Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature: Date

_____ _____

Ziyi Li

Statistical Learning Methods for Big Biomedical Data

By

Ziyi Li

Doctor of Philosophy

Biostatistics

Qi Long, Ph.D.
Advisor

Xiaoqian Jiang, Ph.D.
Committee Member

Jeanne Kowalski, Ph.D.
Committee Member

Sandra Safo, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Statistical Learning Methods for Big Biomedical Data

By

Ziyi Li
MPH, Yale University, 2014
BS, Peking University, 2012

Advisor: Qi Long, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Biostatistics
2018

Abstract

Statsitical Learning Methods for Big Biomedical Data

By

Ziyi Li

The rapid advancement of biological and clinical technologies has generated several distinct types of big biomedical data, including -omics data and electronic health record data. Such data and their distinct features have created challenges in obtaining meaningful and applicable research findings. In this dissertation, we develop three statistical learning methods for the analysis of big biomedical data.

Principal component analysis (PCA) is a popular tool for dimensionality reduction, data mining, and visualization of high dimensional data. It has been recognized that complex biological mechanisms occur through concerted relationships of multiple genes working in networks that are often represented by graphs. Recent work has shown that incorporating such biological information improves feature selection and prediction performance in regression analysis, but there has been limited work on extending this approach to PCA. In the **first project**, we propose two new sparse PCA methods called Fused and Grouped sparse PCA that enable incorporation of prior biological information in variable selection, leading to improved feature selection and more interpretable principal component loadings and potentially providing insight on molecular underpinnings of complex diseases. Our simulation studies suggest that, compared to existing sparse PCA methods, the proposed methods achieve higher sensitivity and specificity when the graph structure is correctly specified, and are fairly robust to misspecified graph structures. Application to a glioblastoma gene expression dataset identified pathways that are suggested in the literature to be related with glioblastoma.

Electronic health record (EHR) data provide promising opportunity to explore personalized treatment regime and to make clinical predictions. Compared with genomics data, EHR data are known for their irregularity and complexity. In addition, analyzing EHR data involves privacy issues and sharing such data among multiple research sites may not be feasible due to privacy concerns and regulatory hurdles. Recent work uses contextual embedding models and successfully builds one predictive model for analysis of EHR data from multiple sites for more than seventy common diagnoses. Although the existing model can achieve a relatively high predictive accuracy, it cannot build global models without sharing data among sites. In the **second project**, we propose three novel contextual embedding methods to build predictive models called Naive updates, Dropout updates, and Distributed Noise Contrastive Estimation (Distributed NCE). In addition, we also propose Distributed NCE with DP, which is an updated version of Distributed NCE, to obtain reliable privacy protections. Our simulation study with a real dataset demonstrates that the proposed

methods not only can build predictive model with privacy protection distributedly, but also well preserve the model structure and achieve comparable prediction accuracy compared with hidden-truth model built with all the data.

Biclustering technique can identify local patterns of a data matrix by clustering rows and columns at the same time. Various biclustering methods have been proposed and successfully applied to analyze gene expression data. While existing biclustering methods have many desirable features, most of them are developed for continuous data and none of them can handle genomic data of various types, for example, binomial data as in Single Nucleotide Polymorphism(SNP) data or negative binomial data as in RNA-seq data. In addition, none of existing methods can utilize biological information such as those from functional genomics or proteomics. Recent work has shown that incorporating biological information can improve variable selection and prediction performance in analyses such as linear regression and multivariate analysis. In the **third project**, we propose a novel Bayesian biclustering method that can handle multiple data types including Gaussian, Binomial, Negative binomial, and Poisson data. In addition, our method uses a Bayesian adaptive structured shrinkage prior that enables feature selection guided by biological information such as those from functional genomics. Our simulation studies and application to mutiple genomics datasets demonstrate robust and superior performance of the proposed method, compared to other existing biclustering methods.

For future work, we can continue the direction of the first topic and explore the potential extension of sparse PCA combining neural network, or continue the direction of the second topic and replace Word2Vec with recently proposed embedding approaches, or continue the direction of the third topic to incorporate subject level phenotype information into the biclustering process.

Statistical Learning Methods for Big Biomedical Data

By

Ziyi Li
MPH, Yale University, 2014
BS, Peking University, 2012

Advisor: Qi Long, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Biostatistics
2018

Acknowledgement

I would never have been able to finish my dissertation without the guidance of my committee members, help from friends, and support from my family.

First I would like to thank the Department of Biostatistics and Bioinformatics at Emory University, for choosing me and supporting me as a doctoral student, and providing an excellent atmostphere for doing research.

I would like to express my deepest gratitude to my advisor, Dr. Qi Long, for his excellent guidance, caring and patience. Dr. Long was both my acedemic advisor and dissertation advisor and started to guild me since day one of my Ph.D. study. I would like to thank Dr. Sandra Safo and Dr. Xiaoqian Jiang, with whom I had honor to work in separate dissertation topics and who taught me the essence of research with great detail and patientce. I would also like to thank Dr. Jeanne Kowalski for serving in my dissertation committee and actively providing me feedbacks. Thanks also go to Dr. Hao Wu, who, although not a member of dissertation committee, generously supports me financially through RA and TA positions in my last two years of study.

I would like to thank Changgee Chang, who closely worked with me in my third dissertation topic. He taught me advanced Bayesian techniques and helped me in coding when I struggled in research. Many thanks to Nancy Jin, Harry Feng, Yunxiao Li, and Bo Wei, who as my good friends, were always willing to help and give their best suggestions.

I would also like to thank my parents. They were always supporting me and encouraging me with their best wishes.

Finally, I would like to thank my husband, Louie, and my two lovely babies, Bradley and Hayley. They were always there cheering me up and stood by me through the good times and bad.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview of Big Biomedical Data

Big data is not an exclusive treasure for Biomedical field – it is the product of this era. High-tech companies collect hundreds of millions of data everyday for customer records or transaction information. For instance, Amazon maintains records of more than 59 million customers and their personal information from general backgrounds, transaction history, to any websites records related to potential purchase interests. These collections add up to more than 42 terabytes data according to an online source in 2010 (CBP, 2010). In another instance, AT&T holds a database of huge size. According an online article published in 2006, AT&T contains electronic records of 1.92 trillion telephone calls at that time (Markoff, 2006). Although big datasets exist in all walks of life, the fact that big datasets in biomedical field attracts a large amount of attention may rarely happens for datasets in other fields. These attentions come from researchers, doctors, entrepreneurs, government agency officers, even from the president.

There are many types of big biomedical data. Both "-omics" data and electronic health records data unarguably belong to big biomedical data. The new emerging mobile health data can also be big if data is collected frequently and the dimension of variables is high. Other big biomedical data includes imaging data and some environmental health data. This list of data types is limited to the author's knowledge and is very likely to be incomplete, since more and more big biomedical data is under development with the rapid growth of technology. For the above mentioned data types, we provide detailed descriptions in the following paragraphs.

According to Wikipedia, the term "-omics" unofficially refers to the study fields of Biology such as genomics, proteomics, metabolomics, and etc. (Wikipedia, 2016b). The research of omics data dates back to the efforts of investigating DNA structure

by Watson and Crick in 1953 (Watson and Crick, 1953). With the development of different DNA sequencing technologies, such as Shortgun sequencing (Sanger, 1981), Illumina sequencing (Kawashima et al., 2005), and next-generation sequencing (Hall, 2007), genomics research field grows rapidly and genomics data becomes one of the most popular and most common data used in analysis when genome is related in research. Proteomics focus on the large-scale study of proteins. The data of this field include the information of protein structure and functions. This notion appears in 1997 (James, 1997) and mass spectrometry is the main technology used to collect data in the field of proteomics (Sparkman, 2000). Metabolomics target on the systematic research of metabolites left behind by the chemical process in human body. The idea of building metabolic profile is first proposed by Gates and Sweeley (1978). The data in metabolomics are collected by multiple techniques, for example, RNA can be examined using sequencing technology and protein can be tested using mass spectrometry.

Omics data has a few features. First of all, the data is usually "big". A binary alignment/map file from whole genome sequencing of a single person is about $80-90$ gigabytes according to Koboldt (2014). Thus the data for a 1000 sample size whole genome sequencing can consume 80 terabytes of storage (Koboldt, 2014). The second feature of omics data is that data is "structured", i.e., the data usually has a hidden structure. This is determined by the underlying biological process of human body. For instance, genes are connected and form different pathways. This feature may be utilized when analyzing omics data and benefit the research results. The third feature is that data is sparse and sample size n is much smaller than the variable size p. Because for omics data, dimension of variables is usually very much. According to the data sheet of Illumina Infinium OmniExpress-24 v1.2 BeadChip (Illumina, 2016), one such chip includes 713,599 markers. Thus the one axis of the tested results has dimension more than 710,000 and another axis of the results, which is usually the

sample size, can be as small as a few hundreds. Omics data can have other features such as having special noise, including protected health information, and etc. .

In this dissertation work, we focus on genomics data in the first topic. To be specific, we develop our methods under the framework that variables can be genes and are structured. We analyze a gene expression data for Glioblastoma patients in Section 2.4.

The term electronic health record (EHR) data and electronic medical record (EMR) data are usually used interchangeably. Both of them can refer to the digital-formatted and electronically stored collection of patient health information (Gunter and Terry, 2005). In some cases, EHR may have a broader meaning than EMR in that EMR refers to the standard care information collected in health provider's office while EHR is a system designed to maintain and share information across all the providers (HealthIT, 2014). The US congress encourages the use of EHR system to replace traditional paper-based records through incentives (for using EHR) and penalties (for using paper-based records) announced through the Health Information Technology for Economic and Clinical Health (HITECH) Act (USDHHS, 2006). By the end of 2009, more than 48.3 percent of the health institutions in US have adopted EMR system according to CDC estimates (NCHS, 2009).

EHR data contains a variety of health information such as demographics, prescriptions, diagnoses, lab results, and etc.. Usually these items for single patients are stored in separate modules and can be linked by unique patient IDs. Because EHR data include a quite detailed record of patients' health information, it is usually very large, especially when the number of subjects is large. According to an online talk given by Dr. Hartzband from MIT, typical EHR data ranges from 1MB for a healthy young person, to 40MB for a middle-aged person with health issue to $3-5$ GB for patients who have imaging documents (Hartzband, 2011). And the healthcare group

Kaiser Permanente is estimated to maintain about 26.5 to 44 petabytes (1 PT = 1,000 TB) EHR data (Hartzband, 2011).

Besides the large size, EHR data has some other features. First of all, "inaccuracy". Data mistakes can happen frequently, since there are so many factors that may impact the record of EHR data. These impacting factors include random typos from a tired medical staff or systematic biased recording influenced by billing requirements or natural tendency to avoid liability. Another feature is complexity. EHR data is highly complex because the variables in EHR data are a mix of continuous variables and categorical variables (Hripcsak and Albers, 2013). Word sequences are usually used to record symptoms and miscellaneous. In addition, there is no uniform coding system for almost all the categories except diagnoses, i.e. different hospitals can have completely different term to represent all medical events except diagnoses. This poses great challenges in analyzing several EHR datasets at the same time. Some other features of EHR data are high-dimensionality, sparsity, irregularity and etc.. All these features make analyzing EHR data using traditional methods not feasible or very difficult.

Even if analyzing EHR data is not an easy task, EHR data has attracted increasing attentions from medical researchers recent years. One reason is that EHR data contains a comprehensive profile of each patient and their treatment histories. Analyzing EHR data may be able to reveal the general treatment pattern and thus further help the decision making by doctors and patients. Another reason is the great promotion of precision medicine initiative (Collins and Varmus, 2015) from government agencies and the president. Both EHR data and the mobile health data which we will discuss later are all useful resources to investigate personalized optimal treatment regime.

Besides the data types mentioned above, there are other big biomedical data which are also very important and provide insights into biomedical research. For example,

imaging data can be generated by a variety of modalities including Functional magnetic resonance imaging (fMRI), magnetic resonance imaging (MRI), and Diffusion tensor imaging (DTI). Imaging data help us understand the brain structure and brain activity during rest state or non-rest state (Bowman, 2014). In this dissertation work, we mainly focus on genomic data in the first and third topics and EHR data in the second topic.

1.2 Principal Component Analysis (PCA) : A multivariate analysis method

One difficulty of analyzing genomic data is that they usually have high dimensions which makes traditional statistical methods inappropriate. To solve this, one way is to reduce data dimension using statistical tools and principal component analysis (PCA) is a most popular tool for data dimension reduction. Although PCA is invented by Pearson more than one hundred years ago (Peason, 1901) and is more formally developed by Hotelling more than seventy years ago (Hotelling, 1933), PCA is still widely used now for descriptive analysis and dimension reduction.

Suppose we have a random data matrix $X = (x_1, \ldots, x_p)$ with dimension $n \times p$ and $x_i, i = 1, \ldots, p$ are $n \times 1$ vectors. PCA is defined as an orthogonal linear transformation so that the greatest variance is achieved by the first transformation or first principal component, the second greatest variance is achieved by the second transformation, and so on. Each principal component is orthogonal to the other components. Define a linear combination of $x_i, i = 1, \ldots, p$ is $X\alpha$. The first principal component loading α_1 satisfies

$$\max_{||\alpha_1||=1} \alpha_1^{\mathrm{T}} X^{\mathrm{T}} X \alpha_1 \tag{1.1}$$

And the $r-$th principal component loading α_r satisfies

$$\begin{aligned} \max_{||\alpha_r||=1} \quad & \alpha_r^{\mathrm{T}} X^{\mathrm{T}} X \alpha_r \\ & \text{subject to } \alpha_s^{\mathrm{T}} \alpha_r = 0 \\ & \forall s < r, \quad r = 2, \ldots, q \ll \min(p, n-1). \end{aligned} \tag{1.2}$$

We can use Lagrangian multipliers to solve optimization problem 1.2 as follows:

$$\mathcal{L}(X, \alpha_r, \lambda) = \alpha_r^{\mathrm{T}} X^{\mathrm{T}} X \alpha_r - \lambda \alpha_r^{\mathrm{T}} \alpha_r \tag{1.3}$$

By taking derivatives of $\mathcal{L}$ we obtain the following solution to 1.2:

$$X^{\mathrm{T}} X \alpha_r = \lambda \alpha_r \tag{1.4}$$

Thus the $r-$th principal component loadings for data matrix X is the $r-$th eigenvector corresponding to the $r-th$ eigenvalues, assuming eigenvalues are ranked from largest to smallest. In this dissertation work, we call the α_r the $r-$th PC loading and $X\alpha_r$ the $r-$th PC. One can further show that eigenvalues of a PC loading is equal to the variance explained by this PC. We use this property to calculate proportions of variation explained by the first two PCs in the first topic.

Even if PCA can effectively reduce data dimension and is widely applied in many fields such as signal processing and image compression, PCA has one drawback: PCA finds linear combinations of all variables. This is especially problematic when the number of variables is large because a linear combination of many variables is very hard to interpret. To overcome this drawback, many sparse PCA methods have been proposed and we demonstrate some of the most popular sparse PCA methods in the

following section.

1.2.1 Sparse PCA

The most intuitive and straight-forward approach of obtaining sparse PC solutions is to treat PC loadings elements smaller than some threshold value as zero. However, this approach has been demonstrated as misleading because the importance of variables is not determined by the magnitude of variables (Cadima and Jolliffe, 1995a).

Another straight-forward approach is to impose sparsity constraints on PC loadings. Following this direction, Jolliffe et al. (2003a) proposed a LASSO (least absolute shrinkage and selection operator) (Tibshirani, 1996) based PCA method, which is named *SCoTLASS* (Simplified Component Technique-LASSO). This method imposes LASSO constraints on PC loadings, which sacrifices explained variance to achieve sparsity and improve interpretability. We follow the notation used in 1.2 and *SCoTLASS* finds α_r so that

$$\begin{aligned} \max_{||\alpha_r||=1} \quad & \alpha_r^{\mathrm{T}} X^{\mathrm{T}} X \alpha_r \\ \text{subject to } & \alpha_s^{\mathrm{T}} \alpha_r = 0, \quad \forall s < r, \quad r = 2, \ldots, q \ll \min(p, n-1), \\ & \text{and } \textstyle\sum_{k=1}^{p} |\alpha_{rk}| \leq \gamma \end{aligned} \tag{1.5}$$

where α_{rk} is the $k-$th element of the rth PC loading and γ is some tuning parameter.

Although *SCoTLASS* is easy to understand and has been approved in Jolliffe et al. (2003a) to be effective, there is no efficient algorithm to solve 1.5. According to Zou et al. (2006), the algorithm proposed in Jolliffe et al. (2003a) is expensive and

sometimes make cross-validation for selecting optimal tuning parameter infeasible. Thus Zou et al. (2006) proposes $SPCA$ (Sparse Principal Component Analysis). In $SPCA$, they formulate PCA into a regression-type optimization problem and impose lasso or elastic net (Zou and Hastie, 2005a) constraint on the regression coefficients. Zou et al. (2006) proves that PCA can be transformed into the following regression-type formulation:

$$\begin{aligned}(\hat{A}, \hat{B}) \quad &= \arg\min_{A,B} \sum_{i=1}^{n} ||x_i - AB^{\mathrm{T}} x_i||^2 + \lambda \sum_{j=1}^{r} ||\beta_j||^2 \\ &\text{subject to } A^{\mathrm{T}} A = \mathcal{I}_{r \times r}\end{aligned} \tag{1.6}$$

Where $A_{p\times r} = [\alpha_1, \ldots, \alpha_r]$ and $B_{p\times k} = [\beta_1, \ldots, \beta_r]$. λ is a tuning parameter. β_i is proportional to the $i-$th principal component loadings. Using formalation1.6, $SPCA$ consider the following optimization problem:

$$\begin{aligned}(\hat{A}, \hat{B}) \quad &= \arg\min_{A,B} \sum_{i=1}^{n} ||x_i - AB^{\mathrm{T}} x_i||^2 + \lambda \sum_{j=1}^{r} ||\beta_j||^2 + \sum_{j=1}^{r} \lambda_{1,j} ||\beta_j||_1 \\ &\text{subject to } A^{\mathrm{T}} A = \mathcal{I}_{r \times r}\end{aligned} \tag{1.7}$$

Here λ is the tuning parameter for all r components and $\lambda_{1,j}$ allows for different penalization on different components. Besides theoretical proof of the methods, Zou et al. (2006) also proposes efficient algorithms to solve 1.7 which is included in R package elasticnet.

Besides $SPCA$, another popular sparse principal component method, called SPC, is published by Witten et al. (2009a). Their method is based on a penalized matrix decomposition which is a framework for obtaining a rank$-K$ matrix approximation. More specifically, they use

$$\hat{X} = \sum_{k=1}^{K} d_k \mathbf{u}_k \mathbf{v}_k^{\mathrm{T}} \tag{1.8}$$

to approximate matrix X. Rank K is usually much smaller than the dimension of X. To apply formula 1.8 in sparse PCA, they consider to impose penalty $P_2(\mathbf{v}) = ||\mathbf{v}||_1$ on $\mathbf{v}$ and no constraint on $\mathbf{u}$, and SPC can be written as:

$$\text{maximize}_{\mathbf{u},\mathbf{v}} \mathbf{u}^{\mathrm{T}} \mathbf{X} \mathbf{v} \text{ subject to } ||\mathbf{v}||_1 \leq c_2, ||\mathbf{u}||_2^2 \leq 1, ||\mathbf{v}||_2^2 \leq 1. \tag{1.9}$$

In each updating step, SPC uses the simple update $\frac{\mathbf{X}\mathbf{v}}{||\mathbf{X}\mathbf{v}||_2}$ for $\mathbf{u}$. It has been demonstrated in Witten et al. (2009a) that SPC can achieve a higher proportion of variance explained for given number of sparse components used compared with $SPCA$. In addition, an efficient algorithm to solve SPC is implemented in an R package PMA.

1.2.2 Sparse PCA with structural information

Besides the preference for sparse PC solutions, another data feature has brought to the attention of researchers: data is structured in many situations. How to obtain PC solutions with data structure taken into consideration? To the best of our knowledge, two sparse PCA methods for structured data have been proposed recently and we briefly review it here.

Structured Sparse Principal Component Analysis (SSPCA) is proposed by Jenatton et al. (2010). SSPCA not only imposes sparsity constraint on PC loadings but also consider to include a priori structural constraints during PC computation. Their method is inspired by the situation in computer vision where variability of images is closely related with grid regularity. Following the notation used in Jenatton et al. (2010), SSPCA can be formulated as:

$$\min_{\substack{U,V,\Omega_u(U^k)\leq 1^{r\times|\mathcal{G}|_+}\\(\eta^G)_{G\in\mathcal{G}}\in\mathcal{R}}} \frac{1}{2np}||X-UV^{\mathrm{T}}||_F^2+\frac{\lambda}{2}\sum_{k=1}^{r}\left[(V^k)^{\mathrm{T}}\mathrm{Diag}(\zeta^k)^{-1}V^k+||(\eta_k^G)_{G\in\mathcal{G}}||_\beta\right] \quad (1.10)$$

In this formulation, $X \in \mathcal{R}^{n\times p}$ is the centered data with n rows corresponding to n observations with p dimensions. They call $V \in \mathcal{R}^{p\times r}$ the dictionary, so that a linear combination of the r columns of V (dictionary elements) can approximate each observation. $U \in \mathcal{R}^{n\times r}$ is the matrix of the linear combination coefficients or decomposition coefficients. In our understanding, U is the PC loadings and V is the components referred before. $||\cdot||_F$ is the Frobenius norm. $\zeta \in \mathcal{R}^{p\times r}$ is defined by $\zeta_{jk} = \sum_{G\in\mathcal{G},j\in G}(d_j^G)^2(\eta_k^G)^{-1^{-1}}$, and $(d^G)_{G\in\mathcal{G}} \in \mathcal{R}^{p\times|\mathcal{G}|}$ is a $|\mathcal{G}|$-tuple of $p-$dimensional vectors such that $d_j^G > 0$ if $j \in G$ and $d_j^G = 0$ otherwise.

Besides an efficient algorithm to solve formula1.10, Jenatton et al. (2010) also implement it in a Matlab toolbox. Their notation system and formulation might be a little over-complicated for the problem. Also it is worth-noting that they only use the group membership information of variables but do not consider the connections between variables.

Another promising methods that deal with structured data is proposed by Allen et al. (2014). Their method is called generalized least-square matrix decomposition (GMD) and targets on a completely different type of data structure : structure of noise. Their method is inspired by the analysis of fMRI data. It has been noticed that PCA is rarely used to analyze fMRI data because the first several PCs usually capture spatial and temporal dependencies in noise rather than brain activation patterns. This is decided by the feature of fMRI data which contains noise with strong correlation between neighboring voxels in three-dimensional image.

Following the notation used in Allen et al. (2014), they propose to broaden the

low-rank mean model (Anderson, 1962) to incorporate two-way dependencies in the noise using the following GMD mean model:

$$X = \sum_{k=1}^{K} d_k \mathbf{u}_k \mathbf{v}_k^{\mathrm{T}} + \mathbf{E}; \quad \mathbf{E} \sim (0, \boldsymbol{\Delta} \otimes \boldsymbol{\Sigma}), \tag{1.11}$$
$$\text{such that } \mathbf{U}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{U} = \mathbf{I} \text{ and } \mathbf{V}^{\mathrm{T}}\boldsymbol{\Delta}\mathbf{V} = \mathbf{I}.$$

The n by p matrix X is the centered data matrix. $\mathbf{u}_k \in \mathcal{R}^n$ and $\mathbf{v}_k \in \mathcal{R}^p$ are fixed signals of interest. GMD model assume that the additive noise $\mathbf{E}$ comes from a matrix-variate model with row covariance $\boldsymbol{\Sigma} \in \mathcal{R}^{n \times n}$ and column covariance $\boldsymbol{\Delta} \in \mathcal{R}^{p \times p}$.

Based on model (1.11), a GMP optimization problem is defined as

$$\text{minimize}_{\mathbf{U},\mathbf{D},\mathbf{V}} ||\mathbf{X} - \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}||^2_{\mathbf{Q},\mathbf{R}}, \tag{1.12}$$
$$\text{subject to } \mathbf{U}^{\mathrm{T}}\mathbf{Q}\mathbf{U} = \mathbf{I}_{(K)}, \mathbf{V}^{\mathrm{T}}\mathbf{R}\mathbf{V} = \mathbf{I}_{(K)} \text{ and } \mathrm{diag}(\mathbf{D}) \geq 0,$$

where the $\mathbf{Q}, \mathbf{R}-$norm is defined as $||\mathbf{X}||_{\mathbf{Q},\mathbf{R}} = \sqrt{tr(\mathbf{Q}\mathbf{X}\mathbf{R}\mathbf{X}^{\mathrm{T}})}$.

