ (left), $\mathbf{\Theta}_0$ (centre) and corresponding Kronecker product matrix Ω_0 (right) for our exemplar synthetic dataset.

In Figure 4.10(a)-(b) we show the Precision-Recall for the estimated precision matrices. In particular, subfigure (a) refers to the estimate of $\mathbf{\Psi}_{n\times n}$ when varying β_1, while subfigure (b) refers to the estimate of $\mathbf{\Theta}_{p\times p}$ when varying β_2. These curves suggest that optimal choices of β_1 lie within the interval $[0.007, 0.01]$ and similarly β_2 should lie within the interval $[0.006, 0.008]$. When choosing values within these intervals, one tries to strike a balance between Precision and Recall. In order to explore further the impact of the regularization parameters, we also computed the Akaike Information Criteria (AIC) (Akaike, 1998).

$$AIC_\Psi = -2\ln\pi^*(\mathbf{Y}|\mathbf{\Psi}) + 2w,$$

$$AIC_\Theta = -2\ln\pi^*(\mathbf{Y}|\mathbf{\Theta}) + 2w,$$

where w is the number of edges in the estimated network, and

$$\ln\pi^*(\mathbf{Y}|\mathbf{\Psi}) = \ln\pi(\mathbf{Y}|\mathbf{\Psi},\mathbf{\Theta}) - \text{terms not concerning } \mathbf{\Psi},$$

$$\ln\pi^*(\mathbf{Y}|\mathbf{\Theta}) = \ln\pi(\mathbf{Y}|\mathbf{\Psi},\mathbf{\Theta}) - \text{terms not concerning } \mathbf{\Theta}.$$

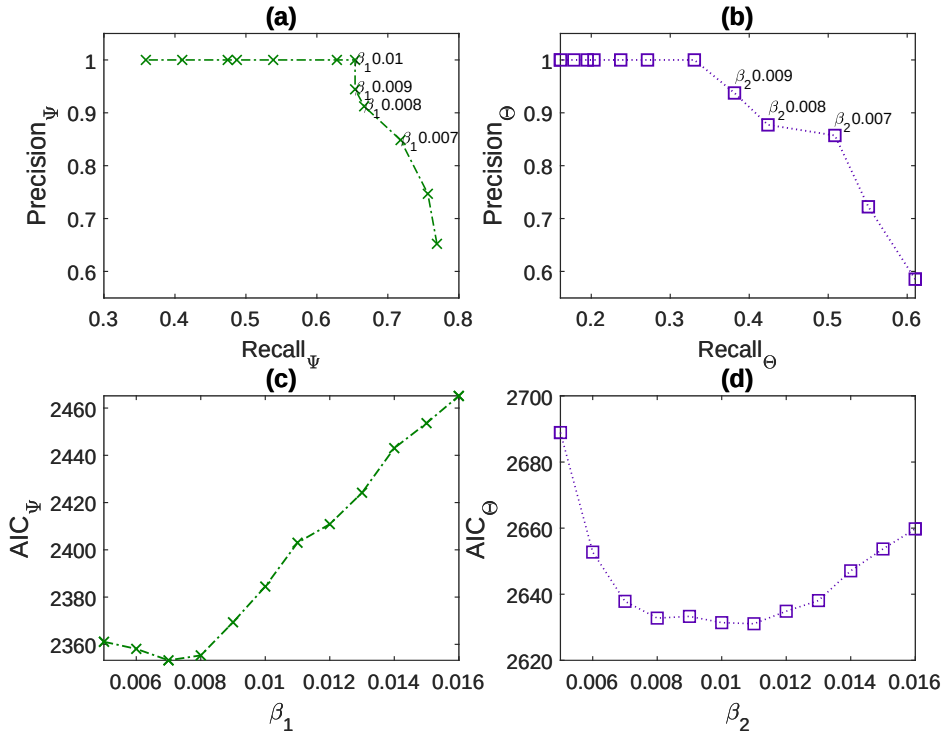

Figure 4.10: Synthetic network recovery results. Information Criterion and regularization parameters.**(a)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Psi}_{n\times n}$; **(b)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Theta}_{p\times p}$; Akaike Information Criterion and regularization parameters. **(c)** β_1-AIC_Ψ; **(d)** β_2-AIC_Θ;

In subfigures (c) and (d) we plot the AIC curves corresponding to the estimated precision matrices when varying β_1 and β_2 respectively. AIC is a heuristic criterion that helps selecting from several models. Ones with lower AIC values are generally preferred, however, a lower AIC does not necessarily indicate one model is better than another and further investigation is usually needed. The AIC curve depicted in subfigure (c) confirms the suggestion on the optimal choices for the regularization parameters obtained with the Precision-Recall plot, but the AIC curve in subfigure (d) suggest a vaguer range for optimal regularization parameter in $[0.006, 0.012]$. Therefore, when dealing with problems without known truth, although AIC can be used to help identify the interval of potential optimal regularization parameters, it is

not necessarily accurate and should be used with caution. Alternative methods to find the optimal regularization parameter should be explored in the future.

4.6 Conclusions

In this work, we present a Scalable Bigraphical Lasso algorithm. In particular, we exploit eigenvalue decomposition of the Cartesian product graph to present a more efficient version of the algorithm presented in Kalaitzis *et al.* (2013). Our approach reduces memory requirements from $O(n^2p^2)$ to $O(n^2+p^2)$, and reduces the computational time by up to a factor of 200 in our experiments (case $p = n = 100$ in Figure 4.1 and Table 4.1). Note that comparisons for $n = p > 100$ were restricted because of the memory limitation in Kalaitzis *et al.* (2013). Additionally, we propose a Gaussian-copula based model and a semiparametric approach that enables the application of the proposed Bigraphical model to non-Gaussian data. This is particularly relevant for count data applications, such as single cell data. Future work will include optimisation of the choice of the regularization parameters, and potential extension to K-way network inference for non-Gaussian data, with $K > 2$.

Data availability

The code and data is available at `https://github.com/luisacutillo78/Scalable_Bigraphical_Lasso.git`.

Chapter 5

Scalable K-graphical Lasso

5.1 Introduction

In this chapter, we extend the two-way Scalable Bigraphical Lasso for application on multi-way data. The main motivation of this extension is that tensor-valued data are more and more common in the real world. For example, coloured images encodes information in three colour channels: red, green and blue (RGB). Another example is that functional magnetic resonance image (fRMI) data are naturally represented in three-way tensors (Xu *et al.*, 2017). Xiong *et al.* (2010) also considered the analysis of tensor-valued data for recommendation systems.

Recently, Gaussian graphical models have been extended for application on tensor-valued data. TeraLasso (Greenewald *et al.*, 2019) considers a Kronecker sum structured model for tensor-valued data. SyGlasso (Wang *et al.*, 2020) considers a Kronecker product structured model for tensor-valued data. In Chapter 4, we have shown that in two-way cases, our Scalable Bigraphical Lasso algorithm obtains better accuracy compared to TeraLasso, while we prefer a KS structured model for its sparsity and interpretability. Moreover, methods such as TeraLasso and SyGlasso have not considered the application on non-Gaussian data. Following the previous arguments in Section 2.2 and Chapter 4, we consider the Gaussian Copula associating with Cartesian products of Gaussian Markov random field graphs, where each individual Gaussian Markov random field graph encodes the dependency relationship along each fibre of the tensor-valued data.

This chapter is structured as follows: In Section 5.2 we present our Scalable K-graphical Lasso algorithm for Gaussian tensor-valued data; In Section 5.3 we present a semiparametric extension to the Scalable K-graphical Lasso method for non-Gaussian data; In Section 5.4 we showcase the performance of our method on both synthetic and real datasets.

5.2 Scalable K-graphical Lasso Algorithm

Similar to Greenewald *et al.* (2019), we assume that the vectorised tensor $\text{vec}(\mathcal{Y})$ follows a multivariate Normal distribution, in particular,

$$\text{vec}(\mathcal{Y}) \sim \mathbf{mN}(\mathbf{0}, \mathbf{\Omega}),$$

where

$$\mathbf{\Omega} = \mathbf{\Psi}^{(1)} \oplus \mathbf{\Psi}^{(2)} \oplus \cdots \oplus \mathbf{\Psi}^{(K)} = \sum_{i=1}^{K} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \mathbf{\Psi}^{(i)} \otimes \mathbf{I}_{[d_{(i+1):K}]}, \tag{5.1}$$

with

$$\mathbf{\Psi}^{(k)} \in \mathbb{R}^{d_k \times d_k},$$

$$\mathbf{I}_{[d_{k:l}]} = \mathbf{I}_{d_k} \otimes \cdots \otimes \mathbf{I}_{d_l},\ k, l \leq K, k \leq l.$$

For M independent identically distributed data samples $\{\mathcal{Y}_1, \ldots, \mathcal{Y}_M\}$ in the form of K-way tensors, we estimate the sparse KS-structured inverse covariance matrices by minimising the ℓ_1-penalized negative log-likelihood:

$$\min_{\mathbf{\Psi}^{(1)},\ldots,\mathbf{\Psi}^{(K)}} \left\{ -\ln |\, \mathbf{\Omega} \,| + \sum_k p_k \text{tr}\left(\mathbf{\Psi}^{(k)} \mathbf{S}^{(k)}\right) + \sum_k \beta_k \|\mathbf{\Psi}^{(k)}\|_1 \right\}, \tag{5.2}$$

where $p_k = \frac{\prod(d_i)_{i=1}^{K}}{d_k}$, $\mathbf{S}^{(k)} = \frac{1}{M p_k} \sum_{n=1}^{N} \mathbf{Y}_{(k)}^{(n)} \mathbf{Y}_{(k)}^{(n)\top} \in \mathbb{R}^{d_k \times d_k}$ with $\mathbf{Y}_{(k)}^{(n)}$ being the matricization of $\mathcal{Y}$ along mode k. Following the idea of flip-flop approach in Chapter 4, we propose to focus on updating one $\mathbf{\Psi}^{(k)}$ at a time while fixing all the other $\mathbf{\Psi}^{(k')}$, $k' \neq k$. In case of no regularization, the k-th step of the optimization problem is reduced to

$$\min_{\mathbf{\Psi}^{(k)}} \left\{ -\ln |\, \mathbf{\Psi}^{(1)} \oplus \cdots \oplus \mathbf{\Psi}^{(k)} \oplus \cdots \oplus \mathbf{\Psi}^{(K)} \,| + p_k \text{tr}\left(\mathbf{\Psi}^{(k)} \mathbf{S}^{(k)}\right) \right\}.$$

Obtaining the stationary point:

$$\mathbf{S}^{(k)} - \frac{1}{2p_k} \mathbf{S}^{(k)} \circ \mathbf{I}_{d_k} = \frac{1}{p_k} \text{tr}_{p_k}(\mathbf{W}) - \frac{1}{2p_k} \text{tr}(\mathbf{W}) \circ \mathbf{I}_{d_k}, \tag{5.3}$$

where $\circ$ is the Hadamard product and we define $\mathbf{W} = \left(\mathbf{\Psi}^{(1)} \oplus \cdots \oplus \mathbf{\Psi}^{(K)}\right)^{-1}$. The block-wise trace $\mathrm{tr}_p(\cdot)$ is an operator as defined in Definition 4.1.

Proof of equation (5.3). Focus on $\mathbf{\Psi}^{(k)}$ and ignore the penalty term, denote the objective function relevant to $\mathbf{\Psi}^{(k)}$, $L_{\mathbf{\Psi}^{(k)}}$, as

$$L_{\mathbf{\Psi}^{(k)}} = -\ln|\,\mathbf{\Psi}^{(1)} \oplus \cdots \oplus \mathbf{\Psi}^{(k)} \oplus \cdots \oplus \mathbf{\Psi}^{(K)}\,| + p_k \mathrm{tr}\left(\mathbf{\Psi}^{(k)}\mathbf{S}^{(k)}\right).$$

Calculate the element-wise first-order derivative of L_k with respect to $\mathbf{\Psi}^{(k)}$:

$$\begin{aligned}
\frac{\partial \ln|\,\mathbf{\Psi}^{(1)} \oplus \cdots \oplus \mathbf{\Psi}^{(k)} \oplus \ldots \mathbf{\Psi}^{(K)}\,|}{\partial \psi_{ij}^{(k)}} &= \mathrm{tr}\left\{\mathbf{\Omega}^{-1}\frac{\partial\left(\mathbf{\Psi}^{(1)} \oplus \cdots \oplus \mathbf{\Psi}^{(k)} \oplus \cdots \oplus \mathbf{\Psi}^{(K)}\right)}{\partial \psi_{ij}^{(k)}}\right\} \\
&= \mathrm{tr}\left\{\mathbf{\Omega}^{-1}\left(\frac{\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{\Psi}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}}{\partial \psi_{ij}^{(k)}}\right)\right\} \\
&= \mathrm{tr}\left\{\mathbf{\Omega}^{-1}\left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \left(\mathbf{J}^{ij} + \mathbf{J}^{ji} - \mathbf{J}^{ij}\mathbf{J}_{ji}\right) \otimes \mathbf{I}_{[d_{(k+1):K}]}\right)\right\} \\
&= \mathrm{tr}\left\{\mathbf{W}\begin{bmatrix} 0 & \ldots & 0 \\ \vdots & \mathbf{I}_{p_k}^{(i,j)} & \vdots \\ 0 & \ldots & 0 \end{bmatrix}\right\} + \mathrm{tr}\left\{\mathbf{W}\left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{J}^{ji} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right)\right\} \\
&\quad - \mathrm{tr}\left\{\mathbf{W}\left(I_{[d_{1:(k-1)}]} \otimes \mathbf{J}^{ij}\mathbf{J}^{ji} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right)\right\} \\
&= 2\mathrm{tr}\left\{\mathbf{W}_{(i,j)}\right\} - \mathbb{1}_{\{i=j\}} \cdot \mathrm{tr}\left\{\mathbf{W}_{ij}\right\},
\end{aligned}$$

where $\mathbf{J}^{ij}$ is the single entry matrix with $J_{ij} = 1$ and zeros elsewhere, $\mathbf{I}^{(i,j)}$ is at the (i,j)-th block of size $p_k \times p_k$, and

$$\mathbb{1}_{\{i=j\}} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Taking into account the whole matrix $\mathbf{\Psi}^{(k)}$, we have

$$\frac{\partial \ln|\,\mathbf{\Psi}^{(1)} \oplus \cdots \oplus \mathbf{\Psi}^{(k)} \oplus \cdots \oplus \mathbf{\Psi}^{(K)}\,|}{\partial \mathbf{\Psi}^{(k)}} = 2\mathrm{tr}_{p_k}(\mathbf{W}) - \mathrm{tr}_{p_k}(\mathbf{W}) \circ \mathbf{I}_{d_k}.$$