To solve a PCA problem using GMD model (1.11) and GMP optimization formulation (1.12), they maximize the sample variance in the space that accounts for the noise structure of the data. And a GPCA problem can be defined as:

$$\text{maximize}_{\mathbf{v}_k} \ \mathbf{v}_k^{\mathrm{T}}\mathbf{R}\mathbf{X}^{\mathrm{T}}\mathbf{Q}\mathbf{X}\mathbf{R}\mathbf{v}_k \tag{1.13}$$
$$\text{subject to } \mathbf{v}_k^{\mathrm{T}}\mathbf{R}\mathbf{v}_k = 1 \ \text{ and } \ \mathbf{v}_k^{\mathrm{T}}\mathbf{R}\mathbf{v}_{k'} = 0 \ \ \forall \ \ k' < k \tag{1.14}$$

By solving the problem (1.13), the$r-$th generalized principal component can be obtained by $\mathbf{X}\mathbf{R}\mathbf{v}_r$ and the proportion of variance explained by the $r-$th generalized principal component is calculated by $d_k^2/||\mathbf{X}||^2_{\mathbf{Q},\mathbf{R}}$.

Allen et al. (2014) has demonstrated that GPCA can effectively eliminate the

impact of noise with known structure or dependencies on PC solutions through a simulation study and an application to fMRI data.

1.3 Predictive Model Construction using EHR data

Although multivariate analysis methods such as PCA have been developed into different versions and have good performance in processing genomic data, they cannot effectively analyze EHR data. The reason is very simple: EHR data is too complex to be analyzed with multivariate analysis methods. As mentioned in section(1.1), EHR combines the information from multiple aspects and the data from each aspect can have a different structure, thus it is far more irregular than genomics data. In addition, EHR data frequently contain text-format records, such as symptom description or doctors' notes, which cannot be analyzed by most of the traditional statistical methods.

To effectively and efficiently analysis EHR data, a few methods have been proposed. These methods can be categorized into two classes: deep-learning (DL) based methods (Cheng et al., 2016+; Farhan et al., 2016+) and non-deep-learning (NDL) methods(Batal et al., 2012; Hripcsak and Albers, 2013; Jensen et al., 2012; Wu et al., 2010; Liu et al., 2015). In the current literatures, phenotyping extraction from EHR data is usually the first step and prediction models are built based on the extracted medical features (Cheng et al., 2016+; Jensen et al., 2012; Wu et al., 2010; Liu et al., 2015; Farhan et al., 2016+). Phenotyping extraction can also be used for other analysis such as cost-effective study and general treatment pattern learning (Batal et al., 2012). In the following sections, we first discuss some non-deep-learning methods of analyzing EHR data; then we present some basic background about deep learning methods and a specific deep learning tool for natural language processing (NLP); lastly we present a deep learning based model for predictive model using EHR data.

1.3.1 Analyzing EHR Data with NDL Methods

When new problems first appear, the most natural reaction from researchers is to try existing methods and examine if they work on new problems. EHR data is not a new problem, but in the most recent several years, it gradually attracts much more attention from researchers than before. One reason is that the development of technology enables researchers to analyze the whole EHR data from one or multiple data sites. Another reason is that the rich information contained in EHR may provide great support to detect existing treatment patterns and to develop personalized treatment regime. In order to analyze EHR data, a few NDL methods have been proposed. We briefly explain what techniques those study utilize and how the proposed methods perform.

The work by Wu et al. (2010) reviews the challenges and strategies of analyzing EHR data. They examine three common machine learning techniques: logistic regression with BIC as selection criterion, support vector machine (SVM), and boosting. The EHR records for a total of 536 subjects with heart failure are extracted and used as the experiment dataset.

SVM is a classification technique, which transform the original data space into a higher dimensional space. To reason to do such transformation is that some classification decision making can be easier in higher-dimensional space. Assume the data is consisted of $(x_i, y_i),\ i = 1, \ldots, N$ for N patients and each x_i is assumed to have p inputs. By defining the M feature function as $h(x_i) = (h_1(x_i), h_2(x_i), \ldots, h_m(x_i))$, the decision boundary can be found by

$$\min_{\beta_0, \beta} \sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \lambda ||\beta||^2, \tag{1.15}$$

where $+$ represents the positive part of the function $[1 - y_i f(x_i)]$. The $f(\cdot)$ in

function(1.15) is defined as

$$f(x) = h(x)^{\mathrm{T}}\beta + \beta_0 = \sum_{i=1}^{N} a_i y_i < h(x), h(x_i) > +\beta_0,\ i = 1, \ldots, N. \tag{1.16}$$

The SVM classification results is decided by $\hat{y}_i = sign(h(x_i)^{\mathrm{T}}\hat{\beta} + \hat{\beta}_0)$. Review current temporal phenotyping extraction models

Boosting is a popular machine learning methods which ensembles several classifiers and produce a stronger classifier. Let $G_m, m = 1, \ldots, M$ denote a sequence of weak classifiers. The additive model formulation of AdaBoost is to minimize the exponential loss function:

$$(\beta_m, G_m) = arg \min_{\beta_0, \beta} \sum_{i=1}^{N} [w_i^m e^{(-\beta y_i G(x_i))}], \tag{1.17}$$

And the final classifier is defined as

$$\hat{y}_i = sign\big(\sum_{m=1}^{M} \beta_m G_m(X)\big). \tag{1.18}$$

Wu et al. (2010) demonstrated that logistic regression Boosting acheives the best AUC of about 0.76 for predicting heart failure. SVM has the poorest performance.

Besides traditional machine learning methods, graph based framework has also been used to analyze EHR data. Liu et al. (2015) uses a graph based method to extract temporal phenotyping from longitudinal EHR data. They represent patient EHRs as temporal graphs and the interested phenotypes are detected in the form of subgraphs instead of subsequences, thus it is more flexible to use graph representation than traditional sequence representation. The technique detail is not presented here but the basic idea is to construct a directed and weighted graph to represent the medical events in the data. They impose similarity based and model based regularization on the existing graph to further improve the performance. They examine the proposed

graph model on a dataset including CHF patients and the proposed method can achieve an AUC of about 0.72.

1.3.2 Deep Learning Methods

Deep learning is a broad branch of machine learning. The basic structure used in deep learning is deep graph with multiple processing layers, including deep neural networks, convolutional deep neural networks, deep belief networks and recurrent neural networks. Deep learning works especially well when there is a huge amount of training data, for example, the deep learning based program AlphaGo uses 30 million moves from games played by human experts to train the model and it becomes the first Computer Go program that can beat professional human player in March, 2016 (Silver et al., 2016; Silver and Hassabis, 2016). Deep learning is such a broad field that books have been published and comprehensive tutorials are also available online (MacKay, 2003; Weigel, 2002; LISA−lab, 2016; Gulcehre, 2016). It is also a cutting-edge field under rapid development. Because of the space limit, we only briefly describe the basic structure of neural network and how to solve it using back-propagation here.

Neural networks or artificial neural networks are the basic element of deep learning methods. As discussed in MacKay (2003), a neural network can be determined by three factors:

- Architecture, which specifies the topology structure of neurons, for example, how neurons are connected.
- Activation function, which decides how each neuron reacts to input signals. A few functions are usually selected as activation function:
    1. Linear function: $y(x) = x$;
    2. Sigmoid function (or logistic function): $y(x) = \frac{1}{1+e^x}$;
    3. Tanh function: $y(x) = tanh(x)$;

4. Thredshold function:

$$y(x) = \begin{cases} 1, & a > 0. \\ 0, & a \leq 0. \end{cases}$$

- Learning rule, which specifies how the weights of neural network change with time. Usually learning rule is decided by activation function and loss function together.

After a neural network is specified, backpropagation is the most commonly used method to solve a neural network. Backpropagation, which is abbreviated for "backward propagation of errors", calculates the gradients of loss function with regards to all the weights and optimize the selection of weight to decrease total loss. When neural networks include multiple layers, gradients of loss function is calculated following chain rule layer by layer backwardly and the gradients for weights in front layers are the multiplication of several gradients for elements in back layers (Hecht-Nielsen, 1989).

Deep learning has applications in a wide range of fields including system identification and control (Zissis et al., 2015), visual identification and classification (Simard et al., 2003), quantum chemistry (Balabin and Lomakina, 2009), game-playing and decision making (Stanley et al., 2005), and etc.. It is worth mentioning that neural network not only can take numerical measurements as input and output, but also can accept text sequence as data. For example, Word2Vec is a three-layers neural network based structure and learns contextual vector representations for words (Mikolov, Chen, Corrado and Dean, 2013). The accommodation of both words and numerical inputs makes neural network a promising tool for analyzing EHR data.

1.3.3 Analyzing EHR data using DL Methods

To the best of our knowledge, although deep learning methods have been widely applied in many fields, it is in recent years that researchers start to analyze EHR

data using DL methods. We review two related works here.

The work by Cheng et al. (2016+) directly adopt the regular convolutional neural network model. To accommodate the temporal features of EHR data, their model consists of four layers - a input layer with the temporal matrix of EHR data, a convolutional layer with different window sizes, a pooling layer with additional normalization function, and a fully connected layer using softmax function. In addition, they proposed different fusing structure to fully utilize the temporal pattern in EHR data. Figure 1.1 demonstrates the four-layered neural network proposed in this paper. They examine the methods using a Congestive Heart Failure (CHD) cohort (1127 cases and 3850 controls) and a Chronic Obstructive Pulmonary Disease (COPD) cohort (477 cases and 2385 controls). And the proposed models can achieve a best AUC of around 0.77 for CHF and 0.74 for COPD.

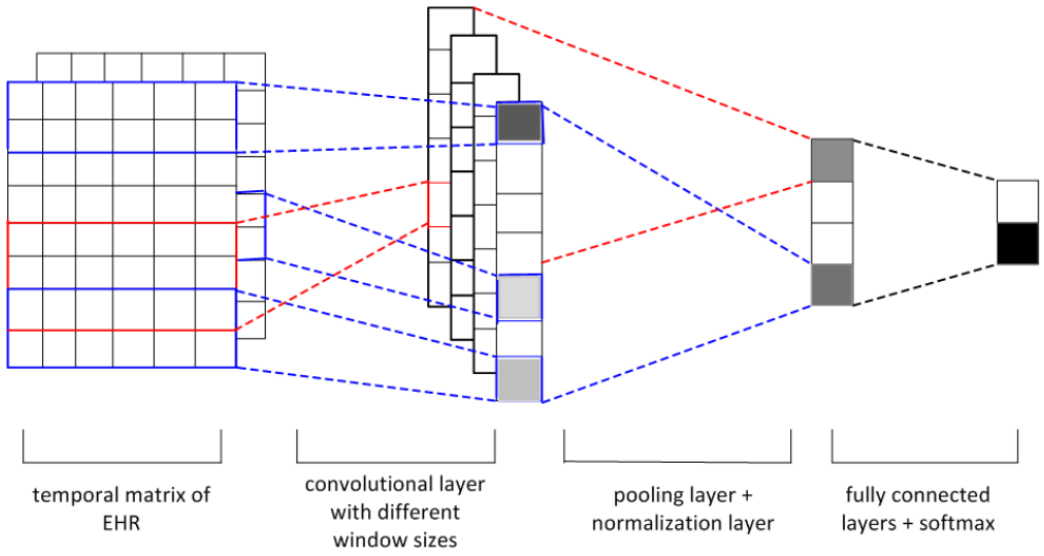


Figure 1.1: The basic model architecture of the model proposed by Cheng et al. (2016+)

Different from the technique utilized by Cheng et al. (2016+), the work by Farhan et al. (2016+) first obtain the contextual vector representations of all medical events by training a Word2Vec model (Mikolov, Chen, Corrado and Dean, 2013). Then they build a prediction model called Patient-Diagnosis Projection Similarity (PDPS),

which projects patients' medical event sequence into vector space and find the most similar diagnoses as prediction for this patients. Assume one patient has a sequence of medical events S, the PDPS prediction for this patient is decided by

$$d^* = \arg\max_{d \in D} \mathrm{CS}(V_d, \frac{\sum_{e \in S} V_e exp(-\lambda t_e)}{\sum_{e \in S} exp(-\lambda t_e)}) \tag{1.19}$$

Here CS is cosine similarity. V_e is the vector representation for event e. V_d is the vector representation for diagnosis d. λ is the decay factor and usually takes a value between 0 and 1. Figure 1.2 demonstrates how to make prediction for a patient with sequence $(e_1, e_2, \ldots, e_N)$. PDPS finds the concatenated vector E computed with event sequence and temporal decay factor. Then it is obvious that Heart Failure is the most similar vector for diagnoses to E. Farhan et al. (2016+) test their method

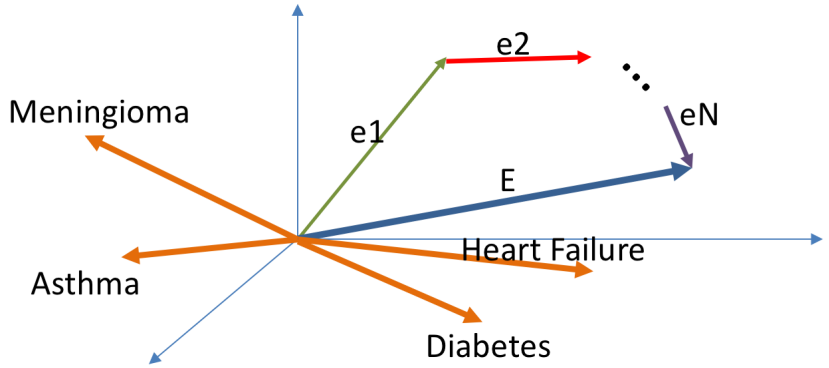


Figure 1.2: The PDPS model proposed by Farhan et al. (2016+)

on a real EHR dataset and achieve an average AUC of 0.76 for over 79 most common diagnosis. PDPS performs especially well for a few common diseases, for example, it acheives an AUC of 0.84 for heart failure and 0.88 for hypertensive chronic kidney disease.

1.4 Biclustering

Biclustering, also called block clustering, co-clustering, or two-model clusting, is a data mining technique which can cluster rows and columns of a data matrix simultaneously. The first proposal of biclustering goes back to 1970s under the term "direct clustering" Hartigan (1972). It was first applied to gene expression data by Cheng and Church in 2000 Cheng and Church (2000). Since then, more than 10 biclustering approaches and several review papers are published Cheng and Church (2000); Hochreiter et al. (2010); Lazzeroni and Owen (2002); Sheng et al. (2003); Ben-Dor et al. (2003); Gu and Liu (2008); Caldas and Kaski (2008); Bergmann et al. (2003); Murali and Kasif (2002); Yu et al. (2017); Liu et al. (2014); Prelić et al. (2006); Pontes et al. (2015); Eren et al. (2012); Padilha and Campello (2017).

According to the recent review paper Padilha and Campello (2017), these algorithms can be categorized into four groups based on their type of heuristic they use: greedy, divide-and-conquer, exhaustive enumeration and distribution parameter identification. As described, a greedy algorithm finds the local optimal sub-patterns at each iterations with the hope of locating the global maximal after a few iterations. Greedy algorithm is a big category of biclsutering methods, including Cheng and Church's Algorithm (CC), Conserved Gene Expression Motifs (xMotifs), Iterative Signature Algorithm (ISA), and many other methods Cheng and Church (2000); Murali and Kasif (2002); Bergmann et al. (2003). Divide-and-conquer algorithm divides the input data matrix into smaller sub-matrices, identifies patterns in each smaller instance, and then combines the partial soluation to a global one. Divide-and conquer algorithms include the Binary Inclusion-Maximal Biclsutering Algorithm (Bimax). Prelić et al. (2006). Exhaustive enumeration algorithms conduct exhaustive search by generating all possible row and column combinations. An example of exhaustive enumeration algorithm is Statistical-Algorithmic Method for Biclsuter Analysis

(SAMBA) Tanay et al. (2002). Distribution parameter identification algorithms assume an underlying structure of bicluster and find associated parameters through optimizing some objective functions. Plaid and Factor Analysis for Bicluster Acquisition (FABIA) are representations for this type of algorithms Caldas and Kaski (2008); Hochreiter et al. (2010).

To better understand the motives and structures of the existing biclustering methods, we use three methods from greedy algorithm group and two methods from exhaustive enumeration algorithm group as examples, and present these methods with more details. These methods are also used as comparatives in Chapter Four.

1.4.1 Greedy algorithms: CC, xMotifs, and ISA

Cheng and Church's Biclustering Algorithm (**CC**) is the first work that use the term "bicluster" and apply it to gene expression datasets Cheng and Church (2000). In this work, given a data matrix (a_{ij}), they define the notion "mean squared residue" of element a_{ij} in the bicluster indicated by the subsets I and J as

$$a_{ij} - a_{i\cdot} - a_{\cdot j} + a_{\cdot\cdot}$$

where $a_{i\cdot}$ is the mean of the i-th row in the bicluster, $a_{\cdot j}$ is the mean of the j-th column in the bicluster, and $a_{\cdot\cdot}$ is the mean of all elements in the bicluster. CC finds large and maximal biclusters with mean squared residue scores below a certain threshold. With this goal in mind, Cheng and Church define the mean squared residue score of a submatrix A_{IJ} by

$$H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$$

where

$$a_{iJ} = \frac{1}{|J|}\sum_{j\in J} a_{ij}, a_{Ij} = \frac{1}{|I|}\sum_{i\in I} a_{ij},$$

and

$$a_{IJ} = \frac{1}{|I||J|}\sum_{i\in I, j\in J} a_{ij}.$$

A submatrix A_{IJ} is called a $\delta-$bicluster if $H(I,J) \leq \delta$ for some $\delta \geq 0$. And their paper proved that the problem of finding the largest square $\delta-$bicluster is $NP-$hard. A number of algorithms were proposed in the paper to efficiently conduct brute-force deletion and addition, single node deletion, multiple nodes deletion and node addition. Applications on the yeast data and the human B-cells expression data have shown the proposed methods can identify meaningful biclusters of genes and conditions.

Conserved gene expression motifs or **xMotifs** is a representation method for gene expression data which identify simultaneous subsets of genes and samples. Given the gene expression measurements for a set of genes over a group of samples and two user-defined parameter $0 < \alpha, \beta < 1$, a conserved gene expression motif is defined as a pair (C, G) where C is a subset of samples and G is a subset of genes. The (C, G) pair satisfies three conditions: first, the number of samples in C is at least an α-fraction of all the samples; second, every gene in G is conserved across all the samples in C, i.e., the gene is in the same state in all the samples in C; third, for every gene not in G, the gene is conserved in at most a β-fraction of the samples in C. They proposed an algorithm that iterates across the random subsets of appropriate sizes and checks whether the selected subset meet the requirements. Although the algorithm does not utilize any fancy technique, the implementation in $C++$ makes their program more efficient. They applied xMotifs to an acute lymphoblastic leukaemia (ALL)/acute myeloid leukaemia (AML) dataset, a colon cancer dataset and a B-cell lymphoma dataset, and found meaningful biclusters which corresponds to patients with different diseases.

The Iterative Signature Algorithm (**ISA**) finds transcription modules from genome-wide expression data and their definition of transcription modules (TM) corresponds to the previous definition of biclusters. A TM contains both a set of genes and a set of experimental conditions. Given a gene expression matrix $\mathbf{E}$ represented by row vectors or column vectors

$$\mathbf{E} = \begin{bmatrix} g_1^T \\ g_2^T \\ \vdots \\ g_{N_c}^T \end{bmatrix}$$

$$or \qquad \mathbf{E} = (c_1, c_2, \cdots, c_{N_g}),$$

ISA first finds the two normalized expression matrices

$$\mathbf{E_G} = \begin{bmatrix} \hat{g}_1^T \\ \hat{g}_2^T \\ \vdots \\ \hat{g}_{N_c}^T \end{bmatrix}$$

$$or \qquad \mathbf{E_C} = (\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_{N_g}).$$

Here the rows of $\mathbf{E_G}$ and the columns of $\mathbf{E_C}$ are defined as normalized gene and condition vectors:

$$\hat{g}_c = \frac{g_c - \langle g_c \rangle_{g \in G}}{|g_c - \langle g_c \rangle_{g \in G}|}, \quad \hat{c}_g = \frac{c_g - \langle c_g \rangle_{c \in C}}{|c_g - \langle c_g \rangle_{c \in C}|}.$$

A TM is a combined set of co-regulated genes G_m and relevant experimental condi-

tions C_m, $M_m = G_m, C_m$ satifying the following conditions:

$$\exists(T_c, T_g) : \begin{cases} C_m(G_m) = c \in C : \langle E_G^{cg} \rangle_{g \in G_m} > T_c, \\ G_m(C_m) = g \in G : \langle E_C^{cg} \rangle_{c \in C_m} > T_g, \end{cases}$$

where T_c and T_g are two threshold parameters. Bergmann et al. (2003) proved that ISA is a generalization of Singular Value Decomposition, which corresponds to the case of no threshold parameters. They have applied ISA to a yeast expression data and identified biologically meaningful co-regulated unites.

1.4.2 Distribution parameter identification algorithms: Plaid and FABIA

Plaid is one of the earliest proposed biclustering models. Plaid was based on a straight-forward model which tries to describe the input data through multiple layers. Each layer capture a pattern that genes and conditions are clustered together. Assume the data matrix is (Y_{ij}), plaid uses the following model to fit data

$$\begin{aligned} Y_{ij} &\doteq \mu_0 + \sum_{k=1}^{K} (\mu_k + \alpha_{ik}) \rho_{ik} \kappa_{jk}, \\ Y_{ij} &\doteq \mu_0 + \sum_{k=1}^{K} (\mu_k + \beta_{jk}) \rho_{ik} \kappa_{jk}, \\ Y_{ij} &\doteq \mu_0 + \sum_{k=1}^{K} (\mu_k + \alpha_{ik} + \beta_{jk}) \rho_{ik} \kappa_{jk}. \end{aligned}$$

Here μ_0 models the background signals for all layers, μ_k models the background signals in layer k, α_{ik} and β_{jk} models the signals for gene i and sample j in layer k. The name "plaid" is used to describe the color of plotting μ_k, α_{ik} and β_{jk}. If one uses notation θ_{ijk} to represent $\mu_k, \mu_k + \alpha_{ik}, \mu_k + \beta_{jk}, \mu_k + \alpha_{ik} + \beta_{jk}$, the whole model can

be writted as the following form:

$$Y_{ij} \doteq \sum_{k=0}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk}.$$

To solve plaid model, one seek to find parameter fits so that the following Q reaches a small value:

$$Q = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}(Y_{ij} - \theta_{ij0} - \sum_{k=1}^{K}\theta_{ijk}\rho_{ij}\kappa_{jk})^2.$$

The authors of plaid model used a simple rule to determine the best number of layers. They proposed to add layers to existing model until the importance of layer k denoted by the sum of squares $\sigma_k^2 = \sum_{i=1}^{n}\sum_{j=1}^{p}\rho_{ik}\kappa_{jk}\theta_{ijk}^2$ is not significant any more. The significance of the sum of squares is assessed by a permutation test. Although plaid uses a very straight-forward formulation, applications on the real datasets have shown good performance of the method.

FABIA represents "factor analysis for biclsuter acquisition" and was proposed by Hochreiter et al. in 2010. FABIA is also the method that motivates the third topic of this dissertation. Their paper Hochreiter et al. (2010) has given a formal definition of a bicluster in a transcriptomic dataset, which is "a pair of a gene set and a sample set for which the genes are similar to each other on the samples and vice versa". The overall model is an outer product $\boldsymbol{\lambda}\boldsymbol{z}^{\boldsymbol{T}}$ of two prototype vectors $\boldsymbol{\lambda}$ and $\boldsymbol{z}$ plus a background noise matrix $\boldsymbol{\Gamma}$:

$$\boldsymbol{X} = \sum_{i=1}^{p}\boldsymbol{\lambda}\boldsymbol{z}^{\boldsymbol{T}} + \boldsymbol{\Gamma}$$

where p is the number of biclusters. The vector $\boldsymbol{\lambda}$ represents a column vector which element vlaues are zero if the genes are not contained in the bicluster, and similary $\boldsymbol{z}$ is a vector denoting the participation of samples with elements equal zero if the samples are not contained in the bicluster. To obtain sparse solutions of $\boldsymbol{\lambda}$ and $\boldsymbol{z}$, the

authors impose a Laplace prior on the factor $\boldsymbol{z}$:

$$p(\boldsymbol{z}) = (\frac{1}{\sqrt{2}})^p \prod_{i=1}^{p} e^{-\sqrt{2}|z_i|}$$

and again a Laplace prior on the factor $\boldsymbol{\lambda}$

$$p(\boldsymbol{\lambda}) = (\frac{1}{\sqrt{2}})^n \prod_{k=1}^{n} e^{-\sqrt{2}|\lambda_k|}$$

Hochereiter et al. used variational EM algorithm to optimize the likelihood and a thresholding approach to extract bicluster members from underlying parameters. They applied the proposed FABIA method and other existing methods on three gene expression datasets. FABIA has been shown to have the satisfactory performance compared to existing methods and have found biologically meaningful biclusters.

1.5 Motivation Examples

The first topic of this dissertation work is motivated by the gene expression dataset of Glioblastoma patients from TCGA project. We hope to extract biologically-meaningful principal components which can help to identify the subtypes of patients. As variables in the data correspond to genes and it is well acknowledged that genes form biological pathways, we would love to take advantage of the pathway information as the prior knowledge of how genes are structured. In addition, we hope to obtain principal components with sparse loadings.

The second topic deals with the hypothesized question during analyzing the MIMIC III database. MIMIC III database collects the EHR data of patients from ICU department of Beth Israel Health Center. The hypothesized questions is that how to build a global predictive with all the data if only part of the data is publicly available while the rest of the data have to be analyzed on-site. This hypothesized scenario appears

frequently in real life and can be generalized to the case when multiple databases are involved. Usually researchers from different hospitals would like to cooperate and build global model with all the data, but they are not willing to share the data for privacy or research reasons. How to build predictive model with all the data distributedly is also equivalent to the hypothesized question we raise above.

The third topic is partly motivated by the NCI 60 cell line datasets (https://discover.nci.nih.gov/cellminer/loadDownload.do) and partly motivated by the work of polya-gamma distribution. The NCI 60 series of datasets contais genomic datasets of multiple types for 60 tumor cell lines, for example, gene expression data which is of continuous observations and RNA-seq data which is of count data. Existing biclustering methods usually can only utilize only datasets of one data type. By joining the work of polya-gamma distribution and biclustering technique, we realize that it is possible to conduct analysis using data with multiple data types. In addition, we hope to explore the possibility of incorporating biological information into the analysis.

1.6 Outlines

In this dissertation, we present some statistical methods for analyzing big biomedical data. In chapter 2, we propose two novel sparse principal component analysis methods which can incorporate biological information - Fused sPCA and Grouped sPCA. We further apply the proposed methods on a Glioblastoma gene expression dataset. In chapter 3, we present three solutions of constructing predictive model by analyzing multiple EHR data distributedly. We also examine our methods on a real world EHR dataset of ICU patients. In chapter 4, we propose a biclustering framework which can utilize genomic datasets of multiple types and incorporate prior structural information in the analysis at the same time. We conduct simulation analysis and a series of real data applications to evaluate the performance. We discuss future work in chapter 5.

Chapter 2

Incorporating Biological Information in Sparse Principal Component Analysis with Application to Genomic Data

2.1 Introduction

A central problem in high-dimensional genomic research is to identify a subset of genes and pathways that can help explain the total variation in high-dimensional genomic data with as little loss of information as possible. Principal component analysis (PCA) (Hotelling, 1936) is a popular multivariate analysis method which seeks to concentrate the total information in data with a few linear combinations of the available data, making it an appropriate tool for dimensionality reduction, data analysis, and visualization in genomic research. Despite its popularity, the traditional PCA is often difficult to interpret as the principal component loadings are linear combinations of all available variables, the number of which can be very large for genomic data. It is therefore desirable to obtain interpretable principal components that use a subset of the available data. To deal with the problem of interpretability of principal component loadings,

Several alternatives to PCA have been proposed in the literature, most of which constrain the size of non-zero principal component loadings. An ad hoc approach sets the absolute value of loadings that are smaller than a threshold to zero. Though simple to understand, this approach has been shown to be misleading in the sense that magnitude of loadings is not the only factor to determine the importance of variables in a linear combination (Cadima and Jolliffe, 1995b). Truncating PCs by loadings may result in quite different PCs explaining much smaller variation compared with the original PCs. Other approaches regularize the loadings to ensure that some are exactly zero, which implies that the corresponding variables are unimportant in explaining the total variation in the data. For instance, Jolliffe et al. (2003b) proposed the SCotLass method that constrains the loadings with a lasso penalty, but their optimization problem is nonconvex, which is difficult to solve and does not guarantee convergence to a global solution. Zou et al. (2006) proposed a convex sparse PCA

method (SPCA) that reformulates the PCA problem as a regression problem and imposes elastic net penalty on the PC loadings. Witten and Tibshirani (2009) also proposed the penalized matrix decomposition (PMD) that approximates the data with its spectral decomposition and imposes a lasso penalty on the right singular vectors, i.e., the principal component loadings.

Although the aforementioned methods can effectively produce sparse principal component coefficients, their main limitation is that they are purely data driven and do not exploit available biological information such as gene networks. It has been recognized that complex biological mechanisms occur through concerted relationships of multiple genes working together in pathways. Recent work(Li and Li, 2008a; Pan et al., 2010b) has demonstrated in the regression setting that utilizing prior biological information among variables can improve variable selection and predication and help gain a better understanding of analysis results. It is therefore desirable to conduct PCA with incorporation of known structural information. Allen et al. (2014) proposed a generalized least-square matrix decomposition framework for PCA that incorporates known structure of noise and generate sparse solutions. Although this method can flexibly account for noise structure in data, they do not utilize prior biological information, and do not consider the relationships among the signal variables in PCA. Jenatton et al. (2010) proposed a structured sparse PCA method that considers correlations among groups of variables and imposes a penalty similar to group lasso on the principal component loadings, but their method does not take into account the complex interactions among variables within a group. In this article, we proposed two new sparse PCA methods called Fused and Grouped sparse PCA that enable incorporation of prior biological information in PCA. The methods will allow for identification of genes an pathways. We utilize the L_γ norm(Pan et al., 2010b) and generalize fussed lasso (Tibshirani et al., 2005) to achieve automatic variable selection and simultaneously account for complex relationships within pathways.

Our work makes several contributions. To the best of our knowledge, this is the first attempt to impose both sparsity and smoothing penalties on principal component loadings to encourage the selection of variables that are connected in a network. Although Jenatton et al. (2010) incorporated group information of variables when generating sparse PC solutions, they did not consider how variables are connected in each group. Our method considers not only the group information, but also any interaction structure of variables within a group. By utilizing the existing biological structure in the data, we are able to obtain sparse principal components that are more interpretable and may shed light on the underlying complex mechanisms in the data. We also develop an efficient algorithm that can handle high dimensional problems. Simulation studies suggest that the methods have higher sensitivity and specificity, and are quite effective in improving the performance of sparse PCA methods when the graph structure is correctly specified. In addition, the proposed methods are robust to misspecified graph structure.

The remainder of the paper is organized as follows. In section 2.2, we present methods and algorithms for the proposed sparse PCA. In Section 2.3, we conduct simulation studies to assess the performance of our methods in comparison with several existing sparse PCA methods. In Section 2.4, we apply the proposed methods to data from a glioblastoma brain multiform study. We conclude with some discussion remarks in Section 2.5.

2.2 Methods

Suppose that we have a random $n \times p$ matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p), \mathbf{x} \in \Re^n$. We also assume that the predictors are centered to have column means zero. The network informaiton for the p variables in $\mathbf{X}$ is represented by a weighted undirected graph $\mathcal{G} = (C, E, W)$, where C is the set of nodes corresponding to the p features, $E = \{i \sim j\}$ is the set of edges indicating that features i and j are associated in a biologically

meaningful way, and W includes the weight of each node. For node i, denote by d_i its degree, i.e., the number of nodes that are directly connected to node i and by $w_i = f(d_i)$ its weight which can depend on d_i. Our goal is to obtain sparse PCA loadings while utilizing available structural information $\mathcal{G}$ in PCA. Our approach to the sparse PCA problem relies on the eigenvalue formulation of PCA, and for completeness sake, we briefly review the classical and sparse PCA problems.

2.2.1 Standard and Sparse Principal Component Analysis

Classical PCA finds projections $\boldsymbol{\alpha} \in \Re^p$ such that the variance of the standardized linear combination $\mathbf{X}\boldsymbol{\alpha}$ is maximized. Mathematically, the first principal component loading $\boldsymbol{\alpha}$ solves the optimization problem

$$\max_{\boldsymbol{\alpha} \neq \mathbf{0}} \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathbf{X} \boldsymbol{\alpha} \quad \text{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{\alpha} = 1. \tag{2.1}$$

For subsequent principal components, additional constraints are added to ensure that they are uncorrelated with previous principal components, so that each principal component axis captures different information in the data. Generally, for the rth PC, we have the optimization problem

$$\begin{aligned} \max_{\boldsymbol{\alpha}_r \neq \mathbf{0}} \quad & \boldsymbol{\alpha}_r^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathbf{X} \boldsymbol{\alpha}_r \\ & \text{subject to} \quad \boldsymbol{\alpha}_r^{\mathrm{T}} \boldsymbol{\alpha}_r = 1, \boldsymbol{\alpha}_s^{\mathrm{T}} \boldsymbol{\alpha}_r = 0 \\ & \forall s < r, \quad r = 2, \ldots, q \ll \min(p, n-1). \end{aligned} \tag{2.2}$$

Using Lagrangian multipliers, one can show that problem (2.2) results in the eigenvalue problem

$$\mathbf{X}^{\mathrm{T}} \mathbf{X} \boldsymbol{\alpha} = \lambda \boldsymbol{\alpha}. \tag{2.3}$$

Then the rth principal component loadings of $\mathbf{X}$ is the rth eigenvector that corresponds to the rth eigenvalue $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_r \geq \cdots \geq 0$ of the sample covariance matrix $\mathbf{X}^{\mathrm{T}}\mathbf{X}$. Of note, the magnitude, α_{rk} of each principal component loading $\tilde{\boldsymbol{\alpha}}_r = [\alpha_{r1}, \ldots, \alpha_{rk}, \ldots, \alpha_{rp}]$ represents the importance of the kth variable to the rth principal component, and these are typically nonzero. When $p \gg n$, interpreting the principal components is a difficult task because the principal components are linear combinations of all variables. Thus for high dimensional data, a certain type of regularization that ensures that some variables have negligible or no effect on the rth principal component is warranted to yield interpretable principal components.

To achieve sparsity of the principal component loadings while incorporating structural information $\mathcal{G}$, we utilize ideas in Safo and Ahn (Safo and Ahn, 2014) which is motivated by the Dantzig Selector for sparse estimation in regression problems. Specifically, we bound a modified version of the eigenvalue difference in (2.3) with a l_∞ norm while minimizing a structured-sparsity inducing penalty of the principal component loadings:

$$\min_{\boldsymbol{\alpha} \neq \mathbf{0}} \mathcal{P}(\boldsymbol{\alpha}, \tau) \text{ subject to } \quad \|\mathbf{X}^{\mathrm{T}}\mathbf{X}\tilde{\boldsymbol{\alpha}}_r - \tilde{\lambda}_r \boldsymbol{\alpha}\|_\infty \leq \tau \text{ and } \mathbf{A}_{r-1}^{\mathrm{T}}\boldsymbol{\alpha} = 0.$$

Here, for a random vector $\mathbf{z} \in \Re^p$, $\|\mathbf{z}\|_\infty$ is the l_∞ norm defined as $\max_{1 \leq i \leq p} |z_i|$, $\tau > 0$ is a tuning parameter that controls how many of the coefficients in the principal component loadings will be exactly zero. In addition, $\mathbf{A} = [\hat{\boldsymbol{\alpha}}_1, \ldots, \hat{\boldsymbol{\alpha}}_s]$ $\forall s < r$ is a concatenation of the previous sparse PCA solutions $\hat{\boldsymbol{\alpha}}_s$, and $\tilde{\boldsymbol{\alpha}}_r$ is the nonsparse rth PCA loading, which is the eigenvector corresponding to the rth largest eigenvalue of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$. In the next sections, we introduce sparse PCA methods that utilize the network information $\mathcal{G}$ in $\mathbf{X}$.

2.2.2 Grouped sparse PCA

The first approach we propose is the grouped sparse PCA, similar in spirit with Pan et al. (2010b). Utilizing the graph structure $\mathcal{G}$, we propose the following structured sparse PCA criterion for the rth principal component loading:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\neq\mathbf{0}} \quad & \left\{(1-\eta)\sum_{i\sim j}\left(\frac{|\alpha_i|^\gamma}{w_i}+\frac{|\alpha_j|^\gamma}{w_j}\right)^{1/\gamma}+\eta\sum_{d_i=0}|\alpha_i|\right\} \\ \text{subject to} \quad & \|\mathbf{X}^{\mathrm{T}}\mathbf{X}\tilde{\boldsymbol{\alpha}}_r-\tilde{\lambda}_r\boldsymbol{\alpha}\|_\infty\leq\tau \ \text{ and } \ \mathbf{A}_{r-1}^{\mathrm{T}}\boldsymbol{\alpha}=0, \end{aligned} \tag{2.4}$$

where $\|\cdot\|_\infty$ is the l_∞ norm , $\tau>0$ is a tuning parameter, $\gamma>1$ and $0<\eta<1$ are fixed, $\mathbf{A}_{r-1}=(\hat{\alpha}_1,\hat{\alpha}_2,\ldots,\hat{\alpha}_{r-1})$ is the matrix constituted of $r-1$ structured sparse PC loadings, and $\tilde{\boldsymbol{\alpha}}_r$ is the rth nonsparse PC loading vector, which is the eigenvector corresponding to the rth largest eigenvalue of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$.

The first term in the objective function (2.4) is the weighted grouped penalty of Pan et al. (2010b), which induces grouped variable selection. The penalty encourages both α_i and α_j to be equal to zero simultaneously, suggesting that two neighboring genes in a network are more likely to participate in the same biological process simultaneously. The second term in the objective function induces sparsity in selection of singletons that are not connected to any other variables in the network. The tuning parameter τ enforces some coefficients of the principal components to be exactly zero with larger values encouraging more sparsity. The selection of τ is usually data-driven, and is discussed in section 2.4. The optimization problem is convex in $\boldsymbol{\alpha}$ and can be solved with any off the shelf convex optimization package such as the CVX package CVX Research (2012) in Matlab.

2.2.3 Fused sparse PCA

The second structured sparse PCA is the Fused sparse PCA, which generalizes fused lasso (Tibshirani et al., 2005) to account for complex interactions within a pathway. Utilizing the graph structure $\mathcal{G}$, we propose the following structured sparse PCA for the rth principal component loading:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\neq\mathbf{0}} \quad & \left\{(1-\eta)\sum_{i\sim j}\left|\tfrac{\alpha_i}{w_i}-\tfrac{\alpha_j}{w_j}\right|+\eta\sum_{d_i=0}|\alpha_j|\right\} \\ \text{subject to} \quad & \|\mathbf{X}^{\mathrm{T}}\mathbf{X}\tilde{\boldsymbol{\alpha}}_r-\tilde{\lambda}_r\boldsymbol{\alpha}\|_\infty\le\tau \ \text{ and } \ \mathbf{A}_{r-1}^{\mathrm{T}}\boldsymbol{\alpha}=0 \end{aligned} \tag{2.5}$$

where $\tau_x > 0$ and $\tau_y > 0$ are tuning parameters, $0 \le \eta \le 1$ is fixed, $\mathbf{A}_{r-1} = (\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{r-1})$ is the matrix constituted of $r-1$ structured sparse PC loadings, and $\tilde{\boldsymbol{\alpha}}_r$ is the rth nonsparse PC loading vector. This penalty is a combination of weighted l_1 penalty on variables that are connected in the network and l_1 penalty on singletons that are not connected to any genes in the network. The first term in the objective function (2.5) is the fused structured penalty that encourages the difference between variable pairs that are connected in the network to be small and hence the variables to be selected together.

This penalty is similar to some existing penalties, but different in a number of ways. First, it is similar to the fused lasso—both attempt to smooth the coefficients that are connected in $\mathcal{G}$. However, the fused lasso does not utilize prior biological information. Instead, it uses a data-driven clustering approach to order the variables that are correlated and imposes l_1 penalty on the difference between coefficients of adjacent variables. It also does not weight neighboring features, which may allow one to enforce various prior relationships among features. Second, the Fused sparse penalty is also similar to the network constrained penalty of Li and Li (2008). Their penalty $\eta_1\sum_j|\alpha_j| + \eta_2\sum_{i\sim j}\left(\frac{\alpha_i}{w_i}-\frac{\alpha_j}{w_j}\right)^2$ uses the l_2 norm and it has been shown that

this does not produce sparse solutions, where sparsity refers to variables that are connected in a network. In other words, it does not encourage grouped selection of variables in the network (Pan et. al, 2010). Also, the additional tuning parameter η_2 increases computational costs for very large p since it requires solving a graph-constrained regression problem with dimension $(n + p) \times p$.

The two proposed methods differ in how the structural information is incorporated in the PCA problem. Grouped sPCA is dependent on γ in the L_γ norm and have different sparsity solution in the PC loadings for different γ. Unlike the fused sPCA, the weights in the grouped sPCA allow for two neighboring nodes to have opposite effects, which may be relevant in some biological process. However, in the fused sPCA, it is easy to understand that the l_1 norm difference of connected pairs allows variables that are connected or behave similarly to be close together, which is not so intuitive in the grouped sPCA.

2.2.4 Algorithms

We present two algorithms for the proposed structured sparse PCA methods. Algorithm 1 obtains the rth principal component loading vector for a fixed tuning parameter τ. Algorithm 2 provides a data driven approach for selecting the optimal tuning parameter value τ from a range of values. The normalization in step (3) of Algorithm 1 eases interpretation, and usually facilitates a visual comparison of the coefficients. Once the principal component loading vector is obtained, the coefficients (in absolute value) can be ranked to assess the contribution of the variables to a given PC. If the variables are measured on different scales or on a common scale with widely differing ranges, then it is recommended to standardize the variables to have unit variance before implementing the proposed methods.

Algorithm 1 is developed to obtain r PC loading vectors. For the best r, we can introduce tuning parameter selection in step (2) using, for example cross validation

Algorithm 1 Optimization for r structured sparse PC

1: Initialize with nonsparse estimates, $\tilde{\boldsymbol{\alpha}}_r$. These are the eigen-decomposition of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$, and let $\tilde{\boldsymbol{\alpha}}_r$ be the rth eigen-vector corresponding to the rth largest eigen-value $\tilde{\lambda}_r$ of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$. Here, one can use ideas in (Hastie and Tibshirani, 2004) for the eigen analysis of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ when p is very large.

2: For a fixed positive tuning parameter τ selected from a set of finite grid values, solve problem (2.4) or (2.5) for the rth grouped sPC or fused sPC vector, $\hat{\boldsymbol{\alpha}}_r$.

3: Normalize $\hat{\boldsymbol{\alpha}}_r$: $\hat{\boldsymbol{\alpha}}_r = \frac{\hat{\boldsymbol{\alpha}}_r}{\|\hat{\boldsymbol{\alpha}}_r\|_2}$.

Algorithm 2 Optimization for r structured sparse PC

1: **for** each τ in a set of fine grid from $(0, \tau_{\max})$, and for a desired number of principal components r, **do**

(i) Apply Algorithm 1 on $\mathbf{X}$ to derive the rth principal component loadings $\hat{\mathbf{A}}_r(\tau)$. Then project $\mathbf{X}$ onto $\hat{\mathbf{A}}_r(\tau)$ to obtain the best principal components as $\mathbf{Y}_r(\tau) = \mathbf{X}^{\mathrm{T}}\hat{\mathbf{A}}_r(\tau)$.

(ii) Calculate the BIC value defined as

$$BIC(\tau) = log\big[\frac{1}{np}\|\mathbf{X} - \mathbf{Y}_r(\tau)\hat{\mathbf{A}}_r^{\mathrm{T}}(\tau)\|_F\big] + \frac{\gamma_\tau log(np)}{np} \tag{2.6}$$

where $\|\cdot\|_F$ is the Frobenius norm and γ_τ is the number of non-zero components of $\hat{\mathbf{A}}_r(\tau)$.

2: **end for**

3: Select the optimal tuning parameter as $\tau_{opt} = \min_\tau\{BIC(\tau)\}$.

to maximize the total variance explained by the rth principal component, with the smallest r explaining some proportion of variance explained selected as the optimal rth principal component. This would add extra layer of complexity to the tuning parameter selection, however.

The tuning parameters $\tau = (\tau_1, \ldots \tau_r)$ control the model complexity and their optimal values need to be selected. We use Bayesian information criterion (BIC) (Allen et al., 2014) and implement Algorithm 2 to select τ that yields a better rank r approximation to the test data. Compared with using cross-validation to select best tuning parameters, BIC can be computationally more efficient, especially for large datasets.

2.3 Simulation

We conduct simulations to assess the performance of the proposed methods in comparison with several existing sparse PCA methods. We consider two simulation settings that differ by the proportions of variation explained by the first two PCs. In the first setting, the first two PCs explain 6% of the total variation which indicates that true signals in the data are weak. In the second setting, the first two PC's explain 30% of the total variation in the data, representing a case where signals are strong. Within each setting, we consider the dimensions $p = 500$ and $p = 10,000$, and also consider two scenarios that differ by the graph structure $\mathcal{G}$ for the proposed methods.

2.3.1 Simulation Settings

Let $\mathbf{X}$ be a $n \times p$ matrix and let $\mathbf{G}_0$ be the true covariance matrix used to generate $\mathbf{X}$. Let $\mathcal{G}_0$ be the corresponding graph structure. The true covariance matrix $\mathbf{G}_0$ is

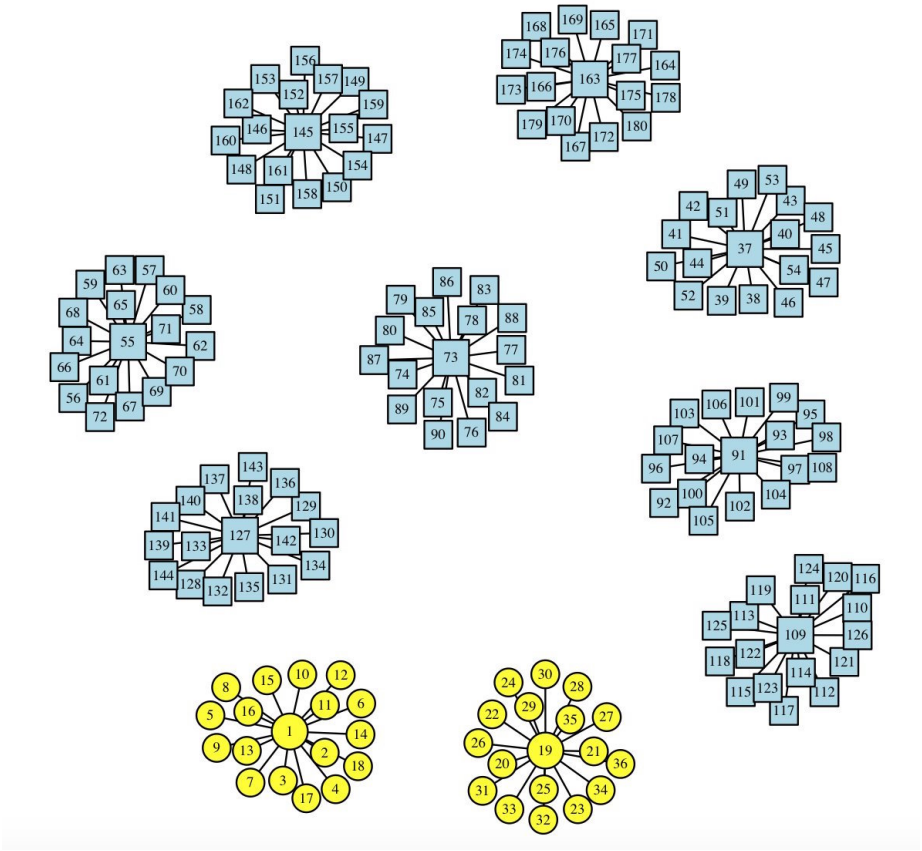


Figure 2.1: Network structure of simulated data : Correctly specified graph. Variables in circle represent signals, and square represent noise. ($\mathcal{G} = \mathcal{G}_0$)

partitioned as

$$\mathbf{G}_0 = \begin{pmatrix} \mathbf{G}_{00} & \mathbf{0} \\ \mathbf{0} & \nu \times \mathbf{I}_{p-36} \end{pmatrix},$$

where $\mathbf{G}_{00}$ is block diagonal with ten blocks each of size 18 for $p = 500$ and size 250 for $p = 10,000$, and between block correlation 0. We set the variance of variables in the first two blocks to be 1, and 0.3 for the remaining eight blocks. In addition, we set the correlation of a main and connecting variable to be 0.9 for the first two blocks and 0.2 for the other blocks. Meanwhile, we let the correlation $\rho_{ik} \sim Uniform(0.7, 0.8), i \neq k \; and \; i, k \geq 2$ for the first two blocks, and $\rho_{ik} \sim Uniform(0, 0.2), i \neq k \; and \; i, k \geq 2$ for the other blocks. This type of covariance matrix $\mathbf{G}_0$ suggests that data structure is determined by ten underlying subnetworks, where the first two PCs of the first two subnetworks are mostly important in detecting signals in the data. In other words, in both settings, the true PCs has 36 important variables and $p - 36$ noise variables when $p = 500$, and $p = 500$ important variables and $p - 500$ noise variables for $p = 10,000$. We note that by changing the value of ν, we control the proportions of variation explained by the first two PCs. For each setting, we specify $n = 100$, and simulate $\mathbf{X}$ from MVN($\mathbf{0}$, $\mathbf{G}_0$). We use 0.5 for all η in the simulation study.