We also have

$$\frac{\partial p_k \mathrm{tr}\left(\mathbf{\Psi}^{(k)}\right)}{\partial \mathbf{\Psi}^{(k)}} = 2p_k\mathbf{S}^{(k)} - \mathbf{S}^{(k)} \circ \mathbf{I}_{d_k}.$$

To sum up, the first-order derivative of the objective function (ignoring the penalties) with respect to $\mathbf{\Psi}^{(k)}$ is

$$\frac{\partial L_k}{\partial \mathbf{\Psi}^{(k)}} = -2\mathrm{tr}_{p_k}(\mathbf{W}) + \mathrm{tr}_{p_k}(\mathbf{W}) \circ \mathbf{I}_{d_k} + 2p_k \mathbf{S}^{(k)} - \mathbf{S}^{(k)} \circ \mathbf{I}_{d_k}.$$

Let $\frac{\partial L_k}{\partial \mathbf{\Psi}^{(k)}} = 0$, we have Equation (5.3). □

To solve (5.3), consider the eigen-decomposition $\mathbf{\Psi}^{(k)} = \mathbf{U}^{(k)} \mathbf{\Lambda}_k \mathbf{U}^{(k)\top}$ for all $k = 1, \ldots, K$, where $\mathbf{\Lambda}_k \in \mathbb{R}^{d_k \times d_k}$ are eigenvalue diagonal matrices, and $\mathbf{U}^{(k)} = \left(u_{ij}^{(k)}\right) \in \mathbb{R}^{d_k \times d_k}$ are orthogonal eigenvector matrices. It follows that Equation (5.1) can be rewritten as

$$\mathbf{\Omega} = \left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \left(\sum_{i=1}^{K} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right) \left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right). \tag{5.4}$$

Proof of Equation (5.4).

$$\begin{aligned}
\mathbf{\Omega} &= \sum_{i=1}^{K} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \mathbf{\Psi}^{(i)} \otimes \mathbf{I}_{[d_{(i+1):K}]} \\
&= \sum_{i=1}^{K} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \mathbf{U}^{(i)} \mathbf{\Lambda}_i \mathbf{U}^{(i)\top} \otimes \mathbf{I}_{[d_{(i+1):K}]} \\
&= \sum_{i=1}^{K} \left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(i-1)} \mathbf{I}_{[d_{1:(i-1)}]} \mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(i-1)\top}\right) \otimes \mathbf{U}^{(i)} \mathbf{\Lambda}_i \mathbf{U}^{(i)\top} \\
&\qquad \otimes \left(\mathbf{U}^{(i+1)} \otimes \cdots \otimes \mathbf{U}^{(K)} \mathbf{I}_{[d_{(i+1):K}]} \mathbf{U}^{(i+1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \\
&= \sum_{i=1}^{K} \left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(i-1)} \otimes \mathbf{U}^{(i)} \otimes \mathbf{U}^{(i+1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \left(\mathbf{I}_{[d_{1:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right) \\
&\qquad \cdot \left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(i-1)\top} \otimes \mathbf{U}^{(i)\top} \otimes \mathbf{U}^{(i+1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \\
&= \left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(i-1)} \otimes \mathbf{U}^{(i)} \otimes \mathbf{U}^{(i+1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \left(\sum_{i=1}^{K} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right) \\
&\quad \cdot \left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right).
\end{aligned}$$

□

We note that the inverse of a symmetric matrix for which an eigenvalue decomposition is provided is obtained by inverting the eigenvalues,

$$\mathbf{W} = \mathbf{\Omega}^{-1} = \left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \left(\sum_{i=1}^{K} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right)^{-1} \left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right).$$

Taking

$$\left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right)\left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{I}_{d_k} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) = \mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]},$$

then $\mathbf{W\Omega} = \mathbf{I}$ can be premultiplied by $\left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right)$ to provide

$$\begin{aligned}&\left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(k-1)\top} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \cdot \mathbf{W\Omega} \\ &= \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \mathbf{D}\left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \mathbf{\Omega},\end{aligned} \tag{5.5}$$

where $\mathbf{D} = \left(\sum_{i=1}^{K} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right)^{-1}$ is a diagonal matrix.

Proof of Equation (5.5).

$$\begin{aligned}&\left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(k-1)\top} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \cdot \mathbf{W\Omega} \\ &= \left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(k-1)\top} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right)\left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \\ &\quad \cdot \mathbf{D}\left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \mathbf{\Omega} \\ &= \left(\mathbf{U}^{(1)\top}\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(k-1)\top}\mathbf{U}^{(k-1)} \otimes \mathbf{I}_{d_k}\mathbf{U}^{k} \otimes \mathbf{U}^{(k+1)\top}\mathbf{U}^{(k+1)} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\mathbf{U}^{(K)}\right) \\ &\quad \cdot \mathbf{D}\left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \mathbf{\Omega} \\ &= \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \mathbf{D}\left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \mathbf{\Omega}.\end{aligned}$$

□

If we multiply both sides of Equation (5.5) by $\left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(k-1)} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right)$, we have

$$\begin{aligned}\mathbf{W\Omega} = &\left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \mathbf{D}\left(\mathbf{I}_{[d_{1:(k_1)}]} \otimes \mathbf{U}^{(k)\top} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \\ &\cdot \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{\Psi}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]} + \sum_{i \neq k} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right).\end{aligned} \tag{5.6}$$

Proof of Equation (5.6). If we multiply the left side of Equation (5.5) by

$\left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(k-1)} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right)$, we have

$$
\begin{aligned}
&\left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(k-1)\top} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \cdot \mathbf{W}\boldsymbol{\Omega} \\
&\quad \cdot \left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(k-1)} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \\
&= \left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(k-1)\top} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \mathbf{I} \\
&\quad \cdot \left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(k-1)} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \\
&= \left(\mathbf{U}^{(1)\top}\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(k-1)\top}\mathbf{U}^{(k-1)} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)\top}\mathbf{U}^{(k+1)} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\mathbf{U}^{(K)}\right) \\
&= \mathbf{I} \\
&= \mathbf{W}\boldsymbol{\Omega};
\end{aligned}
$$

If we multiply the right side of Equation (5.5) by $\left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(k-1)} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right)$, we have

$$
\begin{aligned}
&\left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \mathbf{D} \left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \boldsymbol{\Omega} \\
&\quad \cdot \left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(k-1)} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \\
&= \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \mathbf{D} \\
&\quad \cdot \left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \mathbf{D} \\
&\quad \cdot \left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(k-1)} \otimes \mathbf{I}_{d_k} \otimes \mathbf{U}^{(k+1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \\
&= \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \mathbf{D} \\
&\quad \cdot \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)\top}\mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \left(\sum_{i=1}^{K} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \boldsymbol{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right) \cdot \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)\top} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \\
&= \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \mathbf{D} \\
&\quad \cdot \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)\top} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \left(\sum_{i=1}^{K} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \boldsymbol{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right) \\
&\quad \cdot \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)\top} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \\
&= \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \mathbf{D} \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)\top} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \\
&\quad \cdot \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \boldsymbol{\Psi}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]} + \sum_{i \neq k} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \boldsymbol{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right).
\end{aligned}
$$

□

According to Equation (5.6), we consider the following approximation of $\mathbf{W}$ and $\mathbf{\Omega}$:

$$\hat{\mathbf{W}} = \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right) \mathbf{D} \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)\top} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right),$$

$$\hat{\mathbf{\Omega}} = \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{\Psi}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]} + \sum_{i \neq k} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right).$$

Without loss of generality, consider $k = 1$, then:

$$\hat{\mathbf{W}} = \left(\mathbf{U}^{(1)} \otimes \mathbf{I}_{[d_{2:K}]}\right) \mathbf{D} \left(\mathbf{U}^{(1)\top} \otimes \mathbf{I}_{[d_{2:K}]}\right),$$

$$\hat{\mathbf{\Omega}} = \left(\mathbf{\Psi}^{(1)} \otimes \mathbf{I}_{[d_{2:K}]} + \sum_{i>1} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right).$$

We partition $\hat{\mathbf{W}}$ and $\hat{\mathbf{\Omega}}$ into blocks

$$\hat{\mathbf{W}} = \begin{bmatrix} \hat{\mathbf{W}}_{11} & \hat{\mathbf{W}}_{1\backslash 1} \\ \hat{\mathbf{W}}_{\backslash 11} & \hat{\mathbf{W}}_{\backslash 1\backslash 1} \end{bmatrix},$$

$$\hat{\mathbf{\Omega}} = \begin{bmatrix} \hat{\mathbf{\Omega}}_{11} & \hat{\mathbf{\Omega}}_{1\backslash 1} \\ \hat{\mathbf{\Omega}}_{\backslash 11} & \hat{\mathbf{\Omega}}_{\backslash 1\backslash 1} \end{bmatrix},$$

where $\hat{\mathbf{W}}_{11}$ and $\hat{\mathbf{\Omega}}_{11}$ are $p_1 \times p_1$ matrices and $\hat{\mathbf{W}}_{\backslash 11}$ and $\hat{\mathbf{\Omega}}_{\backslash 11}$ are $p_1 (d_1 - 1) \times p_1$. Note that $\mathbf{I}_{[d_{2:K}]} = \mathbf{I}_{p_1}$, then from the bottom-left block of

$$\hat{\mathbf{W}}\hat{\mathbf{\Omega}} = \hat{\mathbf{W}} \left(\mathbf{\Psi}^{(1)} \otimes \mathbf{I}_{[d_{2:K}]} + \sum_{i>1} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right) = \mathbf{I}_{d_1} \otimes \mathbf{I}_{p_1},$$

we get

$$\begin{aligned} &\hat{\mathbf{W}}_{\backslash 11} \cdot \hat{\mathbf{\Omega}}_{11} + \hat{\mathbf{W}}_{\backslash 1\backslash 1} \hat{\mathbf{\Omega}}_{\backslash 11} \\ &= \hat{\mathbf{W}}_{\backslash 11} \cdot \left(\psi^{(1)}_{11} \mathbf{I}_{p_1} + \sum_{i \neq 1} \mathbf{I}_{[d_{2:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right) + \hat{\mathbf{W}}_{\backslash 1\backslash 1} \cdot \left(\boldsymbol{\psi}_{\backslash 11} \otimes \mathbf{I}_{p_1}\right) \\ &= \mathbf{0}_{d_1 - 1} \otimes \mathbf{I}_{p_1}, \end{aligned}$$

where we use the notation $\mathbf{\Psi}^{(k)} = \left(\psi^{(k)}_{ij}\right)_{i,j=1,\ldots,d_k}$ and $\boldsymbol{\psi}^{(k)}_{\backslash ii}$ representing the corresponding sub-block. Post multiplying both sides of the last equation by $\left(\psi^{(1)}_{11} \mathbf{I}_{p_1} + \sum_{i \neq 1} \mathbf{I}_{[d_{2:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{d_{[(i+1):K]}}\right)$ we have

$$\hat{\mathbf{W}}_{\backslash 11} + \hat{\mathbf{W}}_{\backslash 1\backslash 1} \begin{bmatrix} \left(\psi^{(1)}_{11} \mathbf{I}_{p_1} + \sum_{i \neq 1} \mathbf{I}_{[d_{2:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{d_{[(i+1):K]}}\right)^{-1} \psi^{(1)}_{21} \\ \vdots \\ \left(\psi^{(1)}_{11} \mathbf{I}_{p_1} + \sum_{i \neq 1} \mathbf{I}_{[d_{2:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{d_{[(i+1):K]}}\right)^{-1} \psi^{(1)}_{d_1 1} \end{bmatrix} = \mathbf{0}_{d_1 - 1} \otimes \mathbf{I}_{p_1}. \quad (5.7)$$

Proof of Equation (5.7). In order to prove Equation (5.7), we first note that, from the bottom-left block of $\hat{\mathbf{W}}\hat{\mathbf{\Omega}} = \mathbf{I}_{d_1} \otimes \mathbf{I}_{p_1}$, where

$$\hat{\mathbf{W}} = \begin{bmatrix} \hat{\mathbf{W}}_{11} & \hat{\mathbf{W}}_{1\backslash 1} \\ \hat{\mathbf{W}}_{\backslash 11} & \hat{\mathbf{W}}_{\backslash 1\backslash 1} \end{bmatrix}$$

and

$$\hat{\mathbf{\Omega}} = \begin{bmatrix} \psi_{11}^{(1)}\mathbf{I}_{p_1} + \sum_{i\neq 1} \mathbf{I}_{[d_{2:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]} & \cdots & \psi_{1d_1}^{(1)}\mathbf{I}_{p_1} \\ \vdots & \ddots & \vdots \\ \psi_{d_1 1}^{(1)}\mathbf{I}_{p_1} & \cdots & \psi_{d_1 d_1}^{(1)}\mathbf{I}_{p_1} + \sum_{i\neq 1} \mathbf{I}_{[d_{2:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]} \end{bmatrix},$$

we get

$$\begin{aligned} &\hat{\mathbf{W}}_{\backslash 11} \cdot \hat{\mathbf{\Omega}}_{11} + \hat{\mathbf{W}}_{\backslash 1\backslash 1}\hat{\mathbf{\Omega}}_{\backslash 11} \\ &= \hat{\mathbf{W}}_{\backslash 11} \cdot \left(\psi_{11}^{(1)}\mathbf{I}_{p_1} + \sum_{i\neq 1} \mathbf{I}_{[d_{2:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]} \right) + \hat{\mathbf{W}}_{\backslash 1\backslash 1} \cdot \left(\boldsymbol{\psi}_{\backslash 11} \otimes \mathbf{I}_{p_1} \right) \\ &= \mathbf{0}_{d_1 - 1} \otimes \mathbf{I}_{p_1}. \end{aligned}$$

Thus, multiplying both sides of the last equation by $\left(\psi_{11}^{(1)}\mathbf{I}_{p_1} + \sum_{i\neq 1} \mathbf{I}_{[d_{2:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{d_{[(i+1):K]}} \right)$, one has Equation (5.7). □

Decomposing $\hat{\mathbf{W}}_{\backslash 1\backslash 1}$ in $(d_k - 1)$ adjacent blocks $\mathbf{W}_{\backslash 1k} \in \mathbb{R}^{(d_1-1)p_1 \times p_1}$, $\forall k \in \{2, \ldots, d_1\}$, then Equation (5.7) can be rewritten as

$$\begin{aligned} &\hat{\mathbf{W}}_{\backslash 11} + \hat{\mathbf{W}}_{\backslash 12} \left(\psi_{11}^{(1)}\mathbf{I}_{p_1} + \sum_{i\neq 1} \mathbf{I}_{[d_{2:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]} \right)^{-1} \psi_{21}^{(1)} + \ldots \\ &\cdots + \hat{\mathbf{W}}_{\backslash 1 d_1} \left(\psi_{11}^{(1)}\mathbf{I}_{p_1} + \sum_{i\neq 1} \mathbf{I}_{[d_{2:(i-1)}]} \otimes \mathbf{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]} \right)^{-1} \psi_{p_1 1}^{(1)} = \mathbf{0}_{d_1 - 1} \otimes \mathbf{I}_{p_1}. \end{aligned}$$

Proposition 5.1 Following the assumptions and calculations above we have

$$\mathrm{tr}_{p_k}(\mathbf{W}) = \mathrm{tr}_{p_k}\left(\hat{\mathbf{W}}\right), \forall k = 1, \ldots, K.$$

Proof of Proposition 5.1. Without losing generality, we prove the case where $k = 1$.