For each setting and dimension, we consider two scenarios that differ by the graph structure $\mathcal{G}$ specified in the proposed sPCA methods. In the first scenario, the graph structure is correctly specified, that is $\mathcal{G} = \mathcal{G}_0$. This corresponds to the situation where all true structural information are available in $\mathcal{G}$ so that $\mathcal{G}$ is informative. The resulting network includes 500 variables and 170 edges between each main variable and connecting variable when p equals 500 (or $10,000$ variables and $2,490$ edges when p equals $10,000$), i.e.,$E = \{i \sim j | i, j = 1, \cdots, 180\}$ in $\mathcal{G}$. Figure 2.1 is a graph of the network $\mathcal{G}$ used in Fused and Grouped sPCA when network information is correctly specified.

In the second scenario, the graph structure is randomly generated and does not capture the true information in the data. The resulting network includes a total of 170 random edges when p equals 500 (or $2,490$ edges when p equals $10,000$). This setting assesses the performance of the proposed methods in cases where the structural information is uninformative and sheds light on robustness of the proposed methods. Figure A.1 in the Supplementary shows the graph structure for randomly specified edges.

Performance Metrics We compare the proposed methods Grouped PCA and Fused PCA to the traditional PCA (Hotelling, 1936), SPCA (Zou et al., 2006) and SPC (Witten et al., 2009b). We implement SPCA and SPC using the R-packages *elasticnet* and *PMA* respectively. We evaluate the performance of the methods using the following criteria.

- *Reconstruction error*: $||\mathbf{X}_{test}\mathbf{A}\mathbf{A}^T - \mathbf{X}_{test}\hat{\mathbf{A}}\hat{\mathbf{A}}^{\mathrm{T}}||_F^2$, where $\mathbf{A} = (\boldsymbol{\alpha}_1\ \boldsymbol{\alpha}_2)$ are the true PC loadings and $\hat{\mathbf{A}} = (\hat{\boldsymbol{\alpha}}_1\ \hat{\boldsymbol{\alpha}}_2)$ are the estimated PC loadings. This criterion tests the methods ability to approximate the testing data reconstructed using only the first two PC loadings, since if $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$ is the spectral decomposition of the centered data $\mathbf{X}$, then $\mathbf{V}$ are the eigenvectors of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ and $\mathbf{X}\mathbf{V} = \mathbf{U}\mathbf{D}$. If span($\mathbf{A}$)= span($\mathbf{V}$), then $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{A}^{\mathrm{T}} = \mathbf{X}\mathbf{A}\mathbf{A}^{\mathrm{T}}$, and if span($\mathbf{A}$) $\subset$ span($\mathbf{V}$), then $\mathbf{X}\mathbf{A}\mathbf{A}^{\mathrm{T}}$ is an approximation to the data $\mathbf{X}$.

- *Estimation error*: $||\mathbf{A}\mathbf{A}^T - \hat{\mathbf{A}}\hat{\mathbf{A}}^{\mathrm{T}}||_F^2$. This criterion tests the methods ability to estimate the linear subspace spanned by the true PC loadings (Cai et al., 2013), with a smaller estimate preferred.

- *Selectivity*: We also test the methods ability to select the right variables while ignoring noise variables using sensitivity and specificity which are defined as $Sensitivity = \frac{\#\text{ of True Positive}}{\#\text{ of True Positive+\# of False Negative}}$, $Specificity = \frac{\#\text{ of True Negative}}{\#\text{ of True Negative+\# of False Positive}}$. Sensitivity and specificity capture the accuracy of estimated PC loadings with

high values indicating better performance.

- *Proportion of variance explained*: The fourth comparison criterion is the proportion of variation explained in the testing and training data sets by the first two PC loadings, which is defined as $\frac{\hat{\boldsymbol{\alpha}}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\hat{\boldsymbol{\alpha}}}{\mathrm{trace}(\mathbf{X}\mathbf{X}^{\mathrm{T}})}$, where $\mathbf{X}$ is either the centered training or testing data set, and $\hat{\boldsymbol{\alpha}}$ is the estimated first or second PC.

2.3.2 Simulation Results

Table 2.1 shows the performance of the methods for the first setting where the first two PCs explain only 6% of the total variation in the data. We observe that the proposed methods are competitive for $p = 500$ and even more so when $p = 10,000$. In particular, Grouped sPCA has smaller reconstruction and estimation errors when the graph structure is correctly specified and even when the graph structure is uninformative. On the other hand, Fused sPCA shows a suboptimal performance in comparison to Grouped sPCA, yet better or competitive performance when compared to the traditional PCA and SPCA for correctly specified graph structure and mis-specified graph structure. In terms of sensitivity and specificity, we observe that both Grouped sPCA and more especially Fused sPCA are better in detecting signals even when the graph structure is mis-specified, while Grouped sPCA is more competitive at not selecting noise variables. We also notice that both Grouped sPCA and Fused sPCA have good performance in proportions of cumulative variation explained compared with existing sparse PCA methods, especially compared with SPCA. In Table 2.2 where the first two PC's explain 30% of the total variation in the data, we observe a similar performance of the proposed methods.

A comparison between $p = 500$ and $p = 10,000$ scenarios for both settings indicates that the gain in reconstruction error, estimation error, sensitivity, and proportions of variation explained can be substantial for Grouped sPCA and Fused sPCA compared with the existing sparse PCA methods, as the number of variables increases.

Table 2.1: Simulation results of Setting 1. Cumulative proportions of variance explained by true PCs are 0.03 for PC 1 and 0.06 for PC 1 and 2. P, number of variables. RE, reconstruction error, defined as $||\mathbf{X}_{test}\mathbf{A}\mathbf{A}^T - \mathbf{X}_{test}\hat{\mathbf{A}}\hat{\mathbf{A}}^{\mathrm{T}}||_F^2$, where $\mathbf{A} = (\boldsymbol{\alpha}_1\ \boldsymbol{\alpha}_2)$. EE, estimation error, defined as $||\mathbf{A}\mathbf{A}^T - \hat{\mathbf{A}}\hat{\mathbf{A}}^{\mathrm{T}}||_F^2$. cPVE, proportions of cumulative variation explained. $\cdot(\cdot)$, mean(std).

Method	RE	EE	Sensitivity		Specificity		cPVE	
			1stPC	2ndPC	1stPC	2ndPC	1stPC	2ndPC
P = 500								
PCA	31 (9e-1)	1.1 (3e-2)	1.0	1.0	0.0	0.0	4.3e-2 (2e-3)	8.2e-2 (2e-3)
SPCA	34 (3)	1.2 (1e-1)	0.54	0.50	0.95	0.90	2.0e-2 (2e-3)	4.0e-2 (4e-3)
SPC	16 (8)	0.57 (3e-1)	0.57	0.60	0.98	1.0	2.8e-2 (3e-3)	5.5e-2 (6e-3)
biological information correctly specified								
Fused sPCA	25 (6)	0.90 (2e-1)	1.0	1.0	0.73	0.70	2.9e-2 (4e-3)	5.1e-2 (7e-3)
Grouped sPCA	8.0 (6)	0.29 (2e-1)	0.81	0.80	0.97	1.0	3.2e-2 (2e-3)	6.0e-2 (3e-3)
biological information randomly specified								
Fused sPCA	32 (4)	1.1 (2e-1)	0.95	1.0	0.51	0.51	3.0e-2 (4e-3)	5.2e-2 (7e-3)
Grouped sPCA	9.1 (6)	0.33 (2e-1)	0.81	0.80	0.97	1.0	3.2e-2 (2e-3)	5.9e-2 (3e-3)
P = 10,000								
PCA	112 (3)	1.3 (2e-2)	1.0	1.0	0.0	0.0	2.6e-2 (1e-3)	5.0e-2 (1e-3)
SPCA	160 (4)	1.9 (3e-2)	0.15	0.15	0.99	0.99	2.3e-3 (5e-4)	4.5e-3 (7e-4)
SPC	172 (4)	2.0 (8e-3)	0.01	0.01	1.0	1.0	1.7e-4 (1e-4)	3.4e-4 (3e-4)
biological information correctly specified								
Fused sPCA	81 (50)	0.94 (0.5)	0.62	0.55	0.99	0.99	1.2e-2 (6e-3)	2.2e-2 (1e-2)
Grouped sPCA	54 (40)	0.62 (0.4)	0.62	0.58	0.99	1.0	1.4e-2 (3e-3)	2.6e-2 (6e-3)
biological information randomly specified								
Fused sPCA	140 (30)	1.6 (0.4)	0.60	0.60	0.68	0.68	8.9e-3 (5e-3)	1.6e-2 (1e-2)
Grouped sPCA	58 (40)	0.67 (0.5)	0.59	0.55	0.99	1.0	1.4e-2 (3e-3)	2.6e-2 (7e-2)

This suggests that Grouped sPCA or Fused sPCA can achieve sparse PC loading estimations with higher accuracy, better variable selection, and larger proportion of variation explained, especially when the number of variables is relatively large.

2.4 Application to the Glioblastoma Data

We apply the proposed methods to analyze data from a Glioblastoma cancer study. Glioblastoma brain multiform (GBM) is the most common malignant brain tumor and is defined as grade IV astrocytoma by the Whold Health Organization because of its aggressive and malignant nature (Furnari et al., 2007a). The Cancer Genome Atlas Project (TCGA) (McLendon et al., 2008) integratively analyzed genome information of patients with glioblastoma and expanded the knowledge about the pathways and genes that may relate with glioblastoma. In our data analysis, we obtain part of the genomic data from TCGA project for glioblastoma, which is explained in detail by

Table 2.2: Simulation results of Setting 2. Cumulative proportions of variance explained by true PCs are 0.15 for PC 1 and 0.30 for PC 1 and 2. P, number of variables. RE, reconstruction error, defined as $||\mathbf{X}_{test}\mathbf{A}\mathbf{A}^T - \mathbf{X}_{test}\hat{\mathbf{A}}\hat{\mathbf{A}}^{\mathrm{T}}||_F^2$, where $\mathbf{A} = (\boldsymbol{\alpha}_1\ \boldsymbol{\alpha}_2)$. EE, estimation error, defined as $||\mathbf{A}\mathbf{A}^T - \hat{\mathbf{A}}\hat{\mathbf{A}}^{\mathrm{T}}||_F^2$. cPVE, proportions of cumulative variation explained. $\cdot(\cdot)$, mean(std).

Method	RE	EE	Sensitivity		Specificity		cPVE	
			1stPC	2ndPC	1stPC	2ndPC	1stPC	2ndPC
P = 500								
PCA	31 (0.9)	1.1 (3e-2)	1.0	1.0	0.0	0.0	4.3e-2 (2e-3)	8.2e-2 (2e-3)
SPCA	35 (2)	1.3 (9e-2)	0.49	0.50	0.95	1.0	1.9e-2 (3e-3)	3.9e-2 (4e-3)
SPC	15 (7)	0.54 (3e-1)	0.57	0.60	0.98	1.0	2.8e-2 (3e-3)	5.6e-2 (5e-3)
biological information correctly specified								
Fused sPCA	27 (4)	0.93 (2e-1)	1.0	1.0	0.70	0.70	3.0e-2 (3e-3)	5.3e-2 (5e-3)
Grouped sPCA	7.9 (5)	0.29 (2e-1)	0.80	0.80	0.97	1.0	3.2e-2(2e-3)	6.0e-2 (3e-3)
biological information randomly specified								
Fused sPCA	32 (5)	1.1 (2e-1)	0.96	1.0	0.52	0.50	2.9e-2 (5e-3)	5.1e-2 (8e-3)
Grouped sPCA	9.2 (6)	0.33 (0.2)	0.79	0.8	0.97	1.0	3.2e-2 (2e-3)	5.9e-2 (4e-3)
P = 10,000								
PCA	112 (3)	1.3 (2e-2)	1.0	1.0	0.0	0.0	2.7e-2 (1e-3)	5.0e-2 (1e-3)
SPCA	162 (4)	1.9 (3e-2)	0.16	0.16	1.0	1.0	2.0e-3 (5e-4)	4.0e-3 (8e-4)
SPC	173 (4)	2.0 (5e-3)	5.0e-3	5.0e-3	1.0	1.0	1.6e-4 (1e-4)	3.2e-4 (2e-4)
biological information correctly specified								
Fused sPCA	77 (40)	0.89 (0.5)	0.65	0.57	0.99	1.0	1.3e-2 (5e-3)	2.3e-2 (9e-3)
Grouped sPCA	46 (30)	0.53 (0.4)	0.65	0.62	0.99	1.0	1.5e-2 (2e-3)	2.8e-2 (5e-3)
biological information randomly specified								
Fused sPCA	140 (30)	1.6 (0.4)	0.59	0.60	0.68	0.70	9.0e-3 (5e-3)	1.7e-2 (1e-2)
Grouped sPCA	53 (40)	0.61 (0.4)	0.63	0.60	0.99	1.0	1.5e-2 (3e-3)	2.7e-2 (6e-3)

McLendon et al. (2008),Verhaak et al. (2010), and Cooper et al. (2010). This data set contains microarray data of 558 subjects with glioblastoma. The GBM subtype of each subject is also given.

The goal of the analysis is to identify a subset of relevant genes that contribute to the variation in the different GBM subtypes, and also determine how the first two estimated PCs separate these subtypes. For both datasets, we first select $2,000$ variables with the largest variation following the data preprocessing procedure in Witten et al. (2009b). In the next step, we select patients with subtype *Classical*, *Mesenchymal*, *Neural*, and *Proneural* following the previous work by Verhaak et al. (2010) resulting in 481 patients with subtype data. We obtain the gene network information for Fused and Grouped sparse PCA methods from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database (Kanehisa and Goto, 2000). The resulting network has $2,000$ genes and $1,297$ edges in the network. We center each variable to have

mean 0 and standardize each variable to have variance one. Again, we use 0.5 for η in real data experiment.

To justify the structural information we use for the proposed methods, we conduct exploratory analysis using correlation coefficients of gene pairs. We group the gene pairs consisting of the selected 2000 genes into three categories: unconnected gene pairs (two genes that are not in any pathway), direct-connected gene pairs (two genes that have a direct edge connecting them), indirect-connected gene pairs (two genes that belong to the same pathway but do not have a direct edge connecting them) according to the KEGG Pathway information and we use boxplots to demonstrate the correlation coefficients of these three types of gene pairs. Figure A.4 shows the plot of correlation coefficients of gene pairs by their categories. There is a small but clear decreasing trend in correlation coefficients as one moves from direct-connected gene pairs to unconnected gene pairs. This shows that the gene pairs that are directly connected tend to have stronger correlations than those that are indirectly connected or unconnected, thus justifying the validity of pathway information we use in the analysis.

In the analysis, we equally split each data set into training and testing sets, where the training set is used to estimate the optimal tuning parameters via 5-fold cross validation. The trace plots of tuning parameters for Grouped sPCA and Fused sPCA are shown in Figure A.2. We then apply the optimal parameters on the whole training set to estimate the first two PC loadings $\hat{\boldsymbol{\alpha}}_i, i = 1, 2$, and use the testing set to evaluate the estimated loadings using the following two criteria:

$$\text{Number of non-zero loadings of } \hat{\boldsymbol{\alpha}}_i = \boldsymbol{\Sigma}_{j=1}^{2000} I\{\hat{\boldsymbol{\alpha}}_{ij} \neq 0\}, \quad i = 1, 2;$$

$$\text{Proportion of variation explained by } \hat{\boldsymbol{\alpha}}_i = \frac{\hat{\boldsymbol{\alpha}}_i^{\mathrm{T}} \mathbf{X} \hat{\boldsymbol{\alpha}}_i}{\text{trace}(\mathbf{X}\mathbf{X}^T)}, \quad i = 1, 2,$$

where $\mathbf{X}$ is the centered training or testing data matrix. We also obtain the first two

Table 2.3: Analysis of the GBM Data using Kegg Pathway information. cPVE represents proportions of cumulative variation explained. Classification results are based on 500 repeats.

Method	Non-zero Loadings		cPVE		Subjects correctly classified
	1stPC	2ndPC	1stPC	2ndPC	SVM
PCA	2000	2000	0.1955	0.3175	97
SPCA	240	238	0.0333	0.0591	97
SPC	45	59	0.0215	0.0383	67
Fused sPCA	1644	1410	0.1792	0.2787	123
Grouped sPCA	1330	970	0.1731	0.2652	119

PCs $\hat{\boldsymbol{\beta}}_i$ by $\hat{\boldsymbol{\beta}}_i = \mathbf{X}\hat{\boldsymbol{\alpha}}_i$, $i = 1, 2$ and determine how well they separate patients with different GBM subtypes using support vector machine (SVM).

Table 2.3 shows the number of non-zero loadings, the cumulative proportions of variation explained by the first two PC loadings, and the classification results using SVM. We find that SPC and SPCA are more sparse than the Fused sparse PCA and the Grouped sparse PCA. This is consistent with the simulation settings where SPC and SPCA tend to be more sparse and have higher false negatives that result in lower sensitivity. Regarding cumulative proportions of variation explained, we find that the proposed methods explain higher variation in the data, but this may be due to the large number of variables selected. The last column of table 2.3 gives the classification results from applying SVM on the testing set using the estimated first two PC loadings. The Fused and Grouped sparse PCA have the highest number of correctly specified subjects. Of the existing methods, PCA and SPCA achieve good performance of separating patients with different subtypes, while SPC has the lowest number of subjects correctly classified.

We also conduct pathway enrichment analysis using bioinformatics software ToppGene Suite (Chen et al., 2009). Specifically, we identify the genes that have non-zero loadings in the first PC from the proposed sparse PCA methods and existing methods, and obtain significantly enriched pathways that are associated with glioblastoma for each method. We seek to identify methods that have more glioblastoma-associated

Pathway ID	Pathway name	P-value	Gene	
			from input	in annotation
	Fused sPCA			
739007	Spinal cord injury	7.43E-18	45	112
782000	Proteoglycans in cancer	5.77E-11	55	225
523016	Transcriptional misregulation in cancer	3.312E-7	40	179
83105	Pathways in cancer	3.36E-7	61	327
83115	Bladder cancer	6.10E-6	14	38
	Grouped sPCA			
739007	Spinal Cord Injury	1.97E-14	36	112
523016	Transcriptional misregulation in cancer	4.06E-7	34	179
198901	SIDS Susceptibility Pathways	2.57E-6	30	160
83105	Pathways in cancer	2.58E-5	46	327
P00005	Angiogenesis	4.90E-5	26	150
	SPC			
739007	Spinal Cord Injury	1.43E-5	5	112
	SPCA			
739007	Spinal Cord Injury	6.46E-5	8	112

Table 2.4: Enriched Glioblastoma-related pathways for the genes in first PC by different sPCA methods

pathways, and whether these overlap. Table 2.4 shows the Glioblastoma-related pathways found by the proposed methods and existing sparse PCA methods. Among the existing sparse PCA methods, both SPC and SPCA find Spinal Cord Injury pathway. Compared with the existing methods, Fused and Grouped sPCA find a few new Glioblastoma-related pathways: Proteoglycans in cancer, Transcriptional misregulation in cancer, Pathways in cancer, Bladder cancer, SIDS Susceptibility Pathways, and Angiogenesis. We also plot the first two PC loadings by Fused and Grouped sPCA in Figure A.3 and the loadings of genes enriched in Glioblastoma-related pathways are highlighted in color. These results indicate that the proposed methods may be more sensitive in detecting disease related signals and thus can identify more biologically important genes.

2.5 Discussion

In this paper, we propose two novel structured sparse PCA methods. Through extensive simulation studies and an application to Glioblastoma gene expression data,

we demonstrate that incorporating known biological information improves the performance of sparse PCA methods. Specifically, our simulation study indicates that the proposed methods can decrease reconstruction and estimation errors, and increase sensitivity and proportions of variation explained, especially when number of variables is large. Compared with Fused sPCA and existing PCA methods, Grouped sPCA achieves the lowest reconstruction error and estimation error for correctly specified and mis-specified graph structure. On the other hand, Fused sPCA has higher sensitivity values. Because we utilize prior biological information, the proposed methods usually have less sparse PC loadings compared with the existing sPCA methods and thus lower specificity. However, there is a trade-off between sparsity and the benefit from extra information. Consistent with the simulations results, the real data analysis demonstrates that the proposed methods generate less sparse PC loadings. However, the classification results show the advantages of incorporating biological information into sparse PCA.

The proposed methods require the structure of variables to be known in advance and specified during analysis. In real data analysis, this task is not trivial and it may take some efforts in searching for a proper variable structure to use. Regarding this, we make the following comments. First of all, many sources of structural information may be available to use including KEGG pathway (Kanehisa and Goto, 2000), Panther pathway (Mi et al., 2016), Human protein reference database (Prasad et al., 2009). It may be helpful to conduct some exploratory analysis such as Figure A.4 to confirm the need for using biological information. Figure A.4 demonstrates that gene pairs connected in the same pathway generally have higher correlation than gene pairs unconnected in the same pathway, and further than gene pairs in different pathways. Second, our simulation study indicates that even if the structural information is irrelevant as in the biological information randomly specified section, the proposed methods still performs well, especially Grouped sPCA method.

Our proposed methods have some limitations. First, when structural information includes a large number of edges, the proposed methods, particularly, Fused sPCA, may generate PC loadings that include more false positive selections. To solve this problem, one potential approach is to obtain a smaller but more relevant biological structure. Second, the proposed methods, especially Grouped sPCA may be computationally slow in the presence of a large number of edges. Based on our experience with the motivating data set, Fused sPCA is computationally more efficient than Grouped sPCA.

Some extensions are of potential interest. One may use alternative convex optimization solvers other than the CVX solver in matlab used in our work, potentially to speed up the computations. In addition, Fused and Grouped sPCA only incorporate the edge information in a graph. As variables are often grouped into pathways, sPCA using hierarchical penalties (Zhao et al., 2016) can be developed to incorporate group membership informaiton in addition to edge information.

Chapter 3

Distributed learning from multiple EHR databases : Contextual embedding models for medical events

3.1 Introduction

As promoted by the 2009 Health Information Technology for Economic and Clinical Health (HITECH) Act, more than half of the office-based physician practices adopted Electronic Health Records (EHR) systems to store patient clinical documents at the end of 2011 Hsiao et al. (2011). As a substitution of paper-based records, EHR systems usually include information from multiple clinical aspects, such as syndromes, laboratory test results, prescriptions, diagnoses, and doctor notes. This information not only contains health history and disease progression of each patient, but also reflects how diseases are treated in general. Thus EHR data is a rich depository for understanding disease features and treatment regimes.

Although EHR data are informative, analyzing EHR data can be a challenging task. As described in Hripcsak Hripcsak and Albers (2013) and Cheng Cheng et al. (2016+), EHR data are usually complex, sparse and of high-dimensionality. The huge volume of EHR data can make storage non-trivial, and analyzing such data is even more difficult. Besides its complexity, obtaining access or sharing EHR data can also be complicated, since the use of EHR data involves high-level privacy-preserving requirements Hodge Jr et al. (1999).

There has been a growing body of research studies on analysis of EHR data. Existing works mainly utilize information from EHR data for two goals: medical event phenotyping and predictive model construction. For the task of extracting features from EHR data (i.e. medical event phenotying), Batal Batal et al. (2012) uses temporal pattern mining to obtain abstraction sequences, which can be further applied in predictive models. Liu Liu et al. (2015) uses temporal graphs to construct graph-based frameworks from EHR data and to understand phenotype relationships. At the same time, the direction of obtaining low dimensional vector representation for medical events using deep-learning methods has advanced quickly. Several works have

published applications on this topic Choi, Bahadori, Searles, Coffey and Sun (2016); Choi (n.d.); De Vine et al. (2014); Lasko et al. (2013); Che et al. (2015). These works demonstrate that the obtained medical events provide clinically meaningful interpretation. In addition, they can be applied in downstream analyses, such as identifying disease susceptible populations or disease subtypes as well as making predictions for diagnoses or clinical outcomes.

After medical event phenotyping, prediction model construction is one of the most important downstream tasks. Many attempts have been made recently and been shown promising. Wu Wu et al. (2010) applies several machine learning approaches including logistic regression, Support Vector Machine (SVM), and Boosting on EHR data and they find that logistic regression with Bayesian Information Criterion (BIC) achieves the best accuracy for Heart Failure. Cheng Cheng et al. (2016+) uses a four-layer convolutional neural network model to predict the occurrence of Congestive Heart Failure and Chronic Obstructive Pulmonary Disease. Other deep learning-based predictive models include the work for Heart Failure prediction Choi, Schuetz, Stewart and Sun (2016a,b) and Parkinson's Disease Hammerla et al. (2015).

Although many predictive methods have been proposed, limitations and gaps still exist in real world applications. For one thing, most of the models proposed target only one or two diseases. In real life, at least a number of common diseases should be considered when a doctor sees a new patient. And for another, almost all the current methods assume that training data come from one dataset or training data come from multiple datasets but are available from one single place. It could happen that more than one databases are available at different sites but cannot be transferred or shared between data warehouses. Thus it would be preferable if a model can learn from multiple EHR datasets and at the same time predict the occurrence of multiple diseases.

In this article, we propose the Distributed Noise Contrastive Estimation (Dis-

tributed NCE), a neural-network-based technique for building predictive models of patient diagnoses. This method can learn from multiple EHR databases without sharing data among sites and make predictions for more than seventy common diseases. Our work is an extension of the Word2Vec model by Mikolov Mikolov, Chen, Corrado and Dean (2013) and the Patient-Diagnosis Projection Similarity Model by Farhan Farhan et al. (2016+). The main contribution of our work is the proposal of potential solutions to incorporate information from multiple data sources in one global model. As more methods have been developed based on Word2Vec, the proposed approach can be generalized to any Word2Vec-based models or other neural network based models to expand data sources and protect patient privacy.

The remainder of this article is structured as follows. The prelimary works are presented in section 2, problem setting and the proposed method Distributed NCE in section 3, two alternative methods including Naive Updates and Dropout Updates in section 4, the numerical study using real data in section 5, and some comments and discussions in section 6.

3.2 Preliminaries

This section reviews two existing models, which form the foundation of the proposed methods. We first briefly discuss the skip-gram (SG) model, a model formulation used in the Word2Vec. This model has a three-layer neural network and uses each single word to predict the context words surrounding it. Then we review the Patient-Diagnosis Projection Similarity (PDPS) model on the basis of SG model.

3.2.1 Skip-gram Model

Following the notation in Mikolov Mikolov, Sutskever, Chen, Corrado and Dean (2013), suppose there is a training sequence consisted of words $\omega_1, \omega_2, \ldots, \omega_T$. A three-layer neural network is constructed as demonstrated in Figure 3.1. The input

layer is a vector representing a single word in the training sequence and the output layer is vectors representing the context words around the input word. The hidden layer has the same number of nodes as the dimension of vector representation. The weight matrix from input layer to hidden layer is denoted by W and the weight matrix from hidden layer to output layer by $W^{'}$. The goal is to maximize the log-likelihood of context words given an input word:

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} log\mathcal{P}(\omega_{t-c},\ldots,\omega_{t-1},\omega_{t+1},\ldots,\omega_{t+c}|\omega_t)\\ = \frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-c\leq j\leq c\\ j\neq 0}} log\mathcal{P}(\omega_{t+j}|\omega_t)\end{aligned} \tag{3.1}$$

ω_t is the input word and $\omega_{t-c},\ldots,\omega_{t-1},\omega_{t+1},\ldots,\omega_{t+c}$ are the context words around ω_t.

Define the conditional probability of observing an output word (or an aforementioned context word) $\omega_O \in \{\omega_{t-c},\ldots,\omega_{t-1},\omega_{t+1},\ldots,\omega_{t+c}\}$ given the input word ω_I as:

$$\mathcal{P}(\omega_O|\omega_I) = \frac{exp({\nu^{'}_{\omega_O}}^T \nu_{\omega_I})}{\sum_{\omega=1}^{n_w} exp({\nu^{'}_{\omega}}^T \nu_{\omega_I})} \tag{3.2}$$

Here n_w is the number of words in the vocabulary, ν_{ω_I} is the corresponding weight vector for word ω_I in matrix W, and $\nu^{'}_{\omega_O}$ is the corresponding weight vector for word ω_O in matrix $W^{'}$. Note that if ω_I is the $i-$th word in vocabulary and ω_O is the $j-$th word, ${\nu_{\omega_I}}^T$ and $\nu^{'}_{\omega_O}$ are actually the $i-$th row of W and the $j-$th column of $W^{'}$ representing the input and output vector of ω_I and ω_O. Thus the SG model adjusts two weight matrices in neural network W and $W^{'}$ by maximizing the sum of log-likelihood for pairs consisted of all input words and their context words.

3.2.2 Patient-Diagnosis Projection Similarity Model

Farhan Farhan et al. (2016+) proposed a Word2Vec-based model, known as the patient-diagnosis projection similarity (PDPS), to predict patient diagnoses based on EHR data. Assume one has obtained vector for all events by building an SG model on training datasets. Given a new patient sequence S which consists of his/her medical event codes ordered by time, the goal is to predict the true diagnosis d^* using S and the vector of all medical events. For an event e, denote the vector representation of e by V_e. Specifically, represent diagnosis by d and denote the vector representation for d by V_d. In order to incorporate the time effect of each medical event in sequence S, we calculate the time elapsed between the event e and the last event denoted by t_e. To find the optimal prediction of diagnosis for patient with EHR sequence S, PDPS finds

$$d^* = \arg \max_{d \in Diag} CS(V_d, \frac{\sum_{e \in s} V_e exp(-\lambda t_e)}{\sum_{e \in S} exp(-\lambda t_e)}) \tag{3.3}$$

where $CS(x, y)$ is the cosine similarity between x and y, λ is the decay factor and usually takes a value between 0 and 1, $Diag$ is the list of all diagnoses in vocabulary. Besides the prediction for true diagnosis, PDPS can also predict one's probability for a specified diagnosis $\tilde{d}$ with $CS(V_{\tilde{d}}, \frac{\sum_{e \in s} V_e exp(-\lambda t_e)}{\sum_{e \in S} exp(-\lambda t_e)})$ as long as $\tilde{d}$ can be learned in training datasets. Using the dataset in our simulation study as an example, PDPS can make patient-diagnosis prediction for more than seventy common diagnoses based on this dataset, which is an advantage of PDPS over other existing prediction models. But the current PDPS cannot learn from multiple EHR databases without requiring data sharing, which can be a serious limitation when hospitals are reluctant to share data and merging data from multiple data warehouse is infeasible due to regulatory and other hurdles.

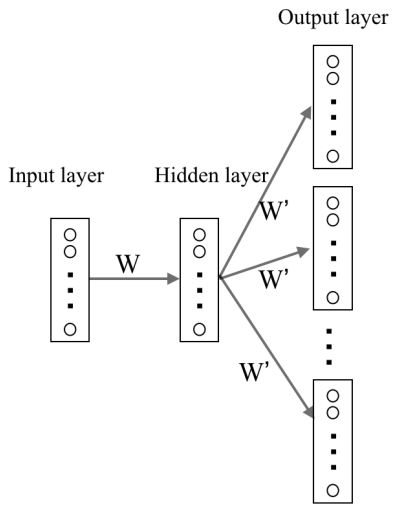


Figure 3.1: Demonstration of SG model structures. One square is a vector representation of one word. Circles represent elements in each vector. W is the weight between input layer and hidden layer, $W^{'}$ is the weight between hidden layer and output layer.

3.2.3 Distributed Noise Contrastive Estimation

Our idea of the Distributed NCE is inspired by the main hurdle of learning from multiple databases - discrepancies between vocabularies. Our proposed solution is to obtain vocabularies from multiple sites first and initialize an empty neural network model. Then this model can be trained using separate dataset locally in a sequential order. After all the datasites have trained the model, a global model is built and the weights between input and hidden layers are the final vector for medical events. This method is effective because it exactly mimics the process of training a Word2Vec model. The original training process feeds batches of word corpus into the gradient-descent algorithms, which is what Distributed NCE would do except that Distributed NCE feeds corpus in two separate locations. The Distributed NCE method is demonstrated in Figure 3.2. NCE is an optimization technique which is use to build neural network and improve efficiency Mnih and Kavukcuoglu (2013).

Suppose one have access to two databases D_1 and D_2, the vocabulary of database

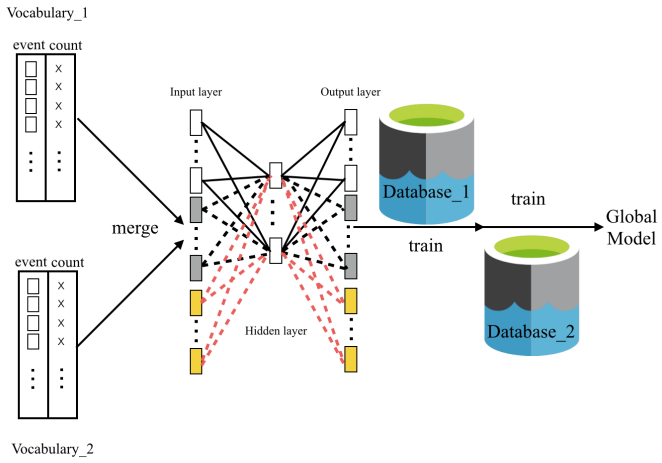


Figure 3.2: Distributed NCE. Squares represent medical events. Crosses represent counts of medical events. After obtain the vocabularies from datasets D_1 and D_2, the event lists and event counts are merged. Neural network is trained sequentially using D_1 and D_2.

D_1 has size n_w and the vocabulary of database D_2 has n_w^* new words. In the SG model, objective function(3.1) is unchanged. The only difference is that function(3.2) is revised to

$$\mathcal{P}(\omega_O|\omega_I) = \frac{exp({\nu'_{\omega_O}}^T \nu_{\omega_I})}{\sum_{\omega=1}^{n_w+n_w^*} exp({\nu'_{\omega}}^T \nu_{\omega_I})} \tag{3.4}$$

In implementation, we notice that Distributed NCE may have privacy concerns in the step of obtaining vocabulary counts if only two datasets are involved. The fact that one site can infer the medical event word counts of the other site from the gobal model may leak private information of patients, especially those with rare diseases. To address such concerns, the following section discusses how to add privacy protection in combined vocabulary and word counts.

3.2.4 Distributed Noise Contrastive Estimation with Privacy Protection

To protect data privacy when two datasets are analyzed by the Distributed NCE, a differential privacy (DP) component Xiao et al. (2012) is added into Distributed NCE. We call this method Distributed NCE with DP.

The DP method is proposed by Xiao Xiao et al. (2012) and its full name is *differentially private histogram release through multidimensional partitioning*. Given a vocabulary and the count table of all the words in vocabulary, to implement DP, the first step is to partition the count table into a few clusters. We use common methods such as k-means clustering or hierarchical clustering to partition data. In the next step, every value in one cluster is replaced by the mean value plus a part of a random Laplace noise. The general workflow of adding DP to a one-dimensional data is illustrated in Figure 3.3.

Suppose the whole count table $x_1, \ldots, x_N$ is divided into s clusters using some clustering technique. N is the number of words in vocabulary. The $i-$th cluster contains elements $x_{i,1}, \ldots, x_{i,N_i}$, $i = 1, \ldots, s$. N_i is the number of elements in the $i-$th cluster. To use the DP on the $i-$cluster, each element is replaced by

$$\hat{x}_{i,j} = \frac{x_{i,1} + \cdots + x_{i,N_i}}{N_i} + \frac{\epsilon_i}{N_i}, \quad i = 1, \ldots, s; j = 1, \ldots, N_i. \tag{3.5}$$

Laplace noise ϵ_i is generated from $Lap(\frac{S_Q}{\alpha})$. Here we follow the notations used in Xiao Xiao et al. (2012). S_Q is the sensitivity of a query and α is a parameter controlling the strength of privacy protection. α usually takes a value smaller than 1 and the smaller α leads to stronger protection.

In the following applications, we adopt k-means clustering as the clustering method and use different k values ranging from 10 to 150. The sensitivity of this problem is 2 so S_Q equals 2. And we choose 0.001 as the value for α. Adding DP with more clusters results in closer group mean values to the original Distributed NCE without DP. But at the same time, more clusters introduce more noise into the model, since creating s clusters means adding noise s times. Thus choosing the number of clusters is a trade-off between value accuracy and noise.

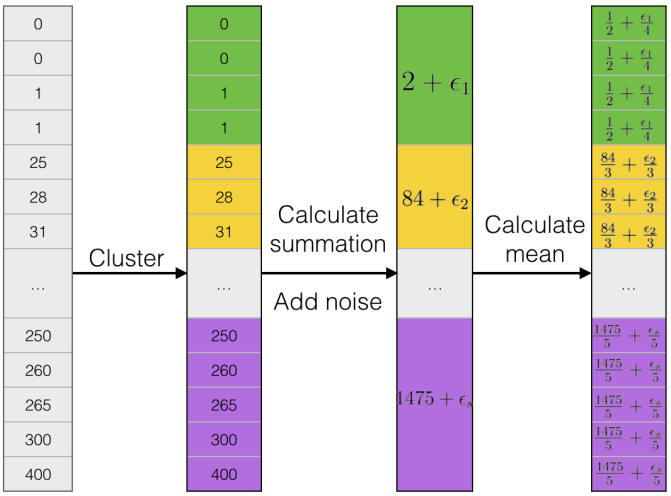


Figure 3.3: A symbolic illustration of implementing differential privacy on one-dimensional data. To apply DP on one-dimensional vectors, cluster counts to subgroups in the first step. For each subgroup, calculate summations and add noise to group summations. Last, average summations to individual cells.

3.3 Two alternative solutions

This section describes two additional methods as alternatives to Distributed NCE. They are applied in the following simulation experiments for comparison. The first one, called "naive updates", directly expands vocabulary when updating an existing Word2Vec model with a new dataset. The second method, called "dropout updates", is inspired by the dropout technique commonly used in neural network to avoid overfitting.

3.3.1 Naive updates

Suppose a Word2Vec model M_1 trained by the first dataset D_1 (see Section 3.2.1) has already been obtained, it can be updated by expanding the vocabulary of M_1 and adding input nodes and output nodes to the existing neural network. In the expanded model, denoted by M_2, the weights inherited from M_1 are initialized by the existing values in M_1 and new weights are randomly initialized. Then we train M_2 with new data D_2. This method has similar idea to an existing package called

"Online Word2Vec" Mulkar-Mehta (2015a).

Assume the input layer, hidden layer, and output layer of M_1 have nodes $\{x_1, x_2, \ldots, x_{N_1}\}$, $\{h_1, h_2, \ldots, h_L\}$, and $\{y_1, y_2, \ldots, y_{N_1}\}$ respectively. N_1 is the number of words in vocabulary of D_1 and L is the number of nodes in hidden layer. Note that only one input node among $x_1, x_2, \ldots, x_{N_1}$ is non-zero for SG model in each training iteration. By applying naive updates, it expands the existing input layer and output layer to $\{x_1, x_2, \ldots, x_{N_1}, x_{N_1+1}, \ldots, x_{N_1+K}\}$ and $\{y_1, y_2, \ldots, y_{N_1}, y_{N_1+1}, \ldots, y_{N_1+K}\}$ respectively. K is the length of new words in D_2 compared with D_1. This step is demonstrated in Figure 3.4. Then the expanded model M_2 is trained with D_2.

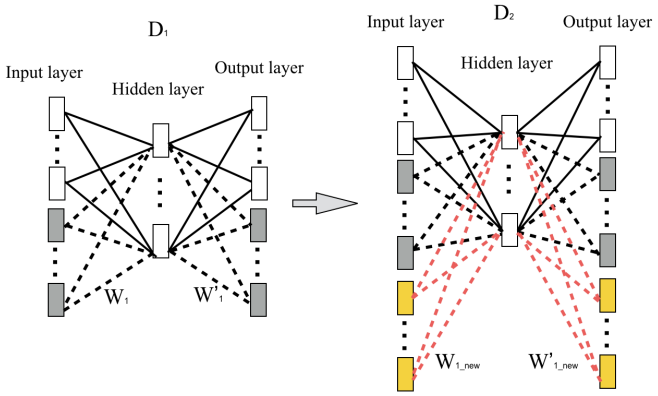


Figure 3.4: Naive updates. Left figure represents M_1 and right figure represents M_2. Empty squares represent the words exclusive to D_1. Gray squares are the words shared by both D_1 and D_2. Yellow squares are the words exclusive to D_2.

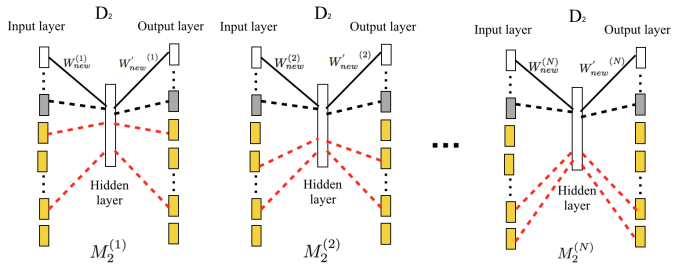


Figure 3.5: Dropout updates. This figure only demonstrate the second step in dropout updates, which is to update existing model using new dataset D_2. Empty squares represent the words exclusive to D_1. Gray squares are the words shared by both D_1 and D_2. Yellow squares are the words exclusive to D_2. When a node is not connected to hidden layer, it means this node is 'dropped' for the current training cycle.

3.3.2 Dropout updates

Following the same notation in section 3.3.1, dropout updates expand the input layer and output layer by a subset of new nodes. If we specify the proportion of random sample selected from new nodes as Q, each time $[Q \times K]$ new nodes $\{x_{(N_1+1)}, \ldots, x_{(N_1+[Q\times K])}\}$ and $\{y_{(N_1+1)}, \ldots, y_{(N_1+[Q\times K])}\}$ are selected from $\{x_{N_1+1}, \ldots, x_{N_1+K}\}$ and $\{y_{N_1+1}, \ldots, y_{N_1+K}\}$ respectively. $[\cdot]$ is the floor operation. $(\cdot)$ means it is a random sample. After adding these nodes to existing model M_1, the new model $M_2^{(i)}$ is trained by new data D_2. We repeat this step N times and obtain N input weights $W_{new}^{(i)}, i = 1, \ldots, N$. From another perspective, this process is the same as randomly dropping a proportion of nodes in the updated model M_2 in section 3.3.1. Since it has the similar flavor to the dropout technique usually used in neural network training to reduce overfitting, we call this model "dropout updates".

We use Figure 3.5 to demonstrate how existing models are updated through dropout method in N iterations. $W_{new}^{(1)}, W_{new}^{(2)}, \ldots, W_{new}^{(N)}$ are the weights of updated models $M_2^{(1)}, M_2^{(2)}, \ldots, M_2^{(N)}$. We calculate the final vector representation V_i for word ω_i by

$$V_i = \frac{\sum_{\omega_i \in Vocab^{(j)}} W_{new,i}^{(j)}}{\sum_{j=1}^{N} \mathbb{1}(\omega_i \in Vocab^{(j)})}. \tag{3.6}$$

$Vocab^{(j)}$ is the vocabulary selected in j-th iteration, $j = 1, \ldots, N$. $W_{new,i}^{(j)}$ is the $i-$th row of weight matrix $W_{new}^{(j)}$, $i = 1, \ldots, N_1 + K$.

3.4 Numerical study with real data

We conduct a series of simulation experiments using a real dataset. We only consider the situation when exactly two datasets are involved to build a global model. The results can be extended to multiple datasets scenarios by iteratively applying such one

versus one combination method. We consider the scenarios when the first database is smaller than, equal to, or larger than the second database. For each setting, we separate the whole data to equal or unequal subsets to mimic real life when two datasets are available locally. We use the model trained with the whole data as the gold standard to compare with the proposed methods.

3.4.1 Data and Data preprocess

MIMIC-III (Medical Information Mart for Intensive Care III) is a freely available database consisted of de-identified clinical data including more than 40000 patients Johnson et al. (2016). All the patients have received treatments from the intensive care units (ICU) of Beth Israel Deaconess Medical Center between 2001 and 2012. The MIMIC-III dataset contains a variety of measurements such as laboratory test results, prescriptions, symptoms, and other clinical measurements.

After MIMIC-III data are obtained, they are pre-processed into temporal sequences so that they can be accepted by the proposed models. Data pre-processing step has been described in detail by Farhan Farhan et al. (2016+) and we briefly summarize it here. For each subject in the database, we concatenate the medical events from multiple hospital admissions and sort the sequence by time. Using such procedures, temporal information of medical events can be preserved. In addition, different prefixes are added to events so that the code from different categories do not duplicate. For instance, '$p\_$', '$l\_$', '$s\_$', '$c\_$', and '$d\_$' are added at the beginning of corresponding terms to represent prescriptions, lab test keys, symptoms, conditions, and diagnoses. We save the latest diagnosis for all patients as the ground truth of prediction and exclude them from training data set. To ensure all patients have enough record for prediction, only those with multiple hospital admissions are kept in the final dataset. After the above preprocessing steps, a dataset including the temporal sequences of $5,642$ patients is obtained.

3.4.2 Settings

In the numerical study, the MIMIC III data are manually divided into different proportions to mimic two locally-available datasets. First, the whole dataset is randomly divided into subsets containing ninety percent versus ten percent of data, and the former part is used as the training set and the latter part as the testing set. Second, the training set (ninety percent of total data) is further divided into 10% plus 80%, 20% plus 70%, 30% plus 60%, 45% plus 45%, 60% plus 30%, 70% plus 20%, 80% plus 10%. Data ordering is kept during the process of dividing thus 20% plus 70% is not equivalent to 70% plus 20%.

In another setting, age is used to divide population into subgroups, in order to evaluate the performance of the proposed methods on different populations. In the first case, the whole patient body is divided into two groups using a probabilistic age-based cutoff, which is generated by $Logit(Pr(S = 1)) = 1 + b_1 \times Age$. Here $S = 1$ means subject is assigned to Group_2. This case is designed to mimic real situations with heterogeneous populations from multiple sites. In addition, we consider an extreme case where the data are divided to two groups using a strict age cutoff (53, 66, or 77 years old). This case, while unrealistic, is designed to assess robustness of our methods. As in the previous setting, 10% of data is randomly chosen as testing data and selection of testing data is not correlated with age.

We define the global model as the model trained with all the training data (90% of data) and the global model is used as the gold standard which the proposed methods are compared with. It is worth mentioning that the algorithm for solving neural networks has different levels of randomness with different settings of parameters. We discuss more about parameter selection in later sections.

We use two criteria to evaluate the performance of the proposed models. The first criterion is called Area Under Curve (AUC). For each disease, the true positive rate is plotted versus the false positive rate and calculate the area under this classification

curve. A good classifier should have true positive rate increase quickly and its AUC should be close to 1. Since there are more than seventy diagnoses, we calculate the average of all the AUCs as the final evaluation, Avg-AUC. Avg-AUC reflects the prediction accuracy of proposed models.

The second criterion is called Precision Top K (PTK). Given two sets of vector representations with the same vocabulary, for each word, we calculate the proportion of overlaps between the K most-similar words using the first set and the second set of vector representations. The similarity between vectors is measured by cosine similarity Steinbach et al. (2000). We repeat this procedure over all the words in vocabulary and take average of the calculated proportions as PTK. PTK reflects the similarity between two sets of vector representations. For example, if the top 3 most similar words of one diagnosis "d_24435" are lab test "l_104", prescription "p_28390", and symptom "s_335" using one model, and the top 3 most similar words using another model for the same diagnosis are "p_28390'', "l_104'' and condition "c_9002", we notice the overlaps between these two sets of results are "p_28390" and "l_104". Thus the PTK (K=3) in this example is 0.667 if the first set of vector representations is obtained from the gold standard model. We only take K=3 as an example, in the analysis, K=10 is used as the default value and K ranges from 1 to 500. In the calculation of PTK, the gold standard model is defined as the global model trained using one worker with all the training data (90% of data).

In this numerical study, Distributed NCE and its two alternatives are firstly evaluated in all scenarios and report PTK and AUC. Better methods should have higher values on both criteria. In the second step, the results of Distributed NCE and Distributed NCE with DP using different parameter values are compared.

3.4.3 Tuning parameters

Like other deep learning models (or statistical models), the proposed methods and the algorithm used to solve neural networks involve a number of parameters that are specified beforehand. This section briefly discusses the functions of these parameters and their impact on the randomness and efficiency of constructed models.

Learning rate is a hidden parameter inside the learning process of neural network. The magnitude of learning rate decides how large the "step size" is for each update step. The default learning rate automatically decreases proportionally from a max value (0.025) to a mim value (0.001) during the training process of one dataset. This is problematic when one want to train the model sequentially and obtain a global model. In Distributed NCE, we are able to adjust this parameter since both data sizes can be obtained when collecting vocabularies from different sites. The learning rate decreases proportional to data size as the way used in global model. On the contrary, the learning rates in naive updates and dropout updates cannot be adjusted since the size of the second data set is usually unknown when applying these two methods.