Proposition 5.1 becomes $\mathrm{tr}_{p_1}(\mathbf{W}) = \mathrm{tr}_{p_1}(\hat{\mathbf{W}})$, which follows from the fact that

$$
\begin{aligned}
&\left[\mathbf{I}_{d_1} \otimes \mathbf{U}^{(2)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right] \mathbf{W} \left[\mathbf{I}_{d_1} \otimes \mathbf{U}^{(2)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right] \\
&= \left[\mathbf{I}_{d_1} \otimes \mathbf{U}^{(2)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right] \left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \left(\sum_{i=1}^{K} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \boldsymbol{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right)^{-1} \\
&\quad \cdot \left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \left[\mathbf{I}_{d_1} \otimes \mathbf{U}^{(2)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right] \\
&= \left[\mathbf{I}_{d_1}\mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)\top}\mathbf{U}^{(2)} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\mathbf{U}^{(K)}\right] \left(\sum_{i=1}^{K} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \boldsymbol{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right)^{-1} \\
&\quad \cdot \left[\mathbf{U}^{(1)\top}\mathbf{I}_{d_1} \otimes \mathbf{U}^{(2)\top}\mathbf{U}^{(2)} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\mathbf{U}^{(K)}\right] \\
&= \left[\mathbf{U}^{(1)} \otimes \mathbf{I}_{[d_{2:K}]}\right] \mathbf{D} \left[\mathbf{U}^{(1)\top} \otimes \mathbf{I}_{[d_{2:K}]}\right] \\
&= \hat{\mathbf{W}}.
\end{aligned}
$$

Then, the $p_1 \times p_1$ blocks of $\mathbf{W}$ and $\hat{\mathbf{W}}$ hold a similarity relation:

$$\hat{\mathbf{W}}_{ij} = \left(\mathbf{U}^{(2)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right) \mathbf{W}_{ij} \left(\mathbf{U}^{(2)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right)$$

and hence $\mathrm{tr}_{p_1}(\mathbf{W}) = \mathrm{tr}_{p_1}(\hat{\mathbf{W}})$. □

Proposition 5.1 enables us to make use of the stationary point given in Equation (5.3). Similarly to Chapter 4, we can partition the empirical covariance along the mode-1 fibre, $\mathbf{S}^{(1)}$ as

$$\mathbf{S}^{(1)} = \begin{bmatrix} \mathbf{s}^{(1)}_{11} & \mathbf{s}^{(1)}_{1\backslash 1} \\ \mathbf{s}^{(1)}_{\backslash 11} & \mathbf{S}^{(1)}_{\backslash 1\backslash 1} \end{bmatrix},$$

where $\mathbf{s}^{(1)}_{11} \in \mathbb{R}$, $\mathbf{s}^{(1)}_{1\backslash 1} \in \mathbb{R}^{d_k - 1}$, $\mathbf{s}^{(1)}_{\backslash 11} \in \mathbb{R}^{d_k - 1}$, $\mathbf{S}^{(1)}_{\backslash 1\backslash 1} \in \mathbb{R}^{(d_1-1)\times(d_1-1)}$. In particular, from the lower left block of Equation (5.3) we get

$$\mathbf{S}^{(1)} = \frac{1}{p_1} \mathrm{tr}_{p_1}(\mathbf{W}). \tag{5.8}$$

Taking the operation $\mathrm{tr}_{p_1}(\cdot)$ on both sides of Equation (5.7), gives

$$\mathrm{tr}_{p_1}\left(\hat{\mathbf{W}}_{\backslash 11}\right) + \begin{bmatrix} \mathrm{tr}_{p_1}\left\{\hat{\mathbf{W}}_{\backslash 1\backslash 1}\left(\psi^{(1)}_{11}\mathbf{I}_{p_1} + \sum_{i\neq 1} \mathbf{I}_{[d_{2:(i-1)}]} \otimes \boldsymbol{\Lambda}_i \otimes \mathbf{I}_{d_{[(i+1):K]}}\right)^{-1} \psi^{(1)}_{21}\right\} \\ \vdots \\ \mathrm{tr}_{p_1}\left\{\hat{\mathbf{W}}_{\backslash 1\backslash 1}\left(\psi^{(1)}_{11}\mathbf{I}_{p_1} + \sum_{i\neq 1} \mathbf{I}_{[d_{2:(i-1)}]} \otimes \boldsymbol{\Lambda}_i \otimes \mathbf{I}_{d_{[(i+1):K]}}\right)^{-1} \psi^{(1)}_{d_1 1}\right\} \end{bmatrix} = \mathbf{0}. \tag{5.9}$$

From Proposition 5.1 we also have

$$\mathrm{tr}_{p_1}(\mathbf{W}) = \mathrm{tr}_{p_1}(\hat{\mathbf{W}}).$$

Combining the above equation with (5.9) and (5.8), we have:

$$p_1 \mathbf{s}_{\backslash 11}^{(1)} + \mathbf{A}_{\backslash 1\backslash 1}^{(1)} \boldsymbol{\psi}_{\backslash 11} = \mathbf{0}_{d_1 - 1}, \tag{5.10}$$

where

$$\mathbf{A}_{\backslash 1\backslash 1}^{(1)} = \begin{bmatrix} \mathrm{tr}_{p_1}\left\{\hat{\mathbf{W}}_{\backslash 1\backslash 1}\left(\psi_{11}^{(1)}\mathbf{I}_{p_1} + \sum_{i\neq 1}\mathbf{I}_{[d_{2:(i-1)}]} \otimes \boldsymbol{\Lambda}_i \otimes \mathbf{I}_{d_{[(i+1):K]}}\right)^{-1} \psi_{21}^{(1)}\right\}^\top \\ \vdots \\ \mathrm{tr}_{p_1}\left\{\hat{\mathbf{W}}_{\backslash 1\backslash 1}\left(\psi_{11}^{(1)}\mathbf{I}_{p_1} + \sum_{i\neq 1}\mathbf{I}_{[d_{2:(i-1)}]} \otimes \boldsymbol{\Lambda}_i \otimes \mathbf{I}_{d_{[(i+1):K]}}\right)^{-1} \psi_{d_1 1}\right\}^\top \end{bmatrix}. \tag{5.11}$$

The problem posed in Equation (5.10) is addressed via a lasso regression method from Friedman *et al.* (2008). In Proposition 5.2 we use some of the previous decomposition in order to reduce the computational complexity of the problem.

Proposition 5.2 Following the assumptions and calculations above we have

$$\begin{aligned} &\mathrm{tr}_{p_1}\left\{\hat{\mathbf{W}}_{\backslash 1\backslash 1}\left(\psi_{11}^{(1)}\mathbf{I}_{p_1} + \sum_{i\neq 1}\mathbf{I}_{[d_{2:(i-1)}]} \otimes \boldsymbol{\Lambda}_i \otimes \mathbf{I}_{d_{[(i+1):K]}}\right)^{-1} \psi_{k1}^{(1)}\right\}^\top \\ &= \sum_{j_2=1}^{d_2} \cdots \sum_{j_K=1}^{d_K} \frac{1}{\psi_{11}^{(1)} + \lambda_{2j_2} + \cdots + \lambda_{Kj_K}} \begin{bmatrix} \sum_{i=1}^{d_1} \frac{u_{2i}^{(1)} u_{ki}^{(1)}}{\lambda_{1i} + \lambda_{2j_2} + \cdots + \lambda_{Kj_K}} \\ \vdots \\ \sum_{i=1}^{d_1} \frac{u_{d_1 i}^{(1)} u_{ki}^{(1)}}{\lambda_{1i} + \lambda_{2j_2} + \cdots + \lambda_{Kj_K}} \end{bmatrix}, \end{aligned}$$

where $\lambda_{k1}, \ldots, \lambda_{kd_k}$ are the diagonal values of $\boldsymbol{\Lambda}_k \in \mathbb{R}^{d_k \times d_k}$, $\forall k = 1, \ldots, K$.

Proof of Proposition 5.2. To prove Proposition 5.2, we note that

$$\begin{aligned} \hat{\mathbf{W}}_{\backslash 1\backslash 1} &= \left[\mathbf{U}_{\backslash 1}^{(1)} \otimes \mathbf{I}_{p_1}\right] \mathbf{D} \left[\mathbf{U}_{\backslash 1}^{(1)\top} \otimes \mathbf{I}_{p_1}\right] \\ &= \begin{bmatrix} u_{21}\mathbf{I}_{p_1} & \ldots & u_{2d_1}\mathbf{I}_{p_1} \\ \vdots & \ddots & \vdots \\ u_{d_1 1}\mathbf{I}_{p_1} & \ldots & u_{d_1 d_1}\mathbf{I}_{p_1} \end{bmatrix} \mathbf{D} \begin{bmatrix} u_{21}\mathbf{I}_{p_1} & \ldots & u_{d_1 1}\mathbf{I}_{p_1} \\ \vdots & \ddots & \vdots \\ u_{2d_1}\mathbf{I}_{p_1} & \ldots & u_{d_1 d_1}\mathbf{I}_{p_1} \end{bmatrix}, \end{aligned}$$

where $\mathbf{U}_{\backslash 1} \in \mathbb{R}^{(d_1 - 1)\times d_1}$ is the matrix formed by the last $d_1 - 1$ rows of $\mathbf{U}^{(1)}$. Then, we can decompose $\hat{\mathbf{W}}_{\backslash 1\backslash 1}$ in $(d_1 - 1) \times (d_1 - 1)$ blocks $[\hat{\mathbf{W}}_{\backslash 1\backslash 1}]_{\ell,k} \in \mathbb{R}^{p_1 \times p_1}$, with

$$[\hat{\mathbf{W}}_{\backslash 1\backslash 1}]_{\ell,k} = \begin{bmatrix} \sum_{i=1}^{d_1} \frac{u_{\ell i} u_{ki}}{\lambda_{1i} + \lambda_{21} + \cdots + \lambda_{K1}} & \cdots & 0 \\ 0 & \cdots & \sum_{i=1}^{d_1} \frac{u_{\ell i} u_{ki}}{\lambda_{1i} + \lambda_{2d_2} + \cdots + \lambda_{Kd_K}} \end{bmatrix}, \quad \ell, k \in \{2, \ldots, d_1\}.$$

This formulation allows us to write each trace term of Equation (5.10) as

$$\begin{aligned}
&\operatorname{tr}_{p_1}\left\{\hat{\mathbf{W}}_{\setminus 1k}\left(\psi_{11}^{(1)}\mathbf{I}_{p_1}+\sum_{i\neq 1}\mathbf{I}_{[d_{2:(i-1)}]}\otimes\boldsymbol{\Lambda}_i\otimes\mathbf{I}_{d_{[(i+1):K]}}\right)^{-1}\right\}\\
&=\begin{bmatrix}\operatorname{tr}\left\{\hat{\mathbf{W}}_{\setminus 1\setminus 1}\right\}_{1,k}\left(\psi_{11}^{(1)}\mathbf{I}_{p_1}+\sum_{i\neq 1}\mathbf{I}_{[d_{2:(i-1)}]}\otimes\boldsymbol{\Lambda}_i\otimes\mathbf{I}_{d_{[(i+1):K]}}\right)^{-1}\\ \vdots\\ \operatorname{tr}\left\{\hat{\mathbf{W}}_{\setminus 1\setminus 1}\right\}_{(d_1-1),k}\left(\psi_{11}^{(1)}\mathbf{I}_{p_1}+\sum_{i\neq 1}\mathbf{I}_{[d_{2:(i-1)}]}\otimes\boldsymbol{\Lambda}_i\otimes\mathbf{I}_{d_{[(i+1):K]}}\right)^{-1}\end{bmatrix}, \quad k\in\{2,\dots,d_1\}.
\end{aligned}$$

More explicitly,

$$\begin{aligned}
&\operatorname{tr}_{p_1}\left\{\hat{\mathbf{W}}_{\setminus 1k}\left(\psi_{11}^{(1)}\mathbf{I}_{p_1}+\sum_{i\neq 1}\mathbf{I}_{[d_{2:(i-1)}]}\otimes\boldsymbol{\Lambda}_i\otimes\mathbf{I}_{d_{[(i+1):K]}}\right)^{-1}\right\}\\
&=\begin{bmatrix}\sum_{j_2=1}^{d_2}\cdots\sum_{j_K=1}^{d_K}\sum_{i=1}^{d_1}\frac{1}{\psi_{11}^{(1)}+\lambda_{2j_2}+\cdots+\lambda_{Kj_K}}\cdot\frac{u_{2i}^{(1)}u_{ki}^{(1)}}{\lambda_{1i}+\lambda_{2j_2}+\cdots+\lambda_{Kj_K}}\\ \sum_{j_2=1}^{d_2}\cdots\sum_{j_K=1}^{d_K}\sum_{i=1}^{d_1}\frac{1}{\psi_{11}^{(1)}+\lambda_{2j_2}+\cdots+\lambda_{Kj_K}}\cdot\frac{u_{3i}^{(1)}u_{ki}^{(1)}}{\lambda_{1i}+\lambda_{2j_2}+\cdots+\lambda_{Kj_K}}\\ \vdots\\ \sum_{j_2=1}^{d_2}\cdots\sum_{j_K=1}^{d_K}\sum_{i=1}^{d_1}\frac{1}{\psi_{11}^{(1)}+\lambda_{2j_2}+\cdots+\lambda_{Kj_K}}\cdot\frac{u_{d_1i}^{(1)}u_{ki}^{(1)}}{\lambda_{1i}+\lambda_{2j_2}+\cdots+\lambda_{Kj_K}}\end{bmatrix}\\
&=\sum_{j_2=1}^{d_2}\cdots\sum_{j_K=1}^{d_K}\frac{1}{\psi_{11}^{(1)}+\lambda_{2j_2}+\cdots+\lambda_{Kj_K}}\begin{bmatrix}\sum_{i=1}^{d_1}\frac{u_{2i}^{(1)}u_{ki}^{(1)}}{\lambda_{1i}+\lambda_{2j_2}+\cdots+\lambda_{Kj_K}}\\ \vdots\\ \sum_{i=1}^{d_1}\frac{u_{d_1i}^{(1)}u_{ki}^{(1)}}{\lambda_{1i}+\lambda_{2j_2}+\cdots+\lambda_{Kj_K}}\end{bmatrix}.
\end{aligned}$$

□

We note that by imposing an ℓ_1 penalty on $\boldsymbol{\Psi}_{\setminus 11}^{(1)}$, the problem posed in (5.10) reduces to a lasso regression involving now only the matrix $\mathbf{U}$, the diagonal of $\boldsymbol{\Lambda}_1,\dots,\boldsymbol{\Lambda}_K$ and ψ_{11}. This decomposition frees the prohibitive amount of memory needed to store the matrix $\hat{\mathbf{W}}$, which is of size $d_1^2p_1^2$.

The lasso regression will provide an estimation on the first column of $\boldsymbol{\Psi}^{(1)}$. For updating all the other columns $\boldsymbol{\Psi}_{\setminus ii}^{(1)}$ we need to reiterate the same approach. Indeed we partition $\boldsymbol{\Psi}^{(1)}$ into $\psi_{ii}^{(1)}$, $\boldsymbol{\psi}_{\setminus ii}^{(1)}$ and $\boldsymbol{\Psi}_{\setminus i\setminus i}^{(1)}$ for $i=1,\dots,d_1$. We then find a sparse solution of $p_1\mathbf{s}_{\setminus ii}^{(1)}+\mathbf{A}_{\setminus i\setminus i}\boldsymbol{\psi}_{\setminus ii}^{(1)}=\mathbf{0}_{n-1}$ with lasso regression. Hence, after n steps, the columns of $\boldsymbol{\Psi}^{(1)}$ are estimated. Given all the new values $\boldsymbol{\psi}_{\setminus ii}^{(1)}$, $i=1,\dots,d_1$, we then compute the eigenvalues matrix $\boldsymbol{\Lambda}_1$ and eigenvectors matrix $\mathbf{U}^{(1)}$ of $\boldsymbol{\Psi}^{(1)}$. This will

provide the updated values to be used in Proposition 5.2. Similarly the estimation of $\boldsymbol{\Psi}^{(k)}$, $k=2,\ldots,K$, conditionally on fixed $\boldsymbol{\Psi}^{(1)},\ldots,\boldsymbol{\Psi}^{(k-1)},\boldsymbol{\Psi}^{(k+1)},\ldots,\boldsymbol{\Psi}^{(K)}$ becomes directly analogous to the above simply by focusing on the estimated covariance $\mathbf{S}^{(k)}$ along the mode-k fibre and is obtained in d_k steps.

Our approach is summarised in Algorithm 7 for Gaussian data. We point out that the convergence of Algorithm 7 could also be directly verified on the value of the objective function (5.2) at each step. However, due to the computation of $\left|\boldsymbol{\Psi}^{(1)}\oplus\cdots\oplus\boldsymbol{\Psi}^{(K)}\right|$, this becomes unfeasible with bigger K and bigger $d_1,\ldots,d_K$. Indeed, the space complexity can be reduced from $O\left(\prod_{k=1}^{K}d_k^2\right)$ to $O\left(\sum_{k=1}^{K}d_k^2\right)$ by means of Proposition 5.3.

Algorithm 7 scKGLasso

Input: Maximum iteration number N, tolerance ε,
M many observations of $d_1\times\cdots\times d_K$ tensors $\mathcal{Y}_m$, $m=1,\ldots,M$.
Regularization parameters $\beta_1,\ldots,\beta_K$, initial estimates of $\boldsymbol{\Psi}^{(k)}$, denoted as $\boldsymbol{\Psi}^{(k)(0)}$, $k=1,\ldots,K$.

For each $\mathcal{Y}_m$, $k=1,\ldots,K$, perform matricization along mode-k fibre,
obtaining $\mathbf{Y}_m^{(k)}$.

Calculate $\hat{\mathbf{S}}_m^{(k)}\leftarrow p_k^{-1}\mathbf{Y}_m^{(k)}\mathbf{Y}_m^{(k)\top}$.

$\hat{\mathbf{S}}^{(k)}\leftarrow\frac{1}{M}\sum_{m=1}^{M}\hat{\mathbf{S}}_m^{(k)}$

repeat

  for iteration $\tau=1,\ldots,N$ **do**

    Decompose $\boldsymbol{\Psi}^{(k)(\tau-1)}=\mathbf{U}^{(k)(\tau-1)}\boldsymbol{\Lambda}_k^{(\tau-1)}\mathbf{U}^{(k)(\tau-1)\top}$, $k=1,\ldots,K$.

    for $k=1,\ldots,K$ **do**

      # *Estimate* $\boldsymbol{\Psi}^{(k)}$:

      for $i=1,\ldots,d_1$ **do**

        Partition $\boldsymbol{\Psi}^{(k)(\tau-1)}$ into $\psi_{ii}^{(k)(\tau-1)},\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)},\boldsymbol{\psi}_{\setminus ii}^{(k)(\tau-1)}$ and $\boldsymbol{\Psi}_{\setminus i\setminus i}^{(k)(\tau-1)}$.

        Calculate $\mathbf{A}_{\setminus i\setminus i}^{(k)(\tau-1)}$ similar to Proposition 5.2

        with $\psi_{ii}^{(k)(\tau-1)}$, $\mathbf{U}^{(k)(\tau-1)}$, $\boldsymbol{\Lambda}_1^{(k)(\tau-1)}$ and $\boldsymbol{\Lambda}_1^{(k)(\tau-1)}$.

        With Lasso regression (Friedman *et al.*, 2008), find a sparse solution, $\boldsymbol{\psi}_{i\setminus i}^{(k)*}$,

        for $p_k\mathbf{s}_{i\setminus i}^{(k)}+\mathbf{A}_{\setminus i\setminus i}^{(k)(\tau-1)}\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)}=\mathbf{0}_{d_k-1}$.

Calculate the direction vector from
$\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}$ to $\boldsymbol{\psi}_{i\setminus i}^{(1)*}$: $\Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} = \boldsymbol{\psi}_{i\setminus i}^{(k)*} - \boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}$.
Since the objective of solving $p_k \mathbf{s}_{i\setminus i} + \mathbf{A}_{\setminus i\setminus i}^{(k)(\tau-1)} \boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} = \mathbf{0}_{d_k-1}$
can be written as $f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)}\right) = \left\| \mathbf{s}_{i\setminus i} + \frac{1}{p_k} \mathbf{A}_{\setminus i\setminus i}^{(k)(\tau-1)} \boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} \right\|_F^2$,
Calculate $\nabla f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)}\right) = 2\frac{1}{p_k^2} \mathbf{A}_{\setminus i\setminus i}^{(k)(\tau-1)\top} \mathbf{A}_{\setminus i\setminus i}^{(k)(\tau-1)} \boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} + 2\frac{1}{p_k} \mathbf{A}_{\setminus i\setminus i}^{(k)(\tau-1)\top} \mathbf{s}_{i\setminus i}^{(k)}$.
Take $\zeta = \min\left\{\lambda_{k1}, \ldots, \lambda_{kd_k}\right\}$.
 # Implement FISTA (Beck & Teboulle, 2009) with backtracking line search.
Calculate $Q = f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}\right) + \nabla f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}\right)^\top \Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} + \frac{1}{2\zeta} \left\| \Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} \right\|_F^2$ and $f(\boldsymbol{\psi}_{i\setminus i}^{(k)*})$.
$t_0 = 1, a = 0$.
while $f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)*}\right) > Q$ **do**

$$a = a + 1, t_{a+1} = \frac{1 + \sqrt{1 + 4t_a^2}}{2}.$$

$$\boldsymbol{\psi}_{i\setminus i}^{(k)*} = \boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)} + \frac{t_a - 1}{t_{a+1}} \Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)}.$$

$$\Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} = \boldsymbol{\psi}_{i\setminus i}^{(k)*} - \boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}.$$

Calculate $Q = f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}\right) + \nabla f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}\right)^\top \Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} + \frac{1}{2\zeta} \left\| \Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} \right\|_F^2$ and $f(\boldsymbol{\psi}_{i\setminus i}^{(k)*})$.
end while
Update the non-diagonal column $\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} = \boldsymbol{\psi}_{i\setminus i}^{(k)*}$.
end for
Calculate the change in the estimated matrices from each iteration

$$\Delta\boldsymbol{\Psi}^{(k)(\tau)} = \|\boldsymbol{\Psi}^{(k)(\tau)} - \boldsymbol{\Psi}^{(k)(\tau-1)}\|_F^2,$$

end for
end for
until Maximum iteration number reached, or
$\max_{\tau^*=\tau-2,\tau-1,\tau} \left\{\sum_{k=1}^K \Delta\boldsymbol{\Psi}^{(k)(\tau^*)}\right\} < \varepsilon$, for $\tau \geq 3$.

Proposition 5.3 Following the assumptions and calculations above we have

$$\left|\boldsymbol{\Psi}^{(1)} \oplus \cdots \oplus \boldsymbol{\Psi}^{(K)}\right| = \prod_{j_1=1}^{d_1} \cdots \prod_{j_K=1}^{d_K} \left(\lambda_{1j_1} + \cdots + \lambda_{Kj_K}\right).$$

Proof of Proposition 5.3. Proposition 5.3 follows from the fact that

$$\mathbf{W} = \boldsymbol{\Omega}^{-1} = \left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \left(\sum_{i=1}^{K} \mathbf{I}_{[d_{1:(i-1)}]} \otimes \boldsymbol{\Lambda}_i \otimes \mathbf{I}_{[d_{(i+1):K}]}\right)^{-1} \left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right),$$

and

$$\mathbf{D} = \begin{bmatrix} \frac{1}{\sum_{k=1}^{K} \lambda_{k1}} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \frac{1}{\lambda_{11}+\lambda_{22}+\sum_{k=3}^{K} \lambda_{k1}} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \frac{1}{\lambda_{11}+\sum_{k=1}^{K} \lambda_{kd_k}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \frac{1}{\sum_{k=1}^{K} \lambda_{kd_k}} \end{bmatrix},$$

where $\lambda_{k1} \ldots \lambda_{kd_k}$ are the diagonal values of $\boldsymbol{\Lambda}_k \in \mathbb{R}^{d_k \times d_k}$, $k = 1, \ldots, K$. Then, we can write

$$\begin{aligned} \left|\boldsymbol{\Psi}^{(1)} \oplus \cdots \oplus \boldsymbol{\Psi}^{(K)}\right| &= \left|\left(\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right) \mathbf{D}^{-1} \left(\mathbf{U}^{(1)\top} \otimes \cdots \otimes \mathbf{U}^{(K)\top}\right)\right| = \left|\mathbf{U}^{(1)} \otimes \cdots \otimes \mathbf{U}^{(K)}\right|^2 \left|\mathbf{D}^{-1}\right| \\ &= \prod_{k=1}^{K} \left(\left|\mathbf{U}^{(k)}\right|^{2p_k}\right) \cdot \prod_{j_1=1}^{d_1} \cdots \prod_{j_K=1}^{d_K} \left(\lambda_{1j_1} + \cdots + \lambda_{Kj_K}\right) \\ &= \prod_{j_1=1}^{d_1} \cdots \prod_{j_K=1}^{d_K} \left(\lambda_{1j_1} + \cdots + \lambda_{Kj_K}\right). \end{aligned}$$

□

It follows that:

$$\ln\left|\boldsymbol{\Psi}^{(1)} \oplus \cdots \oplus \boldsymbol{\Psi}^{(K)}\right| = \sum_{j_1=1}^{d_1} \cdots \sum_{j_K=1}^{d_K} \ln\left|\lambda_{1j_1} + \cdots + \lambda_{Kj_K}\right| = C.$$

Hence we can write the objective function as

$$\min_{\boldsymbol{\Psi}^{(1)}, \ldots, \boldsymbol{\Psi}^{(K)}} \left\{ \sum_k p_k \mathrm{tr}\left(\boldsymbol{\Psi}^{(k)} \mathbf{S}^{(k)}\right) - C + \sum_k \beta_k \|\boldsymbol{\Psi}^{(k)}\|_1 \right\}.$$

In summary, as we are not interested in the estimation of the overall $\mathbf{W}$ nor $\mathbf{\Omega}$, we will never explicitly update them, but we will rather focus on the estimation of $\mathbf{\Psi}^{(k)}$, $k = 1, \ldots, K$. This leads to a space complexity of $O\left(\sum_{k=1}^{K} d_k^2\right)$ by means of Proposition 5.2 and Proposition 5.3.

Our Scalable K-graphical Lasso algorithm (ScK) benefits from the same statistical convergence properties embedded in the original Bigraphical Lasso model (Kalaitzis *et al.*, 2013) and its extension for K-way tensor data in Greenewald *et al.* (2019). Greenewald *et al.* (2019) gives the statistical convergence rates (Greenewald *et al.*, 2019, Theorems 1-3) (Greenewald *et al.*, 2019, Lemma 19, Supplementary Material) of the Bigraphical Lasso model and its generalisation for K-way tensor data.

5.3 Nonparanormal K-graphical Lasso Model

The method in Section 5.2 only deals with Gaussian data, while in real world many data come in the form of count data. In this Section, we introduce a Gaussian copula based method to adapt Algorithm 7 for count data and other non-Gaussian data. We start by introducing the definition of the tensor nonparanormal distribution with a Kronecker sum structure.

Definition 5.1 Consider a $d_1 \times \cdots \times d_K$ non-Gaussian data tensor $\mathcal{Y}$. $\mathcal{Y}$ follows a tensor nonparanormal distribution with a Kronecker sum structure $\mathbf{TNPN}_{KS}\left(\mathcal{M}; \left\{\left(\mathbf{\Psi}^{(k)}\right)^{-1}\right\}_{k=1}^{K}; f\right)$, with mean tensor $\mathcal{M}$, and where $\mathbf{\Psi}^{(k)}$ is the precision matrix along the mode-k fibre, $\forall k = 1, \ldots, K$, if and only if there exists a set of monotonic transformations $f = \left\{f_{j_1 \ldots j_K}\right\}_{j_k \in \{1,\ldots,d_k\}, k \in \{1,\ldots,K\}}$ such that

$$\operatorname{vec}\left(f\left(\mathcal{Y}\right)\right) \sim \mathbf{mN}\left(\operatorname{vec}(\mathcal{M}), \left(\mathbf{\Psi}^{(1)} \oplus \cdots \oplus \mathbf{\Psi}^{(K)}\right)^{-1}\right).$$

In this chapter, we only consider the model after centering, i.e $\operatorname{vec}(\mathcal{M}) = \mathbf{0}_{\prod_{k=1}^{K} d_k}$. The choices $f_{j_1 \ldots j_K}\left(Y_{j_1 \ldots j_K}\right) = Y_{j_1 \ldots j_K}$ and $f_{j_1 \ldots j_K}\left(Y_{j_1 \ldots j_K}\right) = \ln Y_{j_1 \ldots j_K}$ give us multivariate Normal distribution and multivariate log-Normal distribution respectively. Since we only require f to be monotone, this model provides us with a wider family of distributions to work on, thus extends the K-graphical model to non-Gaussian data. We note that the model in Definition 5.1 can be viewed as a latent model, with latent variable $\mathcal{Z} = f\left(\mathcal{Y}\right)$ and $\operatorname{vec}\left(\mathcal{Z}\right) \sim \mathbf{mN}\left(\mathbf{0}_{\prod_{k=1}^{K}(d_k)}, \left(\mathbf{\Psi}^{(1)} \oplus \cdots \oplus \mathbf{\Psi}^{(K)}\right)^{-1}\right)$.