Iteration is a parameter controls the number of times the model is trained iteratively over the entire data. Default value for iteration used in Word2Vec is 5. More iterations result in more stable results but too many iterations would also dramatically increase computing time. Figure S3 demonstrates the PTK performance of Distributed NCE using different data partitions and various iterations. When number of iterations increases, the performance of unequal partitions stabilizes but the performance of equal partitions decreases. This happens because when applying Distributed NCE, model is trained with the first dataset repeatedly for N_{iter} times and then trained with the second dataset repeatedly for N_{iter} times. But when applying global model, model is trained with the whole dataset repeatedly for N_{iter} times. Unequal patitions tend to have one large dataset which dominates the training process and thus have stabler performance than equal partitions.

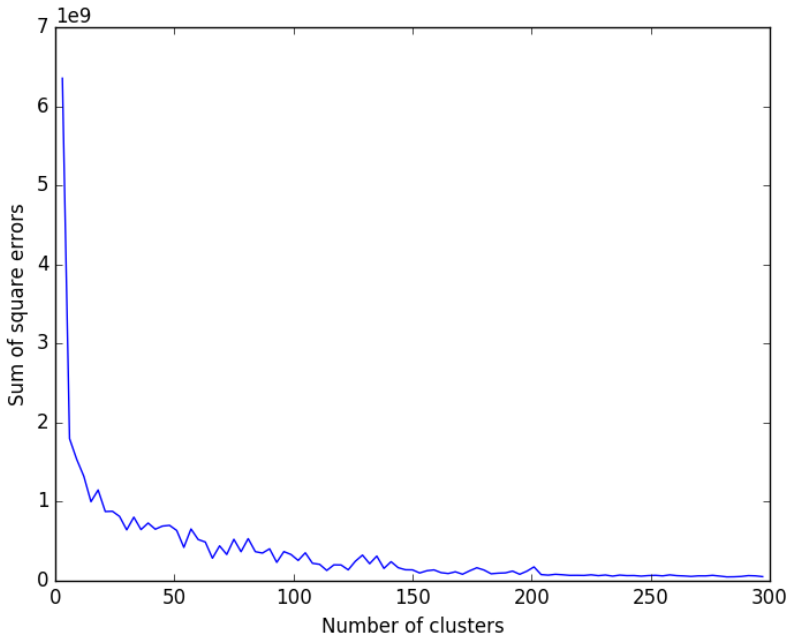


Figure 3.6: Sum of Squared Errors (with noise-added centroids) by Number of Clusters.

Maximum number of words controls the amount of words fed to the Word2Vec model in each iteration. Large maximum number means large training batches but also large learning rate jumps, which may result in poor estimation. Small maximum number makes learning process slow and training batches more susceptible to data partition. The default value is 10 000.

Number of workers : Our algorithm can use parallel computation to speed up the training process and the default number of workers is 20. Figure **??** shows the PTK change of two repetitive global models versus number of workers. Generally, more workers increase computing speed but also add randomness to final results. When number of workers equals one, two repetitive global models have the same set of results thus PTK equals one. PTK value decreases when number of workers increases and the reduction becomes minimum after more than ten workers are used. To reduce randomness in results, one worker is used in the numerical study.

As discussed, multiple workers, change of sentence ordering, and a small num-

ber of iterations all contribute to model randomness. Multiple workers, change of feeding batches resulted from data partition, and more iterations contribute to the performance gap between Distributed NCE and the gold standard model. Parameter selection is a trade-off between computing time, stability, and convergence. We believe the parameter set used here (iteration = 5, max number of words = 10 000, workers = 1 for stable results or 20 for faster speed) have achieved a good balance.

3.4.4 Results

Table 3.1 presents PTK and AUC of three proposed methods. To emphasize the main findings, only three settings are demonstrated: 45% and 45% of all data as training data plus 10% as testing data, 10% and 80% as training data plus 10% as testing data, 80% and 10% as training data plus 10% as testing data. These three scenarios represent three extreme conditions: two training sets are of equal sizes; the first set is much smaller than the second set; the first set is much larger than the second set. In this experiment, 1 worker and 5 iterations are used to reduce result randomness. The default parameter selection of the original Word2Vec program is 20 workers and 5 iterations, which can greatly improve computing speed but introduce some result randomness. A complete table of results can be found in Table S2, where the separation of training set ranges across 10:80, 20:70, 30:60, 45:45, 60:30, 70:20, and 80:10.

Table 3.1 shows that Distributed NCE has the best performance among the proposed methods, especially comparing the measurement PTK. Dropout updates perform worst. Partition of data heavily influences on the performance of Dropout updates.

Figure S1 re-confirms the findings from Table 3.1. In this figure, PTK of three methods is ploted against a wide range of K selection under three scenarios. Although the performance of all methods fluctuates across three settings, it is consistent that

Distributed NCE always has the highest PTK values. Again, the same parameter selection is used as mentioned above (workers = 1, iterations = 5) during this experiment.

Both Table 3.1 and Figure S1 indicate that Distributed NCE outperforms other proposed methods with regard to the evaluation criterion PTK. As mentioned in Section 3.2.4, privacy issue is a potential concern for the current Distributed NCE, thus Distributed NCE with DP is proposed to provide better privacy protection. Table S1 demonstrates Precision-Top-K of Distributed NCE with DP using different numbers of clusters comparing with Distributed NCE without DP. As mentioned above, selection of K is a trade-off between cluster mean accuracy and added noise. Table S1 indicates that PTK of Distributed NCE with DP is closest to Distributed NCE when number of clusters is around 30 to 50. And generally speaking, adding privacy protection does not decrease PTK greatly.

	45 : 45 : 10		10 : 80 : 10		80 : 10 : 10	
	PTK	$Avg-AUC$	PTK	$Avg-AUC$	PTK	$Avg-AUC$
Naive updates	0.52 (2e-3)	0.77 (8e-3)	0.50 (2e-3)	0.77 (8e-3)	0.49 (3e-3)	0.78 (8e-3)
Dropout updates	0.22 (3e-3)	0.72 (7e-3)	0.13 (9e-4)	0.72 (5e-3)	0.37 (4e-3)	0.73 (7e-3)
Distributed NCE	0.58 (2e-3)	0.77 (8e-3)	0.64 (2e-3)	0.77 (8e-3)	0.65 (3e-3)	0.77 (7e-3)

Table 3.1: Results of all methods using $Skip-Gram$ model. Results are summarized over 10-folds cross validation. Distributed NCE is Distributed Noise Contrastive Estimation. PTK is Precision-Top-K. K equal 10 in all experiments. Avg-AUC is averaged Area-Under-Curve. 45 : 45 : 10 means that the two training datasets are 45% and 45% of total data. Testing dataset is 10% of total data. Global model uses all 90% data as training data.

To decide the most appropriate selection for cluster number, we also plot the sum of squared errors using noise-added centroids by number of clusters. This type of plot is usually used to identify the optimal number of clusters in k-means clustering. Consistently, the elbow place of Figure S2 is around 30 to 50. Thirty clusters may be an optimal number of clusters according to both Table S1 and Figure S2.

Table S3 and Table S4 show the simulation results with age-correlated subsets. In Table S3, $b_1 = -0.04$ has uneven population separation than $b_1 = -0.02$ and

$b_1 = -0.002$. Both Naive updates and Dropout updates have worse results when subset differences are larger. But Distributed NCE and Distributed NCE with DP not only have best PTK and AUC, but also perform stably for all three separations. In Table S4, Distributed NCE and Distributed NCE with DP have the best performance for all three age cutoffs. Although Table S4 shows an extreme case of population separation, it demonstrates the robustness of our method.

3.5 Conclusion and Discussion

In this work, we propose and investigate several methods to extend current neural network based predictive models for medical events. The proposed methods allow researchers to build predictive models using multiple EHR datasets sequentially and distributedly, avoiding the potential inconvenience of sharing EHR data.

To validate an established model, a few downstream analysis can be performed, including grouping medical concepts from different institutions, finding similar patients by constructing patient profiles from observations, and making predictions based on records, etc.. For example, biomedical ontologies is increasing in the context of health system interoperability, which are the keys to understanding the semantics of information exchange Schulz and Martínez-Costa (2013). The diversity of biomedical ontologies call for advanced tools to harmonize them and the ability to find similar concept without exchanging raw data is highly appreciated. Our model can be evaluated when two similar concepts (in a global sense) are presented in a distributed setting (e.g., appearing in different sources). We can test their similarity using our proposed method against the baseline approach (concepts trained in a centralized manner) to see how well the semantics are preserved. Such evaluation can be extended to search similar patients (based on profiles synthesized by distributed embedding of their corresponding concepts). In this sense, we would expect similar patients (in a global setting) remain similar after the distributed training approach is adopted.

It should be clarified that this work is constructed on the 'old' EHR system and on standardized clinical data. The 'old' EHR systems use ICD-10 for medical event prototyping in contrast to the modern systems which support medical event prototyping through both ICD-10 and e-prescription. In addition, the latest systems use different standardization, which is a challenge to be tackled. And difficulties also exist in data harmonization, especially when different data sources are highly heterogeneous in terms of format. From raw EHR data to standardized EHR data, we envision Common Data Models (CDM) such as Observational Medical Outcomes Partnership (OMOP) Stang et al. (2010) will bridge the gap. In addition, since data harmonization is not a unique issue in the distributed analysis and is needed whenever people try to use multiple data sources, ongoing efforts exist to overcome such hurdles.

Our work has some limitations. First, the proposed methods are evaluated based on one data source instead of multiple sources. In practice, the data from different hospitals have heterogeneity in terms of coding system, clinical standards, patient profiles, and etc., which are not addressed in this work. Technically, the proposed models can handle datasets with different levels of differences. Based on our experience, utilizing information from two hospitals has disadvantages when large discrepancy exists in two coding systems or patient profiles. But merging information can be especially beneficial if two hospitals have similar patient populations but not enough size on their own or the information from both hospitals are complementary. Such data usually exist in distributed medical data sets of clinical data research network (CDRN). Compared with two different hospitals, CDRNs use the same or similar coding systems and clinical standards. And the patient profiles are more homogeneous. Our proposed method will be a good fit to datasets in such distributed data networks. Of note, one of our numerical experiments is designed to mimic hetergeous data sources.

Another limitation is that, although the proposed methods can learn predictive model sequentially and distributedly, the learning process is not completely indepen-

dent among datasites. Using the proposed methods, the second datasite need to wait until the first datasite finishes learning, which may not be efficient enough in real life. To learn a global model parallelly, one could obtain vector from separate datasites and conduct downstream combining, which may be a more complicated problem than the current situation. Also, one can consider combining Bayesian ideas with neural network and imposing different prior probability on nodes when updating neural networks.

For future research, the proposed models should be further evaluated using data from different sources, including data from different hospitals and data from distributed medical systems. Also, it is desirable to develop model construction technique which can learn model structure completely parallel.

Chapter 4

Bayesian Generalized Biclustering Analysis via Adaptive Structured Shrinkage

4.1 Introduction

Advances in high-throughput technologies have enabled researchers to uncover secrets of human genome on various levels. From microarray to next-generation sequencing, these tools can reveal understandings of genomic activity including DNA composition, abundance of transcriptome, epigenetic modification, etc.. In recent years, there have been growing interests on integrative analysis of data from multiple genomic modalities for identifying disease subtypes Verhaak et al. (2010), inferring omics network Ideker et al. (2001); Tanay et al. (2004), and uncovering disease culprit genes Network et al. (2011). One significant challenge in integrating multiple genomic data sources is that these data have different characteristics and are usually difficult to be unified and explored by one single method. Although multiple attempts have been made, more analytical techniques are needed to fully realize the potential of existing vast omics data.

Biclustering is a popular unsupervised learning and data mining technique which can identify local patterns of a data matrix by clustering feature space and sample space at the same time. The idea of biclustering was first discussed by Hartigan (1972) using the term "direct clustering". Biclustering of gene expression microarray data was first formally introduced by Cheng and Church (2000). Since then, various biclustering methods have been proposed and successfully applied to the analysis of gene expression data Hochreiter et al. (2010); Lazzeroni and Owen (2002); Sheng et al. (2003); Ben-Dor et al. (2003); Gu and Liu (2008); Caldas and Kaski (2008); Bergmann et al. (2003); Murali and Kasif (2002); Yu et al. (2017); Liu et al. (2014); Huda and Noureen (2016). Biclustering methods have been systematically compared in several review papers Prelić et al. (2006); Pontes et al. (2015); Eren et al. (2012); Padilha and Campello (2017).

Following the review paper by Padilha and Campello (2017), the existing bi-

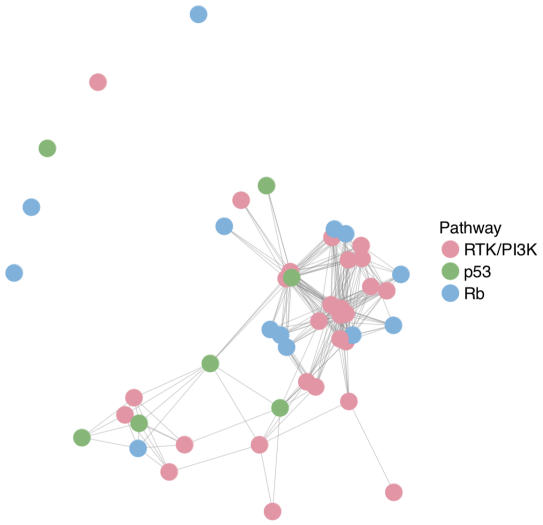


Figure 4.1: Interactions of 48 genes that overlap with the three critical signaling pathways - RTK/PI3K, p53, and Rb, which closely relate with migration, survival and apoptosis progression of cell cycles. This gene network information is extracted from the KEGG pathway and is utilized in the integrative analysis by the proposed method (section 4.4).

clustering methods can be categorized as greedy algorithms, divide-and-conquer algorithms, exhaustive enumeration algorithms and distribution parameter identification algorithms. To be specific, greedy algorithms including CC Cheng and Church (2000), xMotifs Murali and Kasif (2002), ISA Bergmann et al. (2003), etc.; divide-and-conquer algorithms include Bimax Prelić et al. (2006)) and MTBGD Huda and Noureen (2016); exhaustive enumeration algorithms include SAMBA Tanay et al. (2002), BiBit Rodriguez-Baena et al. (2011), and DeBi Serin and Vingron (2011)); distribution parameter identification algorithms include Plaid Caldas and Kaski (2008), FABIA Hochreiter et al. (2010), etc.. Among all, FABIA is of particular interest to us, as it closely relates to our model formulation. FABIA uses a multiplicative model and imposes Laplace priors on latent variables. Both Hochreiter et al. (2010) and Padilha and Campello (2017) show that FABIA achieves robust performance in their simulation studies and real data applications.

Although many biclustering approaches have been developed, few of them can uti-

lize existing biological information for identifying biclustering patterns such as those from functional genomics or proteomics. An example of such biological information is demonstrated in Figure (4.1). Such gene network can be obtained from publicly available databases including KEGG pathway Kanehisa and Goto (2000); Mi et al. (2015); Keshava Prasad et al. (2008). In addition, recent work has shown that incorporating biological information can improve variable selection and prediction performance in methods such as linear regression and multivariate analysis Chang et al. (2016); Li et al. (2017); Safo et al. (2017); Zhao et al. (2016); Li and Li (2008a). Furthermore, most, if not all, existing biclustering methods focus on analyzing gene expression microarray data which are of continuous data type. Our simulation results have shown that the current methods cannot identify biclusters with good accuracy on inputs of mixed data types, for example, data generated from Gaussian distribution and Binomial distribution. To address this challenge, we develop a more generalized approach to identify the biclustering patterns using one or multiple genomic datasets. Our work takes advantage of recent work by Polson et al. (2013), which developed a unified Bayesian inference framework for analysis of data from exponential family distributions through the use of Pólya-Gamma latent variables. Polson tranforms common discrete data distributions into a Gaussian distribution framework by introducing auxilary variables. By combining Pólya-Gamma latent variables with a multiplicative modeling framework, we formulate a Bayesian biclustering model similar in spirit with Hochreiter et al. (2010) but can accept different data types as inputs. In addition, our approach allows the incorporation of prior biological knowledge in the process of biclustering, if such biological information exists. We call this approach Generalized Biclustering (GBC).

Our contributions are summarized as follows: **(1)** We propose **a noval Bayesian biclustering method** GBC to simultaneously cluster feature space and sample space of -omics data while **incorporating prior biological information** during the pro-

cess; **(2)** We present the likelihood functions of different data types in **a unified framework** and employ a novel Bayesian adaptive shrinkage priors to yield sparse solutions; **(3)** We design **an efficient EM algorithm** to solve the biclustering problem and make our program available on GitHub[1]; **(4)** We assess the proposed methods in comparison with several existing biclustering techniques in **a series of simulation studies and analyses of multiple real datasets**. The proposed methods achieve the best or close to the best performance in our numerical experiments.

The structure of this paper is as follows. Section 4.2 introduces our model formulation including the adaptive structured prior and the computation of GBC for different data types. Section 4.3 presents the simulations comparing the proposed method with other popular biclustering methods. Section 4.4 presents the applications on real datasets.

4.2 Methodology

To fix ideas, suppose we have a random sample of n subjects for which data are obtained from H genomic platforms, such as microarray and next-generation sequencing, denoted by $\mathbf{X}_1, \ldots, \mathbf{X}_H$. Each of them is a $p_h \times n$ matrix, $1 \leq h \leq H$, where p_h is the number of features and n is the number of samples. Let $\mathbf{X}$ be their vertical concatenation with size $p \times n$, $\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_H \end{bmatrix}$, where $p = \sum_{h=1}^{H} p_h$. It follows that the rows represent the feature space and the columns represent the sample space. Let $\boldsymbol{\mu}$ denote the mean of $\mathbf{X}$ and $\boldsymbol{\mu}$ is related with latent components through $\boldsymbol{\mu} = \mathbf{m} + \mathbf{WZ}$ where $\mathbf{m}$ is a location vector, $\mathbf{W}$ is a $p \times L$ factor loading matrix and $\mathbf{Z}$ is a $L \times n$ latent factor matrix. To understand this model formulation, one may make an analogy between

[1] https://github.com/ziyili20/GBC

this framework and the generalized linear model $\boldsymbol{\mu} = g^{-1}(\mathbf{Z}\beta)$ with observations $\mathbf{X}$ and covariates $\mathbf{Z}$. $\boldsymbol{\mu}$ in both models are the means of observations. Assuming the observations x_{ij}'s are independent one from each other, the likelihood of observations $\mathbf{X}$ is the multiplication of the likelihood of each individual observation and $\boldsymbol{\mu}$ is the parameter of the likelihood function, $\pi(\mathbf{X}|\boldsymbol{\mu}) = \prod_j \prod_i \pi_j(x_{ji}|\mu_{ji})$. In the remaining of Section 2, we only consider π_j to be an exponential family likelihood for the random variable x_j.

Using the above notations, a number of distributions can be considered to model observed variables. For instance, if the observation $\mathbf{X}$ is continous and of bell shaped curve, one can assume x_{ji} follows the Gaussian distribution of mean μ_{ji} and precision ρ_j with density function as

$$\pi_j(x_{ji}|\mu_{ji}, \rho_j) = \frac{\rho_j^{1/2}}{\sqrt{2\pi}} e^{-\rho_j(x_{ji}-\mu_{ji})^2/2}. \tag{4.1}$$

If the observation $\mathbf{X}$ is discrete and one can assume that x_{ji} follows a Binomial distribution with parameter n_j and p_{ji}. Using the logit link function, the likelihood function is

$$\begin{aligned}\pi_j(x_{ji}|\mu_{ji}, n_j) &= \binom{n_j}{x_{ji}} p_{ji}^{x_{ji}}(1-p_{ji})^{n-x_{ji}} \\ &= \binom{n_j}{x_{ji}} \frac{e^{\mu_{ji}x_{ji}}}{(1+e^{\mu_{ji}})^{n_j}}, x_{ji} = 0, 1, \ldots, n_j. \end{aligned} \tag{4.2}$$

If assuming x_{ji} follows Negative Binomial with r_j and p_{ji} and again using the logit link function for p_{ji}, the likelihood is given by

$$\begin{aligned}\pi_j(x_{ji}|\mu_{ji}, r_j) &= \binom{r_j + x_{ji} - 1}{x_{ji}} p_{ji}^{x_{ji}}(1-p_{ji})^{r_j} \\ &= \binom{r_j + x_{ji} - 1}{x_{ji}} \frac{e^{\mu_{ji}x_{ji}}}{(1+e^{\mu_{ji}})^{r_j+x_{ji}}}, x_{ji} = 0, 1, 2, \ldots. \end{aligned} \tag{4.3}$$

Lastly, if assuming x_{ji} follows Poisson distribution with parameter $e^{\mu_{ji}}$, the likelihood can be approximated with large N and small $p_{ji} = N^{-1}e^{\mu_{ji}}/(1+N^{-1}e^{\mu_{ji}})$ in the form of Binomial distribution. It follows that the likelihood is given by

$$\begin{aligned}\pi_j(x_{ji}|\mu_{ji}) &= e^{-e^{\mu_{ji}}}\frac{e^{\mu_{ji}x_{ji}}}{x_{ji}!} \\ &\approx \binom{N}{x_{ji}}\frac{N^{-x_{ji}}e^{\mu_{ji}x_{ji}}}{\left(1+\frac{1}{N}e^{\mu_{ji}}\right)^N}, x_{ji} = 0, 1, \ldots, N. \end{aligned} \tag{4.4}$$

In the following derivations, we take the above four distributions - Gaussian, Binomial, Negative-Binomial, and Poisson - as examples to illustrate the proposed method. Other exponential family distribution such as Bernoulli, Log-normal can be handled similarly.

4.2.1 Prior Specification

We employ a Bayesian adaptive structured shrinkage prior formulation similar to Chang et al. (2016) and the goal is to achieve sparse estimations for $\mathbf{W}$ and $\mathbf{Z}$ while incorporating existing biological information simultaneously. There are multiple components in this prior. First, a Bayesian Laplacian shrinkage prior is imposed on $\mathbf{W}$:

$$\log \pi(\mathbf{W}|\boldsymbol{\lambda}) = C + \sum_{j,l}\log\lambda_{jl} - \sum_{j,l}\lambda_{jl}|w_{jl}|$$

where λ_{jl} is a parameter controlling the shrinkage level of w_{jl}. Unlike standard Laplacian prior that uses the same shrinkage parameter λ for all w_{jl}'s, our approach adapts the shrinkage parameter to individual w_{jl}, hence the term of adaptive shrinkage. We further impose a Bayesian shrinkage prior on $\boldsymbol{\lambda}$ to incorporate biological information, also known as structural information, hence the term of structured shrinkage prior.

Suppose the biological information is given through graphs. H graphs $\mathcal{G}_h = \langle P_h, E_h\rangle$ are given where P_h is the set of variables $1, \ldots, p_h$ in the h-th dataset and

E_h is the set of edges between pairs of variables. The presence of edges represents the correlations of corresponding variable pairs are nonzero. We combine these H graphs into a single graph $\mathcal{G} = \langle P, E \rangle$ by setting $P = 1, \ldots, p$ and $E = \{(\iota(h,j), \iota(h,k)) : (j,k) \in E_h, 1 \leq h \leq H\}$ where $\iota(h,j)$ is the index in the matrix X of the j-th variable in the h-th dataset. Intuitively, when two variables are connected by edges, we encourage these two variables to share similar factors. One way to achieve such effects is to encourage one variable to load on a factor if the other variable has non-zero loading on the same factor. Translating this to notations shows that, if x_j and x_k are connected in $\mathcal{G}$ and w_{jl} is non-zero for some l, w_{kl} should also be encouraged to have non-zero values. To this end, we employ a graph-Laplacian prior for $\boldsymbol{\lambda}$ given the precision matrix $\boldsymbol{\Omega}$ as:

$$\log \pi(\boldsymbol{\alpha}|\boldsymbol{\Omega}) = C_{\nu_2} + \frac{L}{2} \log |\boldsymbol{\Omega}| - \frac{1}{2\nu_2} \sum_l (\boldsymbol{\alpha}_l - \nu_1 \underline{1}) \boldsymbol{\Omega} (\boldsymbol{\alpha}_l - \nu_1 \underline{1}), \tag{4.5}$$

where $\alpha_{jl} = \log \lambda_{jl}$ and $\boldsymbol{\alpha}_l = (\alpha_{1l}, \ldots, \alpha_{pl})'$ for $1 \leq l \leq L$. ν_1 and ν_2 are hyper-parameters needed to be specified a priori. The precision matrix $\boldsymbol{\Omega}$ is defined as

$$\boldsymbol{\Omega} = \begin{bmatrix} 1 + \sum_{j \neq 1} \omega_{1j} & -\omega_{12} & \cdots & -\omega_{1p} \\ -\omega_{21} & 1 + \sum_{j \neq 2} \omega_{2j} & \ddots & -\omega_{2p} \\ \vdots & \ddots & \ddots & \vdots \\ -\omega_{p1} & -\omega_{p2} & \cdots & 1 + \sum_{j \neq p} \omega_{pj} \end{bmatrix}.$$

Note that $\boldsymbol{\Omega}$ is a symmetric matrix, i.e., $\omega_{jk} = \omega_{kj}$. The following prior is assigned on set $\boldsymbol{\omega} = \{\omega_{jk} : j < k\}$

$$\pi(\boldsymbol{\omega}) \propto |\Omega|^{-L/2} \prod_{(j,k) \in E} \omega_{jk}^{a_\omega - 1} \exp(-b_\omega \omega_{jk}) 1(\omega_{jk} > 0) \prod_{(j,k) \neq E} \delta_0(\omega_{jk}). \tag{4.6}$$

$\delta_0(\cdot)$ is the Dirac delta function concentrated at 0 and $1(\cdot)$ is the indicator function. It can be shown that (4.6) is a propor prior Chang et al. (2016). Thus if x_j and x_k are connected in graph $\mathcal{G}$, the prior formula (4.6) encourages precision matrix components ω_{jk} to be non-zero and the shrinkage terms λ_{jl} and λ_{kl} are encouraged to be correlated through prior (4.5). Since w_{jl} and w_{kl} receive a similar level of shrinkage under this prior specification, they tend to be zero or nonzero at the same time. In other words, if genes j and k are connected in a pathway, they are encouraged to be selected together (or not selected together) in bicluster l. As such, a salient feature of our approach is that the selected feature set in each bi-cluster tends to include gene pathways rather than individual genes, leading to biologically more meaningful results.