Following the arguments in Chapter 4 and Kalaitzis *et al.* (2013), the support of $\mathbf{\Psi}^{(k)}$ encodes the dependence structure of variables along the mode-k fibre of $\mathfrak{Y}$, $\forall k = 1,\ldots,K$, respectively. Following the discussion in Section 2.2, $\mathbf{\Psi}^{(1)} \oplus \cdots \oplus \mathbf{\Psi}^{(K)}$ represents the Cartesian product of the Gaussian Markov random field graphs corresponding to every fibres on mode $k = 1,\ldots,K$. Similarly to Subsection 4.4.1, in next subsection we inroduce a method to infer the nonparanormal distribution without explicitly defining f.

5.3.1 Estimation of the precision matrices

We now consider the estimation of the precision matrices $\mathbf{\Psi}^{(k)}$, $k = 1,\ldots,K$. Like the lasso methods applied in one-way network inference and in Gaussian Bigraphical models, we enforce sparsity on $\mathbf{\Psi}^{(k)}$, $k = 1,\ldots,K$ by regularization on the negative log-likelihood, which gives us the objective function:

$$\min_{\mathbf{\Psi}^{(1)},\ldots,\mathbf{\Psi}^{(k)}} \left\{ -\ln|\,\mathbf{\Omega}\,| + \sum_k p_k \mathrm{tr}\left(\mathbf{\Psi}^{(k)}\mathbf{S}^{(k)}\right) + \sum_k \beta_k \|\mathbf{\Psi}^{(k)}\|_1 \right\},$$

where $\mathbf{S}^{(k)} = \frac{1}{p_k}\left(\mathbf{Z}^{(k)}\mathbf{Z}^{(k)\top}\right)$ is the empirical covariance matrix along the mode-k fibre, and $\mathbf{Z}^{(k)}$ is the matricization of $\mathfrak{Z}$ along mode-k fibre. The only problem that remains now is to estimate the empirical covariance matrices $\mathbf{S}^{(k)}$, $k = 1,\ldots,K$. When estimating one-way network, Liu *et al.* (2012) proposed the nonparanormal skeptic, exploiting Kendall's tau or Spearman's rho, without explicitly calculating the marginal transforming function f. Similarly, we define Kendall's tau and Spearman's rho along each mode-k fibre, $k = 1,\ldots,K$. More specifically, let $r^{(k)}_{j_k i}$ be the rank of $Y^{(k)}_{j_k i}$ among $Y^{(k)}_{1i},\ldots,Y^{(k)}_{d_k i}$ and $\bar{r}^{(k)}_i = \frac{1}{d_k}\sum_{j_k=1}^{d_k} r_{j_k i} = \frac{d_k+1}{2}$. Define $\Delta^{(k)}_{j_k}(i,i') = Y^{(k)}_{j_k i} - Y^{(k)}_{j_k i'}$.

We consider the following statistics:

(mode-k-fibre-wise Kendall's tau)

$$\hat{\tau}^{(k)}_{j_k j'_k} = \frac{2}{p_k\left(p_k-1\right)} \sum_{i_1<i_2} \mathrm{sign}\left(\Delta^{(k)}_{j_k}(i_1,i_2)\Delta^{(k)}_{j'_k}(i_1,i_2)\right)$$

(mode-k-fibre-wise Spearman's rho)

$$\hat{\rho}^{(k)}_{j_k j'_k} = \frac{\sum_{i_1=1}^{p_k}\left(r^{(k)}_{j_k i_1} - \bar{r}^{(k)}_{i_1}\right)\left(r^{(k)}_{j_k i_2} - \bar{r}^{(k)}_{i_2}\right)}{\sqrt{\sum_{i_2=1}^{p_k}\left(r^{(k)}_{j_k i_1} - \bar{r}^{(k)}_{i_1}\right)^2\left(r^{(k)}_{j_k i_2} - \bar{r}^{(k)}_{i_2}\right)^2}}.$$

And the following estimated covariance matrices using Kendall's tau and Spearman's rho:

$$\hat{\mathbf{S}}^{(k)}_{j_k j'_k} = \begin{cases} \sin\left(\frac{\pi}{2}\hat{\tau}^{(k)}_{j_k j'_k}\right), & j_k \neq j'_k, \\ 1, & j_k = j'_k. \end{cases} \tag{5.12}$$

$$\hat{\mathbf{S}}^{(k)}_{j_k j'_k} = \begin{cases} 2\sin\left(\frac{\pi}{6}\hat{\rho}^{(k)}_{j_k j'_k}\right), & j_k \neq j'_k, \\ 1, & j_k = j'_k. \end{cases} \tag{5.13}$$

In Algorithm 8 we summarise the Nonparanormal Scalable K-graphical Lasso approach for count data.

Algorithm 8 Nonparanormal scKGLasso

Input: Maximum iteration number N, tolerance ε,
M many observations of $d_1 \times \cdots \times d_K$ tensors $\mathcal{Y}_m$, $m = 1, \ldots, M$.
Regularization parameters $\beta_1, \ldots, \beta_K$, initial estimates of $\mathbf{\Psi}^{(k)}$, denoted as $\mathbf{\Psi}^{(k)(0)}$, $k = 1, \ldots, K$.

For each $\mathcal{Y}_m$, $k = 1, \ldots, K$, perform matricization along mode-k fibre,
obtaining $\mathbf{Y}^{(k)}_m$.

For each $\mathbf{Y}^{(k)}_m$, calculate $\hat{\mathbf{S}}^{(k)}_m$ according to Equation (5.12) or (5.13).

$\hat{\mathbf{S}}^{(k)} \leftarrow \frac{1}{M}\sum_{m=1}^{M} \hat{\mathbf{S}}^{(k)}_m$

repeat

  for iteration $\tau = 1, \ldots, N$ **do**

    Decompose $\mathbf{\Psi}^{(k)(\tau-1)} = \mathbf{U}^{(k)(\tau-1)} \mathbf{\Lambda}^{(\tau-1)}_k \mathbf{U}^{(k)(\tau-1)\top}$, $k = 1, \ldots, K$.

    for $k = 1, \ldots, K$ **do**

      # *Estimate* $\mathbf{\Psi}^{(k)}$:

      for $i = 1, \ldots, d_1$ **do**

        Partition $\mathbf{\Psi}^{(k)(\tau-1)}$ into $\psi^{(k)(\tau-1)}_{ii}$, $\boldsymbol{\psi}^{(k)(\tau-1)}_{i\setminus i}$, $b\psi^{(k)(\tau-1)}_{\setminus ii}$ and $\mathbf{\Psi}^{(k)(\tau-1)}_{\setminus i\setminus i}$.

        Calculate $\mathbf{A}^{(k)(\tau-1)}_{\setminus i\setminus i}$ similar to Proposition 5.2
        with $\psi^{(k)(\tau-1)}_{ii}$, $\mathbf{U}^{(k)(\tau-1)}$, $\mathbf{\Lambda}^{(k)(\tau-1)}_1$ and $\mathbf{\Lambda}^{(k)(\tau-1)}_1$.

        With Lasso regression (Friedman *et al.*, 2008), find a sparse solution, $\boldsymbol{\psi}^{(k)*}_{i\setminus i}$, for $p_k \mathbf{s}^{(k)}_{i\setminus i} + \mathbf{A}^{(k)(\tau-1)}_{\setminus i\setminus i} \boldsymbol{\psi}^{(k)(\tau)}_{i\setminus i} = \mathbf{0}_{d_k-1}$.

Calculate the direction vector from
$\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}$ to $\boldsymbol{\psi}_{i\setminus i}^{(1)*}$: $\Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} = \boldsymbol{\psi}_{i\setminus i}^{(k)*} - \boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}$.
Since the objective of solving $p_k \mathbf{s}_{i\setminus i} + \mathbf{A}_{\setminus i\setminus i}^{(k)(\tau-1)} \boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} = \mathbf{0}_{d_k-1}$
can be written as $f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)}\right) = \left\| \mathbf{s}_{i\setminus i} + \frac{1}{p_k} \mathbf{A}_{\setminus i\setminus i}^{(k)(\tau-1)} \boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} \right\|_F^2$,
Calculate $\nabla f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)}\right) = 2\frac{1}{p_k^2} \mathbf{A}_{\setminus i\setminus i}^{(k)(\tau-1)\top} \mathbf{A}_{\setminus i\setminus i}^{(k)(\tau-1)} \boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} + 2\frac{1}{p_k} \mathbf{A}_{\setminus i\setminus i}^{(k)(\tau-1)\top} \mathbf{s}_{i\setminus i}^{(k)}$.
Take $\zeta = \min\left\{\lambda_{k1}, \ldots, \lambda_{kd_k}\right\}$.
 # Implement FISTA (Beck & Teboulle, 2009) with backtracking line search.
Calculate $Q = f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}\right) + \nabla f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}\right)^\top \Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} + \frac{1}{2\zeta} \left\| \Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} \right\|_F^2$ and $f(\boldsymbol{\psi}_{i\setminus i}^{(k)*})$.
$t_0 = 1, a = 0$.
while $f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)*}\right) > Q$ **do**

$$a = a + 1, t_{a+1} = \frac{1 + \sqrt{1 + 4t_a^2}}{2}.$$

$$\boldsymbol{\psi}_{i\setminus i}^{(k)*} = \boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)} + \frac{t_a - 1}{t_{a+1}} \Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)}.$$

$$\Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} = \boldsymbol{\psi}_{i\setminus i}^{(k)*} - \boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}.$$

Calculate $Q = f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}\right) + \nabla f\left(\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau-1)}\right)^\top \Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} + \frac{1}{2\zeta} \left\| \Delta\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} \right\|_F^2$ and $f(\boldsymbol{\psi}_{i\setminus i}^{(k)*})$.
end while
Update the non-diagonal column $\boldsymbol{\psi}_{i\setminus i}^{(k)(\tau)} = \boldsymbol{\psi}_{i\setminus i}^{(k)*}$.
end for
Calculate the change in the estimated matrices from each iteration

$$\Delta\boldsymbol{\Psi}^{(k)(\tau)} = \|\boldsymbol{\Psi}^{(k)(\tau)} - \boldsymbol{\Psi}^{(k)(\tau-1)}\|_F^2$$

end for
end for
until Maximum iteration number reached, or
$\max_{\tau^*=\tau-2,\tau-1,\tau} \left\{\sum_{k=1}^K \Delta\boldsymbol{\Psi}^{(k)(\tau^*)}\right\} < \varepsilon$, for $\tau \geq 3$.

5.4 Numerical Results

In this section, we implement our Scalable K-graphical Lasso algorithm in `MATLAB` (MATLAB, 2020). After inferring the precision matrices $\mathbf{\Psi}^{(k)}, k = 1,\ldots,K$, these matrices are transformed into binary matrices to reveal the network structures, where any negative value in the precision matrices become 1 and any non-negative value become 0. We illustrate applications of our overall approach on both synthetic and real datasets as described in the following subsections.

5.4.1 Synthetic Gaussian Data

To compare our Scalable K-graphical Lasso algorithm (Algorithm 7) with TeraLasso (Greenewald *et al.*, 2019), we generate $K = 3$ sparse positive definite matrices $\mathbf{\Psi}_0^{(k)} \in \mathbb{R}^{d_k \times d_k}$, $k = 1,\ldots,K$, then we simulate M many $d_1 \times \cdots \times d_K$ Gaussian data $\text{vec}\left(\mathcal{Y}_G^{(m)}\right)$, $m = 1,\ldots,M$, from $\mathbf{mN}\left(\mathbf{0}, \left(\mathbf{\Psi}^{(1)} \oplus \cdots \oplus \mathbf{\Psi}^{(K)}\right)^{-1}\right)$. We plug $\mathcal{Y}_G^{(m)}$, $m = 1,\ldots,M$ into our implemented Algorithm 7 and TeraLasso from Greenewald *et al.* (2019). For network recovery criteria such as Precision, Recall, Accuracy and TPR and FPR, we refer the readers to our definitions in Subsection 4.5.2.

Figure 5.1 shows a comparison between the convergence times and Accuracy of Algorithm 7 and TeraLasso for increasing problem dimensions $d_1 = d_2 = d_3$. We can observe that, while Scalable K-graphical Lasso (scKGLasso) is slower than Teralasso by a fraction, the Accuracy of scKGLasso is significantly higher than TeraLasso when d_k is small, and the Accuracy of scKGLasso is still comparable to TeraLasso when d_k is higher. In fact, even in higher dimensions, the Accuracy of $bPsi^{(1)}$ recovery is still higher than TeraLasso, while the Accuracy of $\mathbf{\Psi}^{(2)}$ and $\mathbf{\Psi}^{(3)}$ are only slightly lower than TeraLasso in higher dimensions. This means that when a tensor dataset has higher dimensions in some modes and lower dimensions in other modes, our scKGLasso might be a more stable choice.

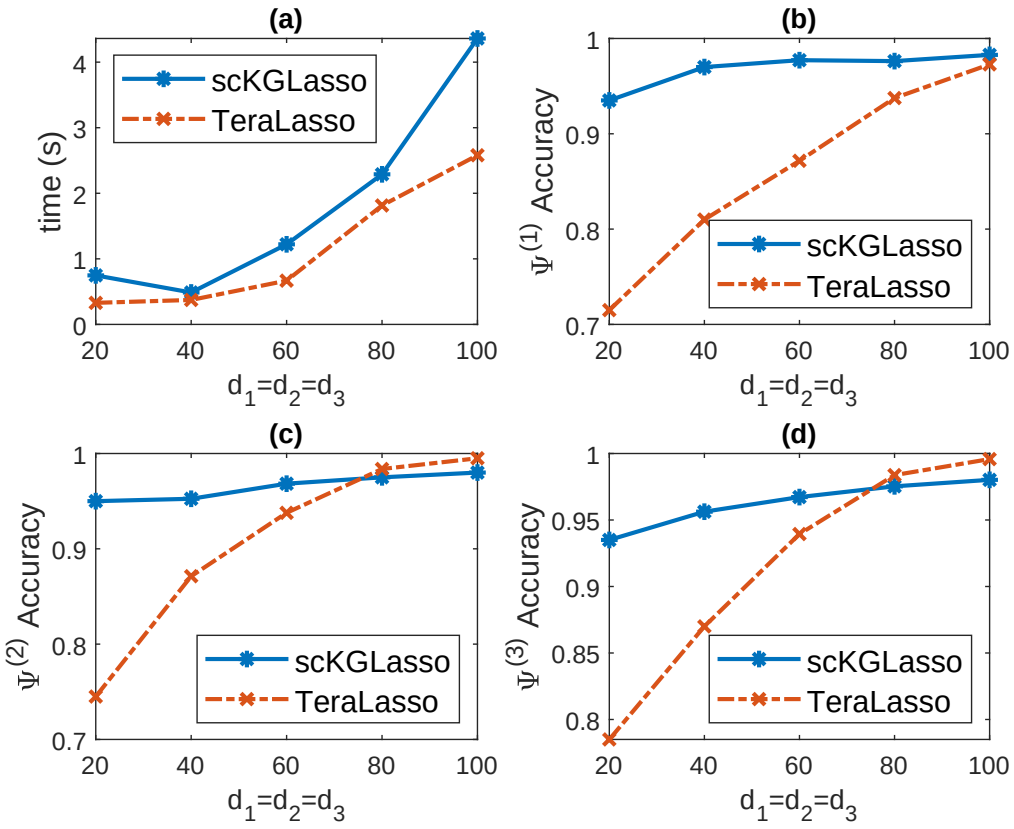

Figure 5.1: Computational convergence time (*seconds*) and accuracy comparison between TeraLasso (Greenewald *et al.*, 2019) and Algorithm 7, for increasing values of the dataset dimensions $d_1 = d_2 = d_3$, $K = 3$.

In Figure 5.2 we present the Precision-Recall curves from synthetic Gaussian data. Figure 5.2 (a) is the Precision-Recall of the recovery of $\mathbf{\Psi}^{(1)}$ with changing β_1 (different points on the graph) and (β_2, β_3) (different colours on the graph). Two arbitrary sets of (β_2, β_3) have been chosen to illustrate how the results do not depend on (β_2, β_3). This is expected as β_1 is the regularization parameter for $\mathbf{\Psi}^{(1)}$, while (β_2, β_3) correspond to $\mathbf{\Psi}^{(2)}, \mathbf{\Psi}^{(3)}$. Similar results are shown in Figure 5.2 (b) and (c), where the Precision-Recall of the recovery of $\mathbf{\Psi}^{(k)}$ heavily depends on the choice of β_k, regardless of the value of other $\beta_l, l \neq k$.

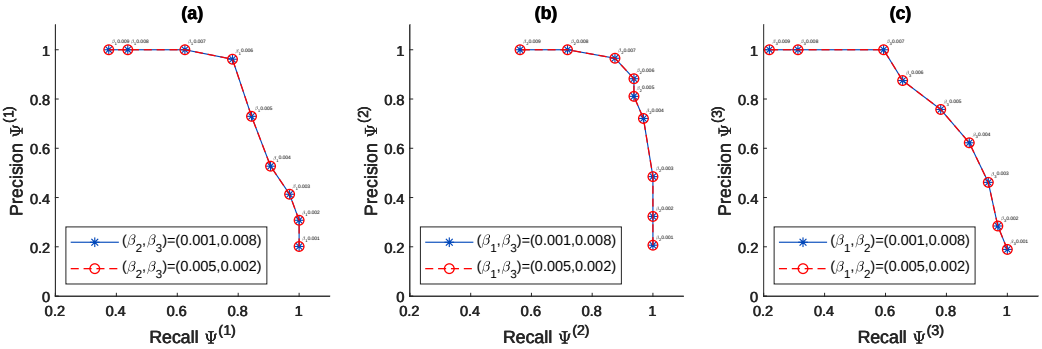

Figure 5.2: Synthetic network recovery results. **(a)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Psi}^{(1)}$; **(b)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Psi}^{(2)}$; **(c)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Psi}^{(3)}$.

Figure 5.3 shows that high values of TPR and Accuracy, with low values of FPR, can be achieved for appropriate choices of β_1, β_2 and β_3 in the range $[0.001, 0.009]$.

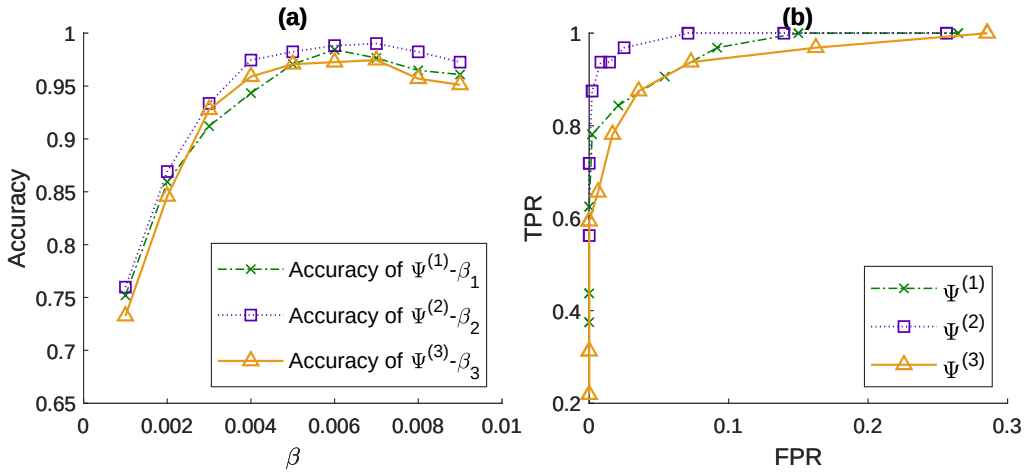

Figure 5.3: Synthetic network recovery results. **(a)** Accuracy vs corresponding regularization parameter β_k of the network recovery relating to the support of $\mathbf{\Psi}^{(k)}$, $k = 1, 2, 3$. **(b)** TPR-FPR of the network recovery relating to the support of $\mathbf{\Psi}^{(k)}$, where the corresponding regularization parameter $\beta_k \in [0.001, 0.009]$.

5.4.2 Synthetic count data

We generate and process Gaussian Copula-based count data through the following steps:

1. Generate sparse positive definite matrix $\mathbf{\Psi}^{(k)}$, $k = 1, \ldots, K$.
2. Perform eigen-decomposition $\mathbf{\Psi}^{(k)} = \mathbf{U}^{(k)\top} \mathbf{\Lambda}_k \mathbf{U}^{(k)}$, $k = 1, \ldots, K$.
3. Calculate $\mathbf{v} = \text{diag}(\mathbf{\Lambda}_1) \otimes \cdots \otimes \text{diag}(\mathbf{\Lambda}_K) \in \mathbb{R}^{\prod_{k=1}^{K} d_k}$. Obtaining $\mathbf{v} = \left(v_1, \ldots, v_{\prod_{k=1}^{K} d_k}\right)$.
4. Generate M vectors of Gaussian samples $\mathbf{x}_m = \left(x_m(1), \ldots, x_m\left(\prod_{k=1}^{K} d_k\right)\right)$, where each $x_m(i) \sim \text{Normal}(0,1)$, $\forall m = 1, \ldots, M$, $\forall i = 1, \ldots, \prod_{k=1}^{K} d_k$.
5. For each $\mathbf{x}_m$, $m = 1, \ldots, M$, let
$$z_m(i) = \frac{x_m(i)}{\sqrt{v_i}}, \ \forall i = 1, \ldots, \prod_{k=1}^{K} d_k.$$
6. For $k = 1, \ldots, K$, repeating
$$\mathbf{z}_m = \left(\mathbf{I}_{[d_{1:(k-1)}]} \otimes \mathbf{U}^{(k)} \otimes \mathbf{I}_{[d_{(k+1):K}]}\right), \ \forall m = 1, \ldots, M.$$
7. Calculate the P_m such that $P_m = \Phi(x_m(i))$, where $\Phi(\cdot)$ is the cumulative density function of the standard normal distribution.
8. For each $m = 1, \ldots, M$, produce the negative binomial variable $y_m(i) = QNB(P_m, r, p)$, where $QNB(\cdot, r, p)$ is the quantile function of Negative-Binomial(r, p), with r the number of success to be observed and p the success rate, resulting in M vectors of count data $\mathbf{y}_m$.
9. Rearrange each $\mathbf{y}_m$ into tensor $\mathcal{Y}_m \in \mathbb{R}^{d_1 \times \cdots \times d_K}$.

We implement Algorithm 8 and plug the synthetic count data in. In Figure 5.4 we present the Precision-Recall curves from synthetic count data. Figure 5.4 (a) is the Precision-Recall of the recovery of $\mathbf{\Psi}^{(1)}$ with changing β_1 (different points on the graph) and (β_2, β_3) (different colours on the graph). Two arbitrary sets of (β_2, β_3) have been chosen to illustrate how the results do not depend on (β_2, β_3). This is expected as β_1 is the regularization parameter for $\mathbf{\Psi}^{(1)}$, while (β_2, β_3) corresponds to $\mathbf{\Psi}^{(2)}, \mathbf{\Psi}^{(3)}$.

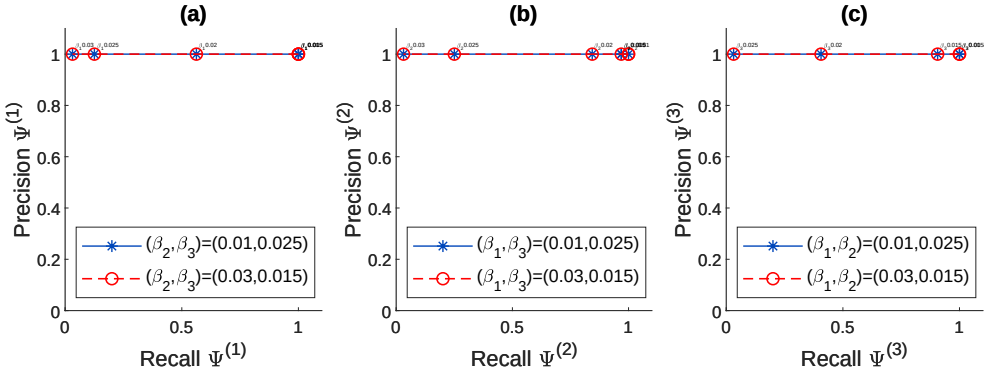

Figure 5.4: Synthetic network recovery results. **(a)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Psi}^{(1)}$; **(b)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Psi}^{(2)}$; **(c)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Psi}^{(3)}$.

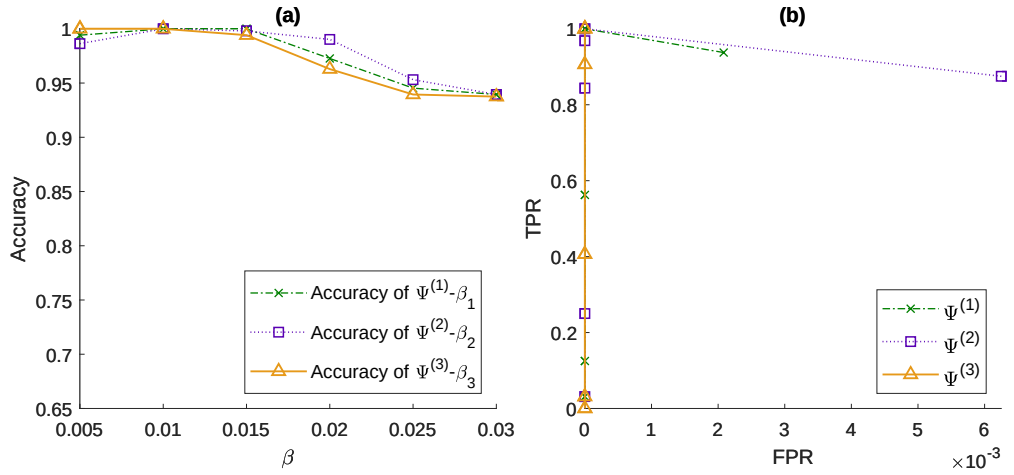

Figure 5.5: Synthetic network recovery results. **(a)** Accuracy vs corresponding regularization parameter β_k of the network recovery relating to the support of $\mathbf{\Psi}^{(k)}$, $k = 1,2,3$. **(b)** TPR-FPR of the network recovery relating to the support of $\mathbf{\Psi}^{(k)}$, where the corresponding regularization parameter $\beta_k \in [0.005, 0.03]$.

Similar results are shown in Figure 5.4 (b) and (c), where the Precision-Recall of the recovery of $\mathbf{\Psi}^{(k)}$ heavily depends on the choice of β_k, regardless of the value of

other $\beta_l, l \neq k$. We note that here the precision of the network recovery stays high, while the recall rate only varies because a bigger value of regularization parameter β_k results in a more sparse network estimation.

Figure 5.5 shows that high values of TPR and Accuracy, with low values of FPR, can be achieved for appropriate choices of β_1, β_2 and β_3 in the range $[0.005, 0.03]$.

5.4.3 An example from the COIL-20 Dataset

In this subsection, our aim is to show the applicability of Algorithm 8 with a real dataset with $K = 3$. In particular, we use frames of several rotating objects from the COIL-20[1] dataset for data analysis. Each frame is a grey-scaled picture, as shown in Figure 5.6.

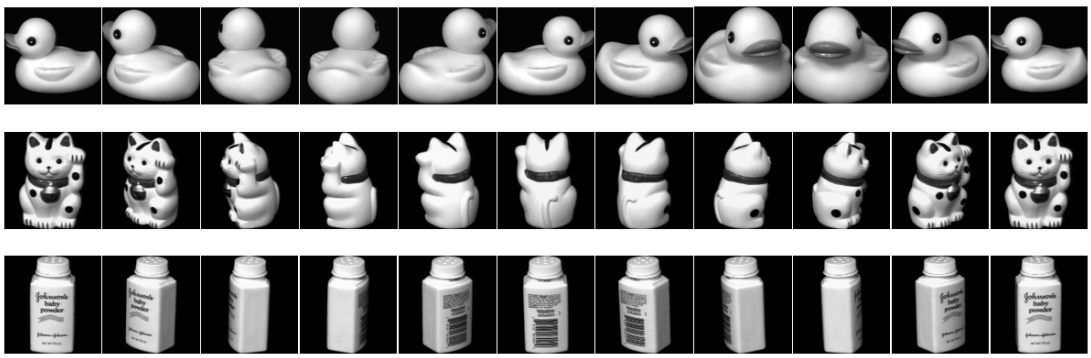

Figure 5.6: First line: frames of a rotating rubber duck from COIL-20 dataset.
Second line: frames of a rotating toy cat from COIL-20 dataset.
Third line: frames of a ratating baby powder bottle from COIL-20 dataset.
Each original frame contained 128 pixels.

We reduced the resolution of each frame from 128×128 to 8×8, and read all the 72 frames for each object. After vectorising each frame (stacking its 8×8 pixels into 64×1 vectors), we rearrange them into 3 matrices with size 64×72, each corresponding to an object. we obtain a 3-way tensor $64 \times 72 \times 3$, $\mathcal{Y}$, by stacking the three matrices together as three slices of the tensor, where mode-1 fibre corresponds to 64 pixels from a frame, mode-2 fibre corresponds to 72 frames for each object, and mode-3 fibre corresponds to 3 objects. Here we aim to test if our model is able to distinguish

[1] https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

the objects while recognising the temporal dependency between each 72 frames. The data were plugged into our Algorithm 8 of Scalable K-graphical Lasso. After inferring the matrix $\mathbf{\Psi}^{(1)}$ (64×64), $\mathbf{\Psi}^{(2)}$ (72×72) and $\mathbf{\Psi}^{(3)}$ (3×3), we use a binary transformation where only the negative values are considered as an edge in the network.