In order to obtain sparse estimates for $\mathbf{Z}$, we also employ a Bayesian Laplacian shrinkage prior on Z as follows:

$$\log \pi(\mathbf{Z}|\boldsymbol{\xi}) = C + \sum_{l,i} \log \xi_{li} - \sum_{l,i} \xi_{li}|z_{li}|,$$

where $\xi_{li} > 0$ are the shrinkage parameters. Since no prior biological information is available for subjects, we impose a conjugate prior, i.e. a Gamma prior on $\boldsymbol{\xi}$ as

$$\log \pi(\boldsymbol{\xi}) = C_{\nu_3,\nu_4} + (\nu_3 - 1) \sum_{l,i} \log \xi_{il} - \frac{1}{\nu_4} \sum_{l,i} \xi_{li}, \tag{4.7}$$

where ν_3 and ν_4 need to be specified a priori.

4.2.2 Computation

As the likelihoods given in functions (4.1) to (4.4) are dissimilar with inputs of different data types, usually the computation procedures to optimize such likelihoods are also not the same. However, by introducting the Pólya-Gamma latent variables Polson et al. (2013), we are able to build a unified likelihood for inputs of different

data types. Such unified likelihood facilitates the following computations and allows the proposed method to have the flexibility in analyzing data from various sources. We use the identity formula provided in Polson et al. (2013):

$$\frac{e^{\mu_{ji}x_{ji}}}{(1+e^{\mu_{ji}})^{b_{ji}}} = 2^{-b_{ji}}e^{\kappa_{ji}\mu_{ji}}\int_0^\infty e^{-\rho_{ji}\mu_{ji}^2/2}\pi_{ji}(\rho_{ji})d\rho_{ji},$$

where $\kappa_{ji} = x_{ji} - b_{ji}/2$ and $\pi_{ji}(\rho_{ji})$ is the density of the Pólya-Gamma class $\mathcal{PG}(b_{ji}, 0)$. This approach transforms exponential family distribution to a Gaussian-distribution-like formula. Thus the likelihood functions (4.1) to (4.4) can be written in the following universal form:

$$\pi_j(\mathbf{x}_j|\mu_j) \propto e^{-\frac{1}{2}\sum_i \rho_{ji}(\mu_{ji}-\psi_{ji})^2 + \sum_i \kappa_{ji}\mu_{ji}}\pi_j^*(\rho_j), \tag{4.8}$$

where the unknown components are summarized in Table 4.1. Besides offering a unified likelihood function, the augmentation of ρ enables the use of efficient lasso algorithms for solving for $\mathbf{W}$ and $\mathbf{Z}$ in the M-steps of EM algorithms, which otherwise is not possible. In addition, the approach of Polson et al. (2013) also enables the use of Gibbs sampling in MCMC instead of Metropolis-Hasting, if MCMC was implemented.

Data type	ψ_{ji}	κ_{ji}	b_{ji}	$\pi_j^*(\rho_j)$
Gaussian	X_{ji}	0	NA	$\rho_{ji} \equiv \rho_j \sim \mathcal{G}\left(\frac{\zeta_j+n}{2}, \frac{\zeta_j}{2}\right)$
Binomial	0	$X_{ji} - n_j/2$	n_j	$\rho_{ji} \sim \mathcal{PG}(b_{ji}, 0)$
Neg Binomial	0	$(X_{ji} - r_j)/2$	$X_{ji} + r_j$	$\rho_{ji} \sim \mathcal{PG}(b_{ji}, 0)$
Poisson	$\log N$	$X_{ji} - N/2$	N	$\rho_{ji} \sim \mathcal{PG}(b_{ji}, 0)$

Table 4.1: Formula components of Pólya-Gamma classes

Similar to Hochreiter et al. (2010), we use expectation-maximization (EM) algorithm to compute maximum a posteriori (MAP) estimation of the likelihood function

(4.8). The MAP estimator $(\hat{\boldsymbol{W}}, \hat{\boldsymbol{Z}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\xi}})$ is defined as,

$$(\hat{\boldsymbol{W}}, \hat{\boldsymbol{Z}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\xi}}) = \underset{W,Z,\alpha,\xi}{\arg\max} \int \int \pi(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\Omega}|\boldsymbol{X}) d\boldsymbol{\rho} d\boldsymbol{\Omega},$$

with $\boldsymbol{\rho}, \boldsymbol{\Omega}$ marginalized out. Of note, our EM algorithm yields sparse soluations for W and Z. Although Markov chain Monte Carlo (MCMC) could also provide solutions, EM algorithm is more scalable to high dimension settings of our interest while a full MCMC can be very expensive. Moreover, it requires additional steps to define bicluster membership from MCMC solutions, which is further complicated by the fact that MCMC solutions do not have exact zeroes and hence may not be sparse. We adopt a recent computational technique called dynamic weighted lasso (DWL) Chang and Tsay (2010) in each EM iteration which further speeds up the algorithm.

EM Algorithm

The inputs of this algorithm include a p by n observed data matrix $\mathbf{X}$, a p element vector for data types, and a p element vector for specific parameter values of each data type. If prior biological information is available, edges between connected variables should also be provided. For Gaussian, Binomial, Negative Binomial and Poisson data, prior parameter for variance specification ζ_j (Gaussian), number of trials n_j(Binomial), number of failures r_j (Negative Binomial) and large number N (Poisson) should be specified. Definitions of these parameters are demonstrated in the likelihood functions (4.1) to (4.4).

We develop an EM algorithm for obtaining MAP. The optimization problem in the M step at the t-th iteration is defined as follows,

$$(\boldsymbol{W}^{(t)}, \boldsymbol{Z}^{(t)}, \boldsymbol{\alpha}^{(t)}, \boldsymbol{\xi}^{(t)}) = \underset{W,Z,\alpha,\xi}{\arg\max}\, \tilde{\mathbb{E}}_t \log \pi(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\Omega}, \boldsymbol{X}),$$

where the expectation $\tilde{\mathbb{E}}_t$ is taken with respect to

$\tilde{\pi}_t(\boldsymbol{\rho}, \boldsymbol{\Omega}) = \pi(\boldsymbol{\rho}, \boldsymbol{\Omega} | \boldsymbol{W}^{(t-1)}, \boldsymbol{Z}^{(t-1)}, \boldsymbol{\alpha}^{(t-1)}, \boldsymbol{\xi}^{(t-1)}, \boldsymbol{X})$. The objective function to be optimized at the t-th EM iteration step is given by

$$\begin{aligned}\mathbf{Q}_t(\mathbf{Z}, \mathbf{W}, \boldsymbol{m}, \boldsymbol{\alpha}, \boldsymbol{\xi}) = &-\frac{1}{2}\sum_{i,j}\rho_{ji}^{(t)}(\mu_{ji} - \psi_{ji})^2 + \sum_{i,j}\kappa_{ji}\mu_{ji} + \sum_{j,l}\alpha_{jl} \\ &- \sum_{j,l}\lambda_{jl}|w_{jl}| + \nu_3\sum_{l,i}\log\xi_{i,l} - \sum_{i,l}\xi_{l,i}(|z_{li}| + \frac{1}{\nu_4}) \\ &- \frac{1}{2\nu_2}\sum_{l}(\boldsymbol{\alpha_l} - \nu_1\mathbf{1})^T\boldsymbol{\Omega}^{(t)}(\boldsymbol{\alpha_l} - \nu_1\mathbf{1})\end{aligned}$$

where $\boldsymbol{\mu} = \boldsymbol{m} + \mathbf{WZ}$, $\rho_{ij}^{(t)} = \mathbb{E}(\rho_{ij}|\mathbf{X}, \mathbf{W}^{(t-1)}, \mathbf{Z}^{(t-1)}, \boldsymbol{m}^{(t-1)}, \boldsymbol{\alpha}^{(t-1)}, \boldsymbol{\xi}^{(t-1)})$, and $\boldsymbol{\Omega}^{(t)} = \mathbb{E}(\omega_{ij}|\mathbf{X}, \mathbf{W}^{(t-1)}, \mathbf{Z}^{(t-1)}, \boldsymbol{m}^{(t-1)}, \boldsymbol{\alpha}^{(t-1)}, \boldsymbol{\xi}^{(t-1)})$. The detailed steps of the EM algorithm are explained as follows.

E-step for $\boldsymbol{\rho}$: If the data type is Gaussian,

$$\widetilde{\mathbb{E}}(\rho_j) = \frac{\zeta_j + n}{\zeta_j + \sum_i(x_{ji} - \mu_{ji})^2}.$$

Otherwise,

$$\widetilde{\mathbb{E}}(\rho_{ji}) = \frac{b_{ji}(e^{\mu_{ji}} - e^{\psi_{ji}})}{2(\mu_{ji} - \psi_{ji})(e^{\mu_{ji}} + e^{\psi_{ji}})}.$$

E-step for $\boldsymbol{\omega_{jk}}$.

$$\widetilde{\mathbb{E}}(\omega_{jk}) = \frac{2\nu_2\alpha_\omega}{2\nu_2 b_\omega + \Sigma_l(\alpha_{jl} - \alpha_{kl})^2}, \quad j < k.$$

M-step for $\boldsymbol{Z}$: Solve the lasso problem

$$\mathbf{z}_{\cdot i} = \underset{\mathbf{z}}{\operatorname{argmin}}\left(\frac{1}{2}\mathbf{z}'\mathbf{W}'\widetilde{\mathbb{E}}(\mathbf{D}_i)\mathbf{W}\mathbf{z} - \mathbf{z}'\mathbf{W}'\widetilde{\mathbb{E}}(\mathbf{c}_i) + \sum_l \xi_{li}|z_l|\right),$$

where $i = 1, \ldots, n$, $\mathbf{D}_i = \operatorname{diag}(\rho_{1i}, \ldots, \rho_{pi})$ and $\mathbf{c}_i$ is the i-th column of the $p \times n$

matrix $\boldsymbol{\kappa} + \boldsymbol{\rho} \circ (\boldsymbol{\psi} - \mathbf{m}\underline{1}_n')$. Here, $\circ$ denotes the Hadamard product.

M-step for $\boldsymbol{W}$: Solve the lasso problem

$$\mathbf{w}_{j\cdot}' = \underset{\mathbf{w}}{\operatorname{argmin}} \left(\frac{1}{2} \mathbf{w}' \mathbf{Z} \widetilde{\mathbb{E}}(G_j) \mathbf{Z}' \mathbf{w} - \mathbf{w}' \mathbf{Z} \widetilde{\mathbb{E}}(\mathbf{f}_j) + \sum_l \lambda_{jl} |w_l| \right),$$

where $j = 1, \ldots, p$, $\mathbf{G}_j = \operatorname{diag}(\rho_{j1}, \ldots, \rho_{jn})$, and $\mathbf{f}_j'$ is the j-th row of $\boldsymbol{\kappa} + \boldsymbol{\rho} \circ (\boldsymbol{\psi} - \mathbf{m}\underline{1}_n')$.

M-step for $\mathbf{m}$:

$$m_j = \frac{-\mathbf{w}_j' \mathbf{Z} \boldsymbol{\rho}_j + \boldsymbol{\psi}_j' \boldsymbol{\rho}_j + \boldsymbol{\kappa}_j' \mathbf{1}}{\boldsymbol{\rho}_j' \mathbf{1}}, \quad j = 1, \ldots, p.$$

M-step for $\boldsymbol{\alpha_l}$:

$$\boldsymbol{\alpha}_l^{(new)} = \boldsymbol{\alpha}_l^{(old)} - \mathbf{H}_l^{-1} \mathbf{g}_l.$$

Here $\mathbf{H}_l = -diag(e^{\boldsymbol{\alpha}_l |w_l|}) - \frac{\boldsymbol{\Omega}}{\nu_2}$ and $\mathbf{g}_l = \mathbf{1}_{p\times 1} - e^{\boldsymbol{\alpha}_l |w_l|} - \frac{\boldsymbol{\Omega}(\alpha_l - v_1 \mathbf{1})}{\nu_2}$, where $l = 1, \ldots, L$,

M-step for $\boldsymbol{\xi}$.

$$\xi_{li} = \frac{\nu_3}{|z_{li}| + \frac{1}{\nu_4}}.$$

Tuning

The parameters needed to be specified a priori include ν_1 and ν_2 from equation (4.5), a_ω and b_ω from equation (4.6), and ν_3 and ν_4 from equation (4.7). Based on our experience in numerical experiments, we fix a_ω as 4 and b_ω as 1 so that the prior of $\boldsymbol{\Omega}$ has large prior correlation and at the same time is relatively uninformative. We also fix ν_2 as $\ln 2$ and ν_3 as 1 so that the corresponding priors for $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$ have a unit

$\mathcal{C} \leftarrow \{j : x_j \text{ is Gaussian}\}$ and $\mathcal{D} \leftarrow \{j : x_j \text{ is discrete}\}$;

$t \leftarrow 0$;

repeat

  for $j \leftarrow 1$ **to** p **do**

    if $j \in \mathcal{D}$ **then** $\rho_{ji} \leftarrow \frac{b_j(e^{\mu_{ji}} - e^{\psi_{ji}})}{2(\mu_{ji} - \psi_{ji})(e^{\mu_{ji}} + e^{\psi_{ji}})}$ for $1 \leq i \leq n$;

    else $\rho_j \leftarrow \frac{\zeta_j + n}{\zeta_j + \sum_i (x_{ji} - \mu_{ji})^2}$;

  end

  for $i \leftarrow 1$ **to** n **do** $\mathbf{z}_{\cdot i} \leftarrow \text{argmin}_{\mathbf{z}} \left(\frac{1}{2} \mathbf{z}'(W)' D_i W \mathbf{z} - \mathbf{z}'(W)' \mathbf{c}_i + \sum_l \xi_{li} |z_l| \right)$;

  for $j \leftarrow 1$ **to** p **do**

    $(\mathbf{w}_{j\cdot})' \leftarrow \text{argmin}_{\mathbf{w}} \left(\frac{1}{2} \mathbf{w}' Z G_j(Z)' \mathbf{w} - \mathbf{w}' Z_{\mathrm{j}} + \sum_{\mathrm{l}} \lambda_{\mathrm{jl}} |\mathbf{w}_{\mathrm{l}}| \right)$;

    $m_j \leftarrow \frac{-\mathbf{w}_j' Z \boldsymbol{\rho}_j + \boldsymbol{\psi}_j' \boldsymbol{\rho}_j + \boldsymbol{\kappa}_j' \mathbf{1}}{\boldsymbol{\rho}_j' \mathbf{1}}$.

  end

  $$\phi_{jl} \leftarrow \begin{cases} \frac{N(e^{\alpha_{jl}} - |w_{jl}|/N)}{2(\alpha_{jl} + \log(|w_{jl}|/N))(e^{\alpha_{jl}} + |w_{jl}|/N)}, & w_{jl} \neq 0, \\ 0, & w_{jl} = 0; \end{cases}$$

  $\boldsymbol{\alpha}_l \leftarrow (\Omega/\nu_2 + \Phi_l)^{-1} (\Omega \underline{1} \nu_1/\nu_2 + \Phi_l \boldsymbol{\chi}_l + (1 - N/2)\underline{1})$;

  $\omega_{jk} \leftarrow \frac{2\nu_2 a_\omega}{2\nu_2 b_\omega + \sum_l (\alpha_{jl} - \alpha_{kl})^2}, \qquad 1 \leq j < k \leq p, (j,k) \in E$;

  $t \leftarrow t + 1$;

until *convergence*;

Algorithm 1: EM algorithm for GBC with EMSHS prior

coefficient of variation. ν_1 and ν_4 control the sparseness of the solutions to $\mathbf{W}$ and $\mathbf{Z}$, i.e., the size of each bicluster. We choose ν_1 and ν_4 by the Bayesian information criterion (BIC). The BIC is given by

$$BIC = -2\ln(L(\mathbf{X}, \hat{\boldsymbol{\mu}})) + (||\hat{\mathbf{W}}||_0 + ||\hat{\mathbf{Z}}||_0)\ln(np)$$

where $L(\mathbf{X}, \hat{\boldsymbol{\mu}})$ is the observed likelihood of $\boldsymbol{\mu}$, $||\hat{\mathbf{W}}||_0$ and $||\hat{\mathbf{Z}}||_0$ are the cardinalities of $\hat{\mathbf{W}}$ and $\hat{\mathbf{Z}}$. We conduct grid search and the combinations of ν_1 and ν_4 with the smallest BIC value are chosen as the optimal tuning parameter values for each simulation dataset and real data application. Again based on our experience, the search area in simulation is $\{2, 3, 4, 5, 6, 7\}$ by $\{10, 20, 30, 40, 50, 60\}$ for ν_1 and ν_4. In real data analysis, we choose higher values in the search $\{7, 9, 11, 13, 15, 20, 25\}$ by $\{20, 40, 50, 60, 70, 90, 110\}$, as the previous experience shows real datasets need larger tuning parameter to achieve the smallest BIC.

4.3 Simulation

4.3.1 Settings

In each simulation setting, we generate 100 simulation datasets. Each dataset has $p = 1000$ genes and $n = 300$ samples. We assume $L = 5$ underlying true biclusters. The parameter μ is computed by a mulplicative model $\boldsymbol{\mu} = \mathbf{WZ}$ where $\mathbf{W}$ is a $p \times L$ matrix and $\mathbf{Z}$ is a $L \times n$ matrix. The number of non-zero elements in each column of $\mathbf{W}$ is set as 50 and the number of non-zero elements in each row of $\mathbf{Z}$ is randomly drawn from Poisson distribution with parameter 30. The row numbers with non-zero elements in $\mathbf{W}$ are consecutive while the column numbers with non-zero elements in $\mathbf{Z}$ are randomly drawn from 1 to n. And the elements of different columns of $\mathbf{W}$ are allowed to have overlaps. The non-zero elements of both $\mathbf{W}$ and $\mathbf{Z}$ are generated from normal distribution with mean 1.5 and standard deviation 0.1, and are randomly

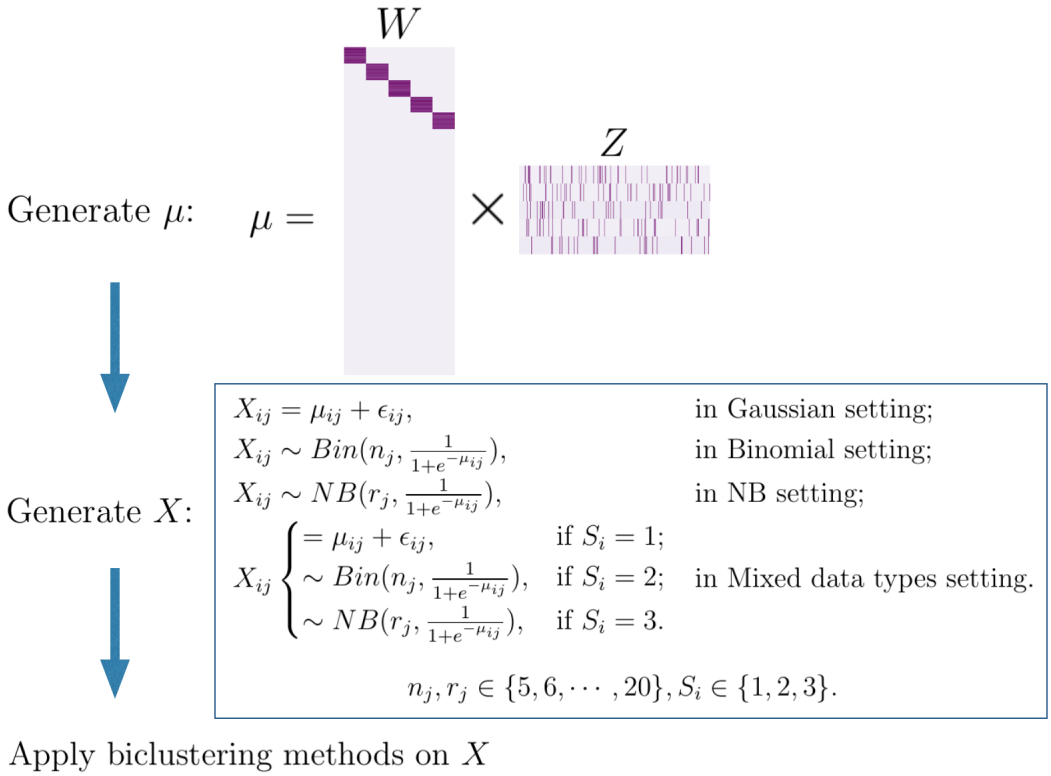


Figure 4.2: Work flow of the simualtion study.

assigned to be positive or negative. We use O to represent the number of overlapping rows/columns between adjacent biclusters. O is set to 0 or 15.

Four simualtion settings are generated: Gaussian, Binomial, Negative Binomial, and mixed data types. For the Gaussian case, the observed $p \times n$ data matrix $\mathbf{X}$ is generated by $\mathbf{X} = \boldsymbol{\mu} + \boldsymbol{\epsilon}$. The noise elements ϵ_{ij} are randomly chosen from $\mathcal{N}(0, 4)$. For the Binomial case, each element of $\mathbf{X}$ is generated from $Binomial(n_j, \frac{1}{1+e^{-\mu_{ij}}})$ and n_j is randomly sampled from 5 to 20. Similarly, for the Negative Binomial case, each element of $\mathbf{X}$ is generated from $NB(r_j, \frac{1}{1+e^{-\mu_{ij}}})$ and the parameter r_j is randomly sampled from 5 to 20. For the mixed data type, we randomly sample each row from these three distributions with the same parameter values as the previous three settings.

4.3.2 Methods

We apply the proposed methods and existing methods on the simulated datasets. GBC represents the proposed method without utilizing any biological information and sGBC is the version incorporating biological information. As discussed in section 2, GBC incorporates structural information by employing a graph-Laplacian prior on the shrinkage parameter $\boldsymbol{\lambda}$. For each simulation dataset, an working edge matrix is generated by assuming that each bicluster is a fully connected graph and randomly sampling five percent of true edges from all the underlying true biclusters. These edge matrices are used as structural information in sGBC.

The existing methods used as comparators are plaid Caldas and Kaski (2008), CC Cheng and Church (2000), FABIA Hochreiter et al. (2010), xMotifs Murali and Kasif (2002), and ISA Bergmann et al. (2003). All the methods have implementations in R. Specifically, FABIA is implemented in R/Bioconductor package *FABIA*, ISA is implemented in R/CRAN package *isa2*, and plaid, CC, and xMotifs are implemented in R/CRAN package *biclust*. To choose appropriate tuning parameters for each method, we have evaluated the tuning parameter options provided in Padilha and Campello (2017) and Eren et al. (2012). For FABIA, we use the default tuning parameter set when fitting the model and a threshold of 0.5 on Z when extracting biclusters. For CC, we choose the δ value as 0.25 and α as 1.2. For Plaid, we find the best combination of row.release and col.release in the interval $[0.1, 0.5]$ with steps of 0.1. For xMotifs, we relax the α to 0.05 as suggested in Padilha and Campello (2017) and used $sd = 5$ in synthetic datasets and $sd = 1$ in real data applications, because otherwise no biclusters can be identified.

4.3.3 Evaluation Criteria

Two evaluation criteria are used in both the simulation study and real data applications: clustering error (CE) Patrikainen and Meila (2006) and consensus score (CS)

Hochreiter et al. (2010). Clustering error finds the maximum overlapping proportions of two biclusters after an optimal matching of clusters. Similarly, CS finds the optimal mapping between clusters that maximizes the sum of similarities between matched pairs. The only difference between CE and CS is that CS uses the size of bicluster union at the denominator. This means CS does not take the size of each bicluster into consideration and gives the same weights on all biclusters. Thus, big biclusters may have greater impact on CE than CS. It is worth noting that our CE is one minus the CE defined in Patrikainen and Meila (2006), which in our opinion might be easier for comparison. Both CE and CS lie between 0 and 1. Higher CE, CS values mean greater overlapping between estimated biclusters and true biclusters.