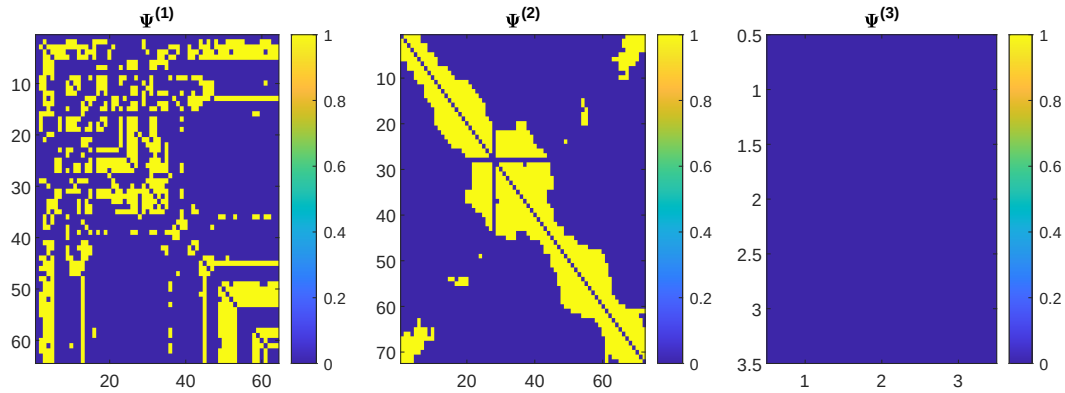

Figure 5.7: Recovered networks of relationships between pixels in frames, between frames and between objects. $\mathbf{\Psi}^{(1)}$ represents the structure in pixels (64 pixels); $\mathbf{\Psi}^{(2)}$ represents the temporal dependencies between frames (72 frames); $\mathbf{\Psi}^{(3)}$ represents the relationship between objects. In this example, we used $(\beta_1, \beta_2, \beta_3) = (0.005, 0.005, 0.2)$.

Figure 5.7 shows the results inferred from the COIL-20 data. The network of pixels ($\mathbf{\Psi}^{(1)}$) shows strong dependencies between the 1st-40nd pixels in intervals of roughly 8, where 8 is the number of pixels we considered in each column of a frame, and the 17th-40th pixels in the subsample of a frame roughly corresponds to where most of the white pixels, i.e. the object itself, are. The network of frames ($\Psi^{(2)}$) shows a clear temporal trajectory of 72 frames, indeed, when we arrange $\mathfrak{Y}$), the mode-2 fibres are the vectorised matrices from the frames of rotating rubber ducks, the vectorised matrices from the frames of rotating toy cats, and the vectorised from the frames of rotating baby powder bottles. The network of objects $\mathbf{\Psi}^{(3)}$ shows no relationship between different objects as we expected. This result shows that our method, the scalable K-graphical Lasso, is able to distinguish objects conditional on the dependencies between pixels and the dependencies in time.

5.5 Conclusion

In this chapter, we present a Scalable K-graphical Lasso algorithm. In particular, we utilised eigenvalue decomposition to simultaneously infer hidden structures in tensor-valued data. Many datasets in different application fields, such as biology, medicine and social science, come as non-Gaussian data, for which Gaussian based models are not applicable. We propose a Gaussian-copula based model and a semi-parametric approach that enables the application of the proposed K-graphical model to tensor-valued non-Gaussian data. Our methodology accounts for the dependencies across different directions in datasets, reduces the computational complexity for high dimensional data and enables us to deal with both discrete and continuous data.

In numerical results we showcase the performance of our method with synthetic and real datasets. we have focused on $K = 3$, but we would expect the algorithm also work for K-way tensor-valued data with a larger K. Our experiment on synthetic Gaussian data shows that, compared to TeraLasso (Greenewald *et al.*, 2019), our method gives better accuracy when dataset dimensions $d_k,\ k = 1,\ldots,K$ are small, while its accuracy is still comparable with TeraLasso when the dataset dimensions $d_k,\ k = 1,\ldots,K$ are larger. This indicates that our method is more stable especially when dealing with tensor data with small d_k along some modes and large d_k along other modes. Further experiments on synthetic count data show that our method can work well on tensor-valued count data. Our real data example is a continuation from the discussion in Subsection 4.5.3, and it shows our approach can infer the hidden structures in tensor-valued real data along different modes.

Chapter 6

Concluding Remarks

In this chapter we summarise the main contributions of this thesis and discuss some potential directions for future work.

This thesis has two aims. The first is to explore a simulation-based evaluation framework for Bayesian Hierarchical Models (BHMs) for scRNAseq data. The second is to introduce a novel and efficient inference algorithm for multi-way network inference for Gaussian and non-Gaussian data.

Regarding the first goal, in Chapter 3, we have illustrated a simulation-based evaluation framework for BHMs for scRNAseq. We explore the reliability of a non-Gaussian distribution based BHM inferred via the Monte Carlo Markov Chain (MCMC) algorithm, using the BASiCS framework developed by Vallejos *et al.* (2015, 2016) and by Eling *et al.* (2018) as an example. From our experiments, both the posterior median and the posterior mean are revealed to be inaccurate point estimates for model parameters at times, showing the limitations of considering point estimates from posterior distributions for downstream analysis, when considering BHMs for scRNAseq. We also show that for a fixed given model, the effect of a contaminated prior distribution on the posteriors varies. For the purpose of this experiment on contaminated prior distribution, we modified the `BASiCS` package from Vallejos *et al.* (2015, 2016) and Eling *et al.* (2018), providing the choice of a mixed prior on a spectrum. We also implemented two validation methods for Bayesian models, namely the Posterior Predictive Check (Gelman *et al.*, 1996; Rubin, 1984) and Simulation based calibration (Talts *et al.*, 2018), specifically for BASiCS framework in `R` (R Core Team, 2013). From our experiments, we identified that a parameter, namely the

global technical noise parameter θ in BASiCS framework, is consistently underestimated, thereby suggesting the future direction for the improvement of the BASiCS framework.

As for our second goal in this thesis, firstly, in Chapter 2, we formally give and proved Theorem 2.2 as the theoretical basis of the Bigraphical model proposed by Kalaitzis *et al.* (2013), thereby closing a theoretical gap in the Bigraphical model and its extension to multi-way graphical model (Greenewald, 2017) based on Gaussian Markov random fields. Secondly, in Chapter 3, we developed the Scalable Bigraphical Lasso, a novel algorithm for simultaneous inference of two-way networks from matrix valued data, which exploits eigen-decomposition and matrix algebra in order to improve the computational efficiency with respect to the original Bigraphical Lasso, which allows one to tackle bigger datasets and problems. Moreover, by introducing a Gaussian copula approach, we enabled the two-way graphical models based on Gaussian Markov random fields to be applied to non-Gaussian data, which are common in real world applications. Our experiment on synthetic Gaussian data shows that, compared to the past methods in the literature (Greenewald *et al.*, 2019; Kalaitzis *et al.*, 2013), our method performs better in terms of computational efficiency while still maintaining high accuracy. Our experiment on synthetic count data shows that our method, which exploits the Gaussian copula transformation, can successfully infer hidden structures from non-Gaussian data. We also illustrate the broad applicability of our method with real data examples of image clustering and scRNAseq gene expression data analysis. Last but not least, we developed the Scalable K-graphical Lasso, leveraging eigen-decomposition and matrix algebra in order to carry out simultaneous inference of multi-way networks from tensor-valued data. We also introduce a Gaussian copula approach to extend our method to structure discovery for non-Gaussian tensor-valued data. Our experiment on synthetic Gaussian data shows that, compared to TeraLasso (Greenewald *et al.*, 2019), our method performs significantly better when the tensor size is small, while when the tensor size is large, our method's performance is still comparable with TeraLasso. This shows that for the tensor-valued real data which varies in sizes on different dimensions, our method could be a safer choice. Furthermore, we show the applicability of our method on non-Gaussian data, which is lacking in previous methods, via experiments on synthetic count data and real data.

The work presented in this thesis leads us to more interesting questions for future research. From the evaluation experiement for BHMs on scRNAseq data in Chapter 3, we could further investigate some other choices of point estimate from posterior samples, and we could also study alternative inference algorithms to deal with the problem of underestimation for certain parameters, which is highlighted in our experiments. Furthermore, future work could be focused on enhancing the multi-way network inference algorithm, presented in Chapter 5, to improve efficiency when dealing with high dimensional data. Finally, we could explore the applicability of our proposed methodologies in alternative application fields, such as neuroscience and traffic sciences.

References

ACHARYA, B.D. (1980). Spectral criterion for cycle balance in networks. *Journal of Graph Theory*, **4**, 1–11. 10

AHN, D., JANG, J.G. & KANG, U. (2022). Time-aware tensor decomposition for sparse tensors. *Machine Learning*, **111**, 1409–1430. 2

AKAIKE, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected papers of hirotugu akaike*, 199–213, Springer. 104

ALMET, A.A., CANG, Z., JIN, S. & NIE, Q. (2021). The landscape of cell–cell communication through single-cell transcriptomics. *Current opinion in systems biology*, **26**, 12–23. 2

BABALEYE, A.O., KURT, R.E. & KHAN, F. (2019). Hierarchical bayesian model for failure analysis of offshore wells during decommissioning and abandonment processes. *Process Safety and Environmental Protection*, **131**, 307–319. 7

BAKER, S., BAUER, S., BEYER, R., BRENTON, J., BROMLEY, B., BURRILL, J., CAUSTON, H., CONLEY, M., ELESPURU, R., FERO, M. *et al.* (2005). The external rna controls consortium: a progress report. *Nature Methods*, **2**, 731–734. 25

BARTLETT, T.E., KOSMIDIS, I. & SILVA, R. (2021). Two-way sparsity for time-varying networks with applications in genomics. *The Annals of Applied Statistics*, **15**, 856–879. 78

BAYES, T. (1763). An essay toward solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFRS. *Philosophical Transactions*, 1683–1775. 6

BECK, A. & TEBOULLE, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, **2**, 183–202. 89, 94, 119, 124

BHARGAVA, V., KO, P., WILLEMS, E., MERCOLA, M. & SUBRAMANIAM, S. (2013). Quantitative transcriptomics using designed primer-based amplification. *Scientific reports*, **3**, 1–9. 24

BIJMA, F., DE MUNCK, J.C. & HEETHAAR, R.M. (2005). The spatiotemporal meg covariance matrix modeled as a sum of kronecker products. *NeuroImage*, **27**, 402–415. 77

BLACKBURN, G.M., GAIT, M.J., LOAKES, D., WILLIAMS, D.M., FLAVELL, A., EGLI, M., WILSON, W.D., ALLEN, S., PYLE, A.M., FISHER, J. *et al.* (2006). *Nucleic acids in chemistry and biology*. Royal Society of Chemistry. 23

BRENNECKE, P., ANDERS, S., KIM, J.K., KOŁODZIEJCZYK, A.A., ZHANG, X., PROSERPIO, V., BAYING, B., BENES, V., TEICHMANN, S.A., MARIONI, J.C. *et al.* (2013). Accounting for technical noise in single-cell rna-seq experiments. *Nature methods*, **10**, 1093–1095. 23, 25, 32

BUETTNER, F., NATARAJAN, K.N., CASALE, F.P., PROSERPIO, V., SCIALDONE, A., THEIS, F.J., TEICHMANN, S.A., MARIONI, J.C. & STEGLE, O. (2015). Computational analysis of cell-to-cell heterogeneity in single-cell rna-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology*, **33**, 155–160. 102

CANG, Z. & NIE, Q. (2020). Inferring spatial and signaling relationships between cells from single cell transcriptomic data. *Nature communications*, **11**, 1–13. 2

CHIQUET, J., ROBIN, S. & MARIADASSOU, M. (2019). Variational inference for sparse network reconstruction from count data. In *International Conference on Machine Learning*, 1162–1171, PMLR. 78

CONGDON, P. (2014). *Applied bayesian modelling*. John Wiley & Sons. 22

COSTA, G. & ORTALE, R. (2012). A bayesian hierarchical approach for exploratory analysis of communities and roles in social networks. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 194–201, IEEE. 7

CSARDI, G., NEPUSZ, T. *et al.* (2006). The igraph software package for complex network research. *InterJournal, complex systems*, **1695**, 1–9. 103

CVETKOVIĆ, D., DOOB, M. & SACHS, H. (1979). *Spectra of Graphs, Series in Pure and Applied Math. 87*. Academic Press. 12, 14

DASGUPTA, A., SELF, S.G. & GUPTA, S.D. (2007). Non-identifiable parametric probability models and reparametrization. *Journal of statistical planning and inference*, **137**, 3380–3393. 8

DENNIS, G., SHERMAN, B.T., HOSACK, D.A., YANG, J., GAO, W., LANE, H.C. & LEMPICKI, R.A. (2003). David: database for annotation, visualization, and integrated discovery. *Genome biology*, **4**, 1–11. 102

DEVONSHIRE, A.S., ELASWARAPU, R. & FOY, C.A. (2010). Evaluation of external rna controls for the standardisation of gene expression biomarker measurements. *BMC genomics*, **11**, 1–15. 25

DONDELINGER, F., LÈBRE, S. & HUSMEIER, D. (2013). Non-homogeneous dynamic bayesian networks with bayesian regularization for inferring gene regulatory networks with gradually time-varying structure. *Machine Learning*, **90**, 191–230. 7

ELING, N., RICHARD, A.C., RICHARDSON, S., MARIONI, J.C. & VALLEJOS, C.A. (2018). Correcting the mean-variance dependency for differential variability testing using single-cell rna sequencing data. *Cell systems*, **7**, 284–294. xiv, 1, 4, 22, 23, 25, 27, 33, 44, 45, 56, 62, 67, 75, 133

EXTERNAL-RNA-CONTROLS-CONSORTIUM (2005). Proposed methods for testing and selecting the ercc external rna controls. *BMC genomics*, **6**, 150. 25

FAN, H.C., FU, G.K. & FODOR, S.P. (2015). Combinatorial labeling of single cells for gene expression cytometry. *Science*, **347**, 1258367. 24

FANG, Z., MA, T., TANG, G., ZHU, L., YAN, Q., WANG, T., CELEDÓN, J.C., CHEN, W. & TSENG, G.C. (2018). Bayesian integrative model for multi-omics data with missingness. *Bioinformatics*, **34**, 3801–3808. 7

FDA, U. (2006). Innovation or stagnation: critical path opportunities list. 25

FODOR, S., RAVA, R.P., HUANG, X.C., PEASE, A.C., HOLMES, C.P. & ADAMS, C.L. (1993). Multiplexed biochemical assays with biological chips. *Nature*, **364**, 555–556. 25

FODOR, S.P., READ, J.L., PIRRUNG, M.C., STRYER, L., LU, A.T. & SOLAS, D. (1991). Light-directed, spatially addressable parallel chemical synthesis. *science*, **251**, 767–773. 25

FRIEDMAN, J., HASTIE, T. & TIBSHIRANI, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**, 432–441. 80, 82, 88, 116, 118, 123

GELMAN, A., MENG, X.L. & STERN, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica sinica*, 733–760. 38, 39, 76, 133

GELMAN, A., CARLIN, J.B., STERN, H.S., DUNSON, D.B., VEHTARI, A. & RUBIN, D.B. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC. 39, 66

GIRVAN, M. & NEWMAN, M.E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, **99**, 7821–7826. 11

GREENEWALD, K. (2017). *High Dimensional Covariance Estimation for Spatio-Temporal Processes*. Ph.D. thesis, University of Michigan. 2, 134

GREENEWALD, K., ZHOU, S. & HERO III, A. (2019). Tensor graphical lasso (teralasso). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **81**, 901–931. vi, xv, 2, 3, 4, 14, 77, 91, 92, 95, 96, 107, 108, 121, 125, 126, 132, 134

GREENWOOD, M. & YULE, G.U. (1920). An inquiry into the nature of frequency distributions representative of multiple happenings with particular reference to the occurrence of multiple attacks of disease or of repeated accidents. *Journal of the Royal statistical society*, **83**, 255–279. 31

HARARY, F. (1962). The determinant of the adjacency matrix of a graph. *Siam Review*, **4**, 202–210. 10

HARTIG, F., MINUNNO, F., PAUL, S., CAMERON, D., OTT, T. & PICHLER, M. (2019). *BayesianTools: General-purpose MCMC and SMC samplers and tools for Bayesian statistics*. Version 0.1.7. 41

HASHIMSHONY, T., WAGNER, F., SHER, N. & YANAI, I. (2012). Cel-seq: single-cell rna-seq by multiplexed linear amplification. *Cell reports*, **2**, 666–673. 24

HEID, C.A., STEVENS, J., LIVAK, K.J. & WILLIAMS, P.M. (1996). Real time quantitative pcr. *Genome research*, **6**, 986–994. 25

HIGUCHI, R., FOCKLER, C., DOLLINGER, G. & WATSON, R. (1993). Kinetic pcr analysis: real-time monitoring of dna amplification reactions. *Bio/technology*, **11**, 1026–1030. 25

HUMPHREY, T. & BROOKS, G. (2008). *Cell cycle control: mechanisms and protocols*, vol. 296. Springer Science & Business Media. xv, 102

ISLAM, S., KJÄLLQUIST, U., MOLINER, A., ZAJAC, P., FAN, J.B., LÖNNERBERG, P. & LINNARSSON, S. (2011). Characterization of the single-cell transcriptional landscape by highly multiplex rna-seq. *Genome research*, **21**, 1160–1167. 24

ISLAM, S., KJÄLLQUIST, U., MOLINER, A., ZAJAC, P., FAN, J.B., LÖNNERBERG, P. & LINNARSSON, S. (2012). Highly multiplexed and strand-specific single-cell rna 5' end sequencing. *Nature protocols*, **7**, 813–828. 24

ISLAM, S., ZEISEL, A., JOOST, S., LA MANNO, G., ZAJAC, P., KASPER, M., LÖNNERBERG, P. & LINNARSSON, S. (2014). Quantitative single-cell rna-seq with unique molecular identifiers. *Nature methods*, **11**, 163–166. 24

JAITIN, D.A., KENIGSBERG, E., KEREN-SHAUL, H., ELEFANT, N., PAUL, F., ZARETSKY, I., MILDNER, A., COHEN, N., JUNG, S., TANAY, A. *et al.* (2014). Massively parallel single-cell rna-seq for marker-free decomposition of tissues into cell types. *Science*, **343**, 776–779. 24

JIA, B., XU, S., XIAO, G., LAMBA, V. & LIANG, F. (2017). Learning gene regulatory networks from next generation sequencing data. *Biometrics*, **73**, 1221–1230. 78

KALAITZIS, A., LAFFERTY, J., LAWRENCE, N.D. & ZHOU, S. (2013). The bigraphical lasso. In *International Conference on Machine Learning*, 1229–1237, PMLR. vi, xiv, 2, 3, 14, 77, 78, 79, 81, 82, 86, 91, 92, 95, 96, 106, 121, 122, 134

KINDERMANN, R. (1980). Markov random fields and their applications. *American mathematical society*. 10, 12

KNAUER, U. & KNAUER, K. (2019). *Algebraic graph theory*. de Gruyter. 11

KOLDA, T.G. & BADER, B.W. (2009). Tensor decompositions and applications. *SIAM review*, **51**, 455–500. 17

KOLMOGOROV, A. (1933). Sulla determinazione empirica di una lgge di distribuzione. *Inst. Ital. Attuari, Giorn.*, **4**, 83–91. 41

LAPLACE, P.S. (1814). Essai philosophique sur les probabilités (1814). *Printed as a preface to Théorie analytique des probabilités in the Oeuvres Complètes edition, 1921*, **7**. 6

LAURITZEN, S.L. (1996). *Graphical models*, vol. 17. Clarendon Press. 10, 13, 78

LAWSON, A.B. (2018). *Bayesian disease mapping: hierarchical modeling in spatial epidemiology*. Chapman and Hall/CRC. 7

LIU, H., LAFFERTY, J. & WASSERMAN, L. (2009). The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, **10**. 19, 20, 78

LIU, H., HAN, F., YUAN, M., LAFFERTY, J. & WASSERMAN, L. (2012). High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics*, **40**, 2293–2326. 19, 78, 79, 80, 92, 122

LOCKHART, D.J., DONG, H., BYRNE, M.C., FOLLETTIE, M.T., GALLO, M.V., CHEE, M.S., MITTMANN, M., WANG, C., KOBAYASHI, M., NORTON, H. *et al.* (1996). Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature biotechnology*, **14**, 1675–1680. 25

LOVE, M.I., HUBER, W. & ANDERS, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, **15**, 1–21. 33

MACOSKO, E.Z., BASU, A., SATIJA, R., NEMESH, J., SHEKHAR, K., GOLDMAN, M., TIROSH, I., BIALAS, A.R., KAMITAKI, N., MARTERSTECK, E.M. *et al.* (2015). Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, **161**, 1202–1214. 24

MAGNUS, J.R. & NEUDECKER, H. (1999). *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons. 15

MAKOWSKI, D., BEN-SHACHAR, M.S. & LÜDECKE, D. (2019). bayestestr: Describing effects and their uncertainty, existence and significance within the bayesian framework. *Journal of Open Source Software*, **4**, 1541. 45, 46

MARGULIES, M., EGHOLM, M., ALTMAN, W.E., ATTIYA, S., BADER, J.S., BEMBEN, L.A., BERKA, J., BRAVERMAN, M.S., CHEN, Y.J., CHEN, Z. *et al.* (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, **437**, 376–380. 24

MASSEY JR, F.J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, **46**, 68–78. 41

MATLAB (2020). *(R2020a)*. The MathWorks Inc., Natick, Massachusetts. 95, 125

MCDAVID, A., GOTTARDO, R., SIMON, N. & DRTON, M. (2019). Graphical models for zero-inflated single cell gene expression. *The annals of applied statistics*, **13**, 848. 78

MISHRA, M., MARTINSSON, J., RANTATALO, M. & GOEBEL, K. (2018). Bayesian hierarchical model-based prognostics for lithium-ion batteries. *Reliability Engineering & System Safety*, **172**, 25–35. 7

NAKAGAWA, T. & HASHIMOTO, S. (2020). Robust bayesian inference via γ-divergence. *Communications in Statistics-Theory and Methods*, **49**, 343–360. 24

NING, Y. & LIU, H. (2013). High-dimensional semiparametric bigraphical models. *Biometrika*, **100**, 655–670. 80

PAULINO, C.D.M. & DE BRAGANÇA PEREIRA, C.A. (1994). On identifiability of parametric statistical models. *Journal of the Italian Statistical Society*, **3**, 125–151. 8

PEAT, J.R., DEAN, W., CLARK, S.J., KRUEGER, F., SMALLWOOD, S.A., FICZ, G., KIM, J.K., MARIONI, J.C., HORE, T.A. & REIK, W. (2014). Genome-wide bisulfite sequencing in zygotes identifies demethylation targets and maps the contribution of tet3 oxidation. *Cell reports*, **9**, 1990–2000. 33

PETERSEN, K.B., PEDERSEN, M.S. *et al.* (2008). The matrix cookbook. *Technical University of Denmark*, **7**, 510. 15

PFEIFFER, P.E. (2013). *Concepts of probability theory*. Courier Corporation. 6

POISSON, S.D. (1837). *Recherches sur la probabilité des jugements en matière criminelle et en matière civile*. Bachelier. 27

R CORE TEAM (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. 44, 76, 133

RAGHAVAN, U.N., ALBERT, R. & KUMARA, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, **76**, 036106. 102

RENTELN, P. (2013). *Manifolds, tensors, and forms: an introduction for mathematicians and physicists*. Cambridge University Press. 16

ROBERTS, G.O. & ROSENTHAL, J.S. (2009). Examples of adaptive mcmc. *Journal of computational and graphical statistics*, **18**, 349–367. 9

ROTHENBERG, T.J. (1971). Identification in parametric models. *Econometrica: Journal of the Econometric Society*, 577–591. 8

ROY, A. & DUNSON, D.B. (2020). Nonparametric graphical model for counts. *Journal of Machine Learning Research*, **21**, 1–21. 78

RUBIN, D.B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, 1151–1172. 38, 76, 133

RUE, H. & HELD, L. (2005). *Gaussian Markov random fields: theory and applications*. Chapman and Hall/CRC. 13

SABIDUSSI, G. (1959a). The composition of graphs. *Duke Mathematical Journal*, **26**, 693–696. 11

SABIDUSSI, G. (1959b). Graph multiplication. *Mathematische Zeitschrift*, **72**, 446–457. 79

SASAGAWA, Y., NIKAIDO, I., HAYASHI, T., DANNO, H., UNO, K.D., IMAI, T. & UEDA, H.R. (2013). Quartz-seq: a highly reproducible and sensitive single-cell rna sequencing method, reveals non-genetic gene-expression heterogeneity. *Genome biology*, **14**, 1–17. 24

SCANNELL, C.M., CHIRIBIRI, A., VILLA, A.D., BREEUWER, M. & LEE, J. (2020). Hierarchical bayesian myocardial perfusion quantification. *Medical image analysis*, **60**, 101611. 7

SCHAD, D.J., BETANCOURT, M. & VASISHTH, S. (2021). Toward a principled bayesian workflow in cognitive science. *Psychological methods*, **26**, 103. 71

SCHENA, M., SHALON, D., DAVIS, R.W. & BROWN, P.O. (1995). Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, **270**, 467–470. 25

SKLAR, A. (1973). Random variables, joint distribution functions, and copulas. *Kybernetika*, **9**, 449–460. 18

SMIRNOV, N. (1939). On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Math. Univ. Moscou*, **2**, 3–14. 41

STATISTICAT & LLC. (2021). *LaplacesDemon: Complete Environment for Bayesian Inference*. R package version 16.1.6. 72

SVENSSON, V., GAYOSO, A., YOSEF, N. & PACHTER, L. (2020). Interpretable factor models of single-cell rna-seq via variational autoencoders. *Bioinformatics*, **36**, 3418–3421. 2

TALTS, S., BETANCOURT, M., SIMPSON, D., VEHTARI, A. & GELMAN, A. (2018). Validating bayesian inference algorithms with simulation-based calibration. *arXiv preprint arXiv:1804.06788*. 23, 39, 40, 41, 42, 71, 72, 73, 74, 76, 133

TANG, F., BARBACIORU, C., WANG, Y., NORDMAN, E., LEE, C., XU, N., WANG, X., BODEAU, J., TUCH, B.B., SIDDIQUI, A. *et al.* (2009). mrna-seq whole-transcriptome analysis of a single cell. *Nature methods*, **6**, 377–382. 22, 24

TENG, S.L. & HUANG, H. (2009). A statistical framework to infer functional gene relationships from biologically interrelated microarray experiments. *Journal of the American Statistical Association*, **104**, 465–473. 77

TSILIGKARIDIS, T. & HERO, A.O. (2013). Covariance estimation in high dimensions via kronecker product expansions. *IEEE Transactions on Signal Processing*, **61**, 5347–5360. 77

VALLEJOS, C.A., MARIONI, J.C. & RICHARDSON, S. (2015). Basics: Bayesian analysis of single-cell sequencing data. *PLoS computational biology*, **11**, e1004333. 1, 4, 22, 23, 25, 26, 27, 28, 44, 45, 75, 133

VALLEJOS, C.A., RICHARDSON, S. & MARIONI, J.C. (2016). Beyond comparisons of means: understanding changes in gene expression at the single-cell level. *Genome biology*, **17**, 1–14. xiv, 1, 4, 22, 23, 25, 26, 27, 28, 44, 45, 51, 67, 68, 70, 74, 75, 133

WANG, Y., JANG, B. & HERO, A. (2020). The sylvester graphical lasso (syglasso). In *International Conference on Artificial Intelligence and Statistics*, 1943–1953, PMLR. 107

WHITTAKER, J. (1990). *Graphical models in applied multivariate statistics*. Wiley Publishing. 10

WITTWER, C.T., HERRMANN, M.G., MOSS, A.A. & RASMUSSEN, R.P. (1997). Continuous fluorescence monitoring of rapid cycle dna amplification. *Biotechniques*, **22**, 130–138. 25

XIONG, L., CHEN, X., HUANG, T.K., SCHNEIDER, J. & CARBONELL, J.G. (2010). Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM international conference on data mining*, 211–222, SIAM. 107

XU, P., ZHANG, T. & GU, Q. (2017). Efficient algorithm for sparse tensor-variate gaussian graphical models via gradient descent. In *Artificial Intelligence and Statistics*, 923–932, PMLR. 107

YOSHIOKA, E., HANLEY, S., SATO, Y. & SAIJO, Y. (2022). Associations between social fragmentation, socioeconomic deprivation and suicide risk across 1887 municipalities in japan, 2009–2017: a spatial analysis using the bayesian hierarchical model. *BMJ open*, **12**, e063255. 7

ZEISEL, A., MUÑOZ-MANCHADO, A.B., CODELUPPI, S., LÖNNERBERG, P., LA MANNO, G., JURÉUS, A., MARQUES, S., MUNGUBA, H., HE, L., BETSHOLTZ, C. *et al.* (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, **347**, 1138–1142. 44

ZHOU, S. (2014). Gemini: Graph estimation with matrix variate normal instances. *The Annals of Statistics*, **42**, 532–562. 77