Besides CE and CS, we also compute sensitivity (SEN), specificity (SPE) and Matthews correlation coefficient (MCC) in the simulation studies. All these metrics also have values between 0 and 1, and higher values indicate better performance.

Gaussian						
overlap	Method	CE	CS	SEN	SPE	MCC
0	Plaid	0.24(3e-02)	0.24(3e-02)	0.29(2e-02)	1(5e-06)	0.43(5e-02)
	CC	0(0e+00)	0(0e+00)	0(0e+00)	1(5e-05)	-0.0025(1e-04)
	FABIA	0.54(3e-02)	0.54(3e-02)	0.57(3e-02)	1(1e-04)	0.72(3e-02)
	XMotifs	0(0e+00)	0(0e+00)	0(0e+00)	1(0e+00)	0(0e+00)
	ISA	0(0e+00)	0(0e+00)	0(0e+00)	0(0e+00)	0(0e+00)
	GBC	0.64(9e-02)	0.63(9e-02)	0.88(1e-01)	0.99(4e-03)	0.78(6e-02)
	sGBC	0.76(7e-02)	0.76(8e-02)	0.95(8e-02)	0.99(2e-03)	0.86(5e-02)
15	Plaid	0.24(2e-02)	0.23(3e-02)	0.28(2e-02)	1(1e-04)	0.42(4e-02)
	CC	0(0e+00)	0(0e+00)	0(0e+00)	1(5e-05)	-0.0027(1e-04)
	FABIA	0.51(8e-02)	0.52(7e-02)	0.56(3e-02)	1(1e-03)	0.68(9e-02)
	XMotifs	0(0e+00)	0(0e+00)	0(0e+00)	1(0e+00)	0(0e+00)
	ISA	0(0e+00)	0(0e+00)	0(0e+00)	0(0e+00)	0(0e+00)
	GBC	0.57(1e-01)	0.57(1e-01)	0.91(1e-01)	0.98(7e-03)	0.76(7e-02)
	sGBC	0.66(9e-02)	0.66(9e-02)	0.95(9e-02)	0.99(4e-03)	0.81(5e-02)

Table 4.2: Simulation results for Gaussian settings. Results are generated based on 100 simulated datasets: mean(sd).

Binomial						
overlap	Method	CE	CS	SEN	SPE	MCC
0	Plaid	0.01(9e-04)	0.18(2e-02)	0.4(2e-02)	0.9(1e-01)	0.036(3e-03)
	CC	0.0048(8e-04)	0.0022(4e-04)	0.015(2e-03)	0.99(2e-04)	0.003(2e-03)
	FABIA	0.072(1e-02)	0.37(2e-02)	0.41(2e-02)	0.98(2e-03)	0.17(2e-02)
	XMotifs	0.0013(9e-04)	0.0013(9e-04)	0.0014(1e-03)	1(4e-05)	0.003(3e-03)
	ISA	0(0e+00)	0(0e+00)	0(0e+00)	0(0e+00)	0(0e+00)
	GBC	0.57(1e-01)	0.6(1e-01)	0.99(1e-02)	0.98(9e-03)	0.77(7e-02)
	sGBC	0.61(1e-01)	0.63(9e-02)	1(8e-04)	0.98(6e-03)	0.79(6e-02)
15	Plaid	0.012(1e-03)	0.17(2e-02)	0.4(2e-02)	0.82(5e-02)	0.039(4e-03)
	CC	0.0064(1e-03)	0.0027(4e-04)	0.017(3e-03)	0.99(2e-04)	0.005(2e-03)
	FABIA	0.1(3e-02)	0.34(4e-02)	0.39(3e-02)	0.98(4e-03)	0.21(4e-02)
	XMotifs	0.0014(9e-04)	0.0014(9e-04)	0.0015(1e-03)	1(5e-05)	0.0036(3e-03)
	ISA	0.012(4e-03)	0.0033(1e-03)	0.017(7e-03)	1(2e-04)	0.025(8e-03)
	GBC	0.43(2e-01)	0.48(1e-01)	1(9e-03)	0.97(1e-02)	0.7(8e-02)
	sGBC	0.6(1e-01)	0.61(9e-02)	1(3e-03)	0.98(6e-03)	0.79(5e-02)

Table 4.3: Simulation results for Binomial settings. Results are generated based on 100 simulated datasets: mean(sd).

4.3.4 Results

Table 4.2-4.5 present simulation results for Gaussian, Binomial, Negative Binomial, and mixed data type settings respectively. All the results are generated based on 100 Monte Carlo datasets. Table 4.2 shows that in the Gaussian case, FABIA, GBC and sGBC outperforms all the other methods. GBC and FABIA have similar CE and CS values, around 0.5 for both non-overlapping scenario and overlap= 15 scenario. sGBC has higher CE and CS, around 0.7 for non-overlapping scenario and around 0.6 for overlap= 15 scenario. CC, xMotifs and ISA have the worst results with CE and CS around 0, suggesting that they fail to identify any biclusters. Plaid has a performance better than CC, xMotifs, and ISA but worse than GBC and FABIA, with CE and CS values around 0.2.

Table 4.3 shows that in the Binomial case, GBC and sGBC still performs best

Negative Binomial						
overlap	Method	CE	CS	SEN	SPE	MCC
0	Plaid	0.012(1e-03)	0.15(2e-02)	0.35(3e-02)	1(0e+00)	0.04(6e-03)
	CC	0.00043(3e-04)	0.00039(3e-04)	0.00047(4e-04)	1(3e-05)	-7.7e-05(1e-03)
	FABIA	0.21(3e-02)	0.21(3e-02)	0.21(3e-02)	1(1e-04)	0.42(6e-02)
	XMotifs	0.0028(1e-03)	0.0028(1e-03)	0.0031(1e-03)	1(4e-05)	0.0075(4e-03)
	ISA	0(0e+00)	0(0e+00)	0(0e+00)	0(0e+00)	0(0e+00)
	GBC	0.49(2e-01)	0.53(1e-01)	1(9e-04)	0.98(1e-02)	0.72(8e-02)
	sGBC	0.48(2e-01)	0.52(1e-01)	1(0e+00)	0.97(1e-02)	0.71(8e-02)
15	Plaid	0.036(3e-02)	0.089(3e-02)	0.26(5e-02)	1(1e-02)	0.09(6e-02)
	CC	0.00023(2e-04)	0.00022(2e-04)	0.00025(3e-04)	1(3e-05)	-0.00083(9e-04)
	FABIA	0.17(3e-02)	0.17(3e-02)	0.17(3e-02)	1(1e-04)	0.38(5e-02)
	XMotifs	0.0024(1e-03)	0.0024(1e-03)	0.0027(1e-03)	1(3e-05)	0.007(5e-03)
	ISA	0.0039(3e-03)	0.0035(3e-03)	0.004(3e-03)	1(6e-05)	0.024(2e-02)
	GBC	0.42(2e-01)	0.47(1e-01)	1(8e-04)	0.97(2e-02)	0.69(9e-02)
	sGBC	0.49(1e-01)	0.53(1e-01)	1(4e-04)	0.97(1e-02)	0.73(6e-02)

Table 4.4: Simulation results for Negative Binomial settings. Results are generated based on 100 simulated datasets: mean(sd).

Mixed						
overlap	Method	CE	CS	SEN	SPE	MCC
0	Plaid	0.011(1e-03)	0.073(1e-02)	0.23(3e-02)	1(1e-02)	0.027(6e-03)
	CC	6.3e-05(9e-05)	6e-05(9e-05)	6.8e-05(1e-04)	1(2e-05)	-0.0011(4e-04)
	FABIA	0.1(2e-02)	0.1(2e-02)	0.11(2e-02)	1(5e-04)	0.3(5e-02)
	XMotifs	1.2e-06(1e-05)	1.1e-06(1e-05)	1.2e-06(1e-05)	1(4e-05)	-0.00012(3e-04)
	ISA	0(0e+00)	0(0e+00)	0(0e+00)	0(0e+00)	0(0e+00)
	GBC	0.48(2e-01)	0.51(1e-01)	0.85(7e-02)	0.98(1e-02)	0.69(9e-02)
	sGBC	0.7(1e-01)	0.71(1e-01)	0.99(1e-02)	0.99(6e-03)	0.84(6e-02)
15	Plaid	0.019(1e-02)	0.043(1e-02)	0.16(3e-02)	1(1e-02)	0.042(3e-02)
	CC	4.1e-05(7e-05)	4e-05(7e-05)	4.4e-05(7e-05)	1(3e-05)	-0.0013(3e-04)
	FABIA	0.1(2e-02)	0.1(2e-02)	0.1(2e-02)	1(7e-04)	0.29(6e-02)
	XMotifs	5.1e-06(3e-05)	4.7e-06(3e-05)	5.2e-06(3e-05)	1(5e-05)	-0.00014(4e-04)
	ISA	0(0e+00)	0(0e+00)	0(0e+00)	0(0e+00)	0(0e+00)
	GBC	0.51(1e-01)	0.53(1e-01)	0.89(6e-02)	0.98(1e-02)	0.72(7e-02)
	sGBC	0.64(1e-01)	0.66(8e-02)	0.97(3e-02)	0.99(6e-03)	0.81(5e-02)

Table 4.5: Simulation results for mixed data types. Results are generated based on 100 simulated datasets: mean(sd).

with CE and CS more than 0.5, but FABIA performs worse than the Gaussian case. In addition, all the other methods, Plaid, CC, xMotifs and ISA all perform poorly in this setting. It is worth noting that incorporating structural information in GBC is shown to effectively improve performance in both settings. For example, in Gaussing setting with zero overlap, sGBC improves CE from 0.557 to 0.724, which is about a 30 percent increase.

Table 4.4 and 4.5 show that in the Negative Binomial and mixed data types, GBC and sGBC still perform best among all the methods. They reach CE and CS around 0.6 in Negative Binomial, around 0.5 in mixed data types. FABIA also outperforms the rest of the methods, reaching CE and CS values ranging from 0.1 to 0.2. Plaid, CC, xMotifs and ISA still have the worst results, with CE and CS around 0.

In addition to CE and CS, the proposed methods also have better performance in sensitivity, specificity and MCC. We find all the methods generally have high specificity and low sensitivity, suggesting that they fail to identify biclusters instead of mis-identifying biclusters. And sGBC usually has higher sensitivity than GBC, indicating that considering structural information helps improve the sensitivity of identifying true biclusters.

4.4 Real data applications

To evaluate our methods in comparison with the existing methods in real data applications, we obtain one gene expression dataset, one proteomics dataset, one RNAseq dataset and one integrative dataset. The first three datasets have validated or known subgroup/cluster information on subject level, which are used as gold standard to compute all evaluation metrics. In the integrative data set, there are no known or validated subgroups. To assess performance, we use patient survival time to define subgroups, which provides evidence that clusters detected by a method are clinically meaningful.

Table 4.6: Results of real data applications.

Method	Breast cancer: microarray		ASD: proteomics		ASD: RNAseq		GBM: mixed	
	CE	CS	CE	CS	CE	CS	CE	CS
PLAID	0.186	0.169	0	0	0	0	0.263	0.175
CC	0.274	0.257	0.238	0.200	0.147	0.125	0.004	0.004
FABIA	0.315	0.262	0.254	0.140	0.147	0.103	0.260	0.186
xMotif	0	0	0.106	0.081	0	0	0	0
ISA	0.014	0.003	0.045	0.010	0.113	0.096	0.045	0.015
GBC	0.331	0.259	**0.313**	**0.167**	**0.239**	**0.211**	0.265	**0.263**
sGBC	**0.609**	**0.430**	**0.313**	0.160	**0.239**	**0.211**	**0.281**	0.221

4.4.1 Gene expression datasets

We re-analyze a gene expression dataset provided in the R/Bioconductor package *fabiaData* (Hochreiter et al., 2010). This dataset is originally provided by the Broad Institute and has been analyzed in Hoshida et al. (2007) and Hochreiter et al. (2010). The cluster information has been validated by gene enrichment analysis (Hoshida et al., 2007) and is used as the ground truth in our analysis. This *"breast cancer"* dataset includes 97 samples and 1213 genes (Van't Veer et al., 2002). It aims at identifying predictive biomarkers for a theraputic treatment of breast cancer patients. Three validated subclasses have been reported on the subject level and are used as ground truth in the biclustering analysis.

We apply the proposed methods and existing methods to analyze the dataset and report the results in the second and third columns of Table 4.6. We set the maximum number of clusters to 5 in all the methods. Using the validated subclass information, we obtain CE and CS for all the methods. The biological information is extracted from KEGG Pathways Kanehisa and Goto (2000) using Bioconductor package *KEGGgraph* and *KEGGREST* (Tenenbaum, 2013; Zhang and Wiemann, 2009). 1616 edges are

obtained and used as biological information in sGBC. The proposed method sGBC has the highest CE and CS values, and GBC has the second best CE and CS among all the other methods. This indicates that the prior biological information is helpful in identifying biclusters and almost doubles the CE performance of GBC. Note that some existing methods also have good performance. For example, FABIA has similar CE and CS values to GBC, and CC also have good CE and CS performance, which is consistent with the findings in Hochreiter et al. (2010).

4.4.2 Proteomics dataset

A proteomics dataset is obtained from the AMP-AD knowledge portal of the Synapse website (www.synapse.org) with ID *syn3607470*. Synapse is an organization dedicated to the research of brain diseases and service patients who have brain injuries. This proteomics dataset includes the measurements for 6533 protein levels from 20 Alzheimer's Disease (AD) patients, 13 Asymptomatic Alzheimer's Disease (AsymAD) patients, and 14 controls. All the measurements are conducted on post-mortem brain tissues from both the dorsolateral prefrontal cortex and precuneus. Both regions have been previously reported to be affected in AD (Cox et al., 2011) . The disease status of all subjects was confirmed through post-mortem neuropathological evaluation, and is used as ground truth in our analyses. According to the data description, the dataset has been normalized based on isotopically labeled retention time peptide standards and the central limit tendency theorem 3 (Callister et al., 2006). To remove noise, we use the top 300 variables with the largest variance.

We apply all the methods on this dataset and report CE and CS in the fourth and fifth columns of Table 4.6. We set the maximum number of clusters to 5 in all the methods. Pathway information is extracted from KEGG Pathway and used in the sGBC. GBC and sGBC achieves the highest CE and CS among all the methods. CC, xMotifs and FABIA have relatively good performance with CE more than 0.20.

On this dataset Plaid does not find any biclusters.

4.4.3 RNAseq dataset

An RNA-seq dataset is obtained from the AMP-AD knowledge portal of the Synapse website with ID *syn5223705*. This dataset include next-generation RNA sequencing (RNAseq) from 82 AD patients, 84 progressive supranuclear palsy(PSP) patients, 28 pathologic aging(PA) subjects, and 77 elder controls. These measurements are from cerebellum RNA samples collected by the Mayo Clinic Brain Bank and Banner Sun Health Research Institute. Reads are aligned by the SNAPR software [2] with the GRCh38 reference and Ensembl v77 gene models and data are normalized by the R/Bioconductor package edgeR (Robinson et al., 2010). The original dataset has 64253 features and we use the top 300 features with largest variability for the biclustering analysis. Pathway information is extracted from KEGG Pathway and used in the sGBC as prior biological information.

We apply all the methods on this dataset and CE and CS are reported in the sixth and seventh columns of Table 4.6. We set the maximum number of clusters to 4 in all the methods. In Table 6, GBC and sGBC have similar CE and CS performance and are the best performing methods among all the methods. CC and FABIA are the second best methods and have CE 0.147 and CS around 0.1. PLAID and xMotif do not find any biclusters in this dataset.

4.4.4 Integrative dataset

The data of this integrative analysis are obtained from a TCGA study in glioblastoma multiforme (GBM), which is the most common and aggressive type of malignant brain tumor Holland (2000). From the TCGA data portal [3], microarray gene expression data, DNA methylation data, and DNA copy number data are downloaded for a co-

[2] https://price.systemsbiology.org/research/snapr/
[3] http://tcga-data.nci.nih.gov/tcga/

hort of 233 GBM patients. All the data are pre-processed, normalized and annotated to the gene level (see Wang et al. (2012) for details). Our analysis focus on 48 genes that overlap with the three critical signaling pathways - *RTK/PI3K*, *p53*, and *Rb*, which have been found to relate with migration, survival and apoptosis progression of cell cycles Furnari et al. (2007b). Thus the data matrix consists of 48 genes mapped to these core pathways from 3 platforms resulting in $p = 48 \times 3 = 144$ for $n = 233$ subjects. Note that both microarray gene expression data and DNA methylation data are continuous, while copy number is converted to binary data via thresholding, having 0 corresponding to normal probes and 1 corresponding to abnormal (gain or loss) probes. The survival information of all subjects is obtained. We use Kaplan-Meier imputed survival time in the case that the subjects are censored, and we categorize the subjects into four groups according to their survival time (or imputed survival time) using 25th, 50th, 75th percentile as cutoffs. These four groups are used as ground truth for clustering patients.

We conduct biclustering analysis using the existing methods and the proposed methods. Five are given to all methods as maximum number of biclusters. In GBC and sGBC, we use normal distribution for both microarray gene expression data and DNA methylation data, and binomial distribution for copy number data. A total of 488 edges are extracted from the KEGG Pathway, and are used as biological information in sGBC. We have visualized the gene interaction graph of these 488 edges in Figure (4.1). We present the CE and CS in the last two columns of Table 4.6. GBC and sGBC have highest CE and CS values among all the methods. Plaid and FABIA also have similar CE values as GBC, which is around 0.26. GBC has higher CS value while sGBC has higher CE value, which may indicate that GBC identify more biclusters regardless of their sizes while GBC with biological information incorporated can identify biclusters with larger size.

4.5 Conclusion

In this paper, we propose a Bayesian biclustering algorithm which not only adapts to inputs of different types but also can incorporate biological information. Although a large number of different biclustering approaches have been developed, we are not aware of any existing biclustering methods that can incorporate prior biological information. In addition, our simulation study demonstrates that none of the existing methods considered can efficiently identify biclusters using input data of various distribution types. The proposed methods fill these gaps and become a useful tool in integrative analysis of multiple genomic datasets or analysis of single genomic dataset including gene expression, proteomics data, RNA-seq data, etc.. In the integrative data set, there are no known or validated subgroups. To assess performance, we use patient survival time to define subgroups, which provides evidence that clusters detected by a method are clinically meaningful.

Future directions of research may address three key challenges. The first challenge is to include more input datatypes in addition to Gaussian, Binomial, and Negative Binomial, for example, beta−Binomial distribution as in bisulfite sequencing data. To achieve this goal, one may need to seek other solutions instead of using the pólya-gamma framework. The second challenge is that the current methods may not be able to retrieve useful biclustering information when the input data matrix is very sprase, such as data matrices containing the information of somatic mutations. Thus the direction of developing biclustering methods for sparse data matrix is worth further investigation. Last but not least, the current biclustering performance is far from satisfactory. Even in simulation studies, the best CE achieved is only around 0.7. One may consider to combine the existing biclustering approaches and develop a ensemble approach for biclustering, similar to the approach of combining multiple machine learning algorithms in the popular AdaBoost algorithm.

Chapter 5

Future work

Although all three topics of this dissertation target on the analysis of big biomedical data, they actually utilize quite different strategies. Topic 1 is an extension of classical principal component analysis while topic 2 obtains its inspiration from machine learning field. Topic 3 combines the classical Bayesian framework with a technique in machine learning field. All three methods have been demonstrated to be useful in real world applications. As future works, we have identified two potential directions.

The first is to combine deep neural network technique into sparse principal component analysis. Neural network has the capability of analyzing large amounts of data and has flexible model structure. As introduced in section 1.3.2, deep neural network has achieved amazing performance in tasks such as picture recognition or playing GO. Genomic data are known for the large number of variables and complication of analysis. Thus it is very possible that appopriate application of deep learning on analyzing genomic data can result in good performance and exciting discoveries. We consider to combine deep learning teniques with multivariate analysis methods, espectially principal component analysis. There has been some recent work on using deep learning in multivariate analysis, such as deep CCA and deep PCA (Andrew et al., 2013; Tian et al., 2015). But none of these methods are suitable for high-dimensional data because they do not encourage variable selection, which is important and yields interpretable results when the number of variables is huge compared to the number of samples. Thus in the first potential direction, one can develop PCA methods using deep learning techniques which encourages sparse loading selection. This may be achieved by replacing the last layer of neural network by a layer of linear combinations of variables. And intead of using the conventional loss function such as cross entropy and hinge loss, one can use an objective function involves variance of the linear combinations on the last layer and tune the deep neural network so that it maximize the component variances.

The second potential direction is to improve the methods proposed in the second topic using the most current context embedding techniques. Recently some new node embedding models have been proposed and shown to outperform Word2vec in tasks like document classification and social network analysis (Tang et al., 2015; Ribeiro et al., 2017). To this end, one may be able to learn embedding vectors of medical events by utilizing these newly proposed approaches and achieve better prediction performance.

Last but not least, one can further extend the third topic by incorporating the phenotype information of subjects. One may achieve this by imposing priors which considers the characteristics of subjects, such as age and gender. Subjects with similar age and same gender may be more correlated than patients with big age difference and opposite genders.

Appendix A

Appendix for Chapter 2

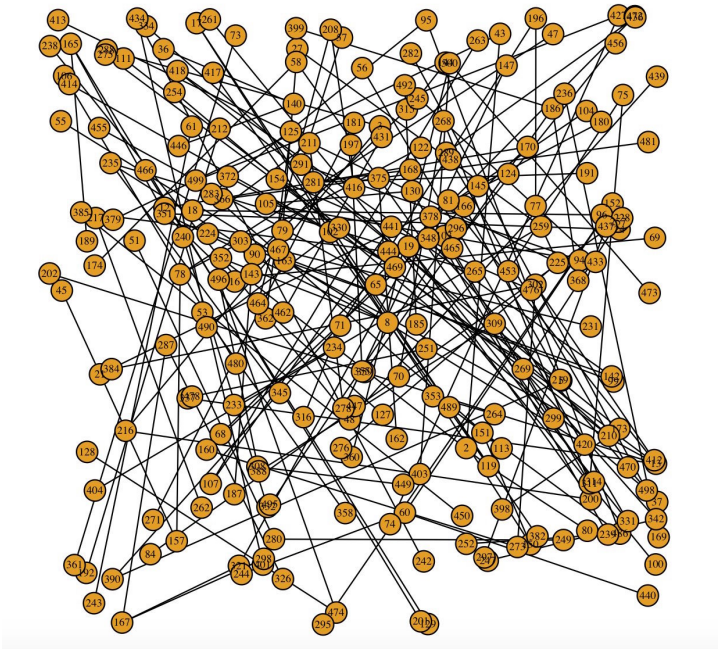
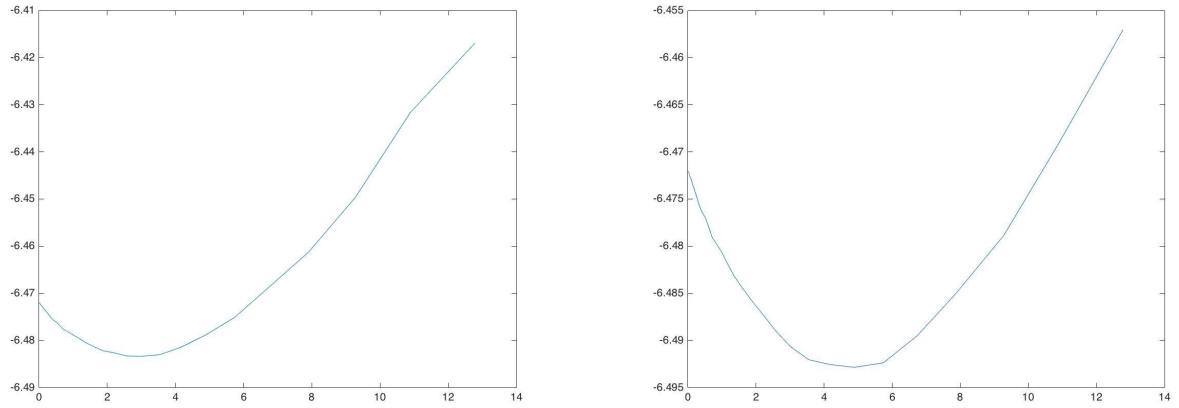


Figure A.1: Network structure of simulated data : Randomly specified graph ($\mathcal{G}$)



(a) Fused BIC value by tuning parameter. (b) Grouped BIC value by tuning parameter.

Figure A.2: BIC value by tuning parameter with GBM microarray data. X-axis is tuning parameter, y-axis is BIC value.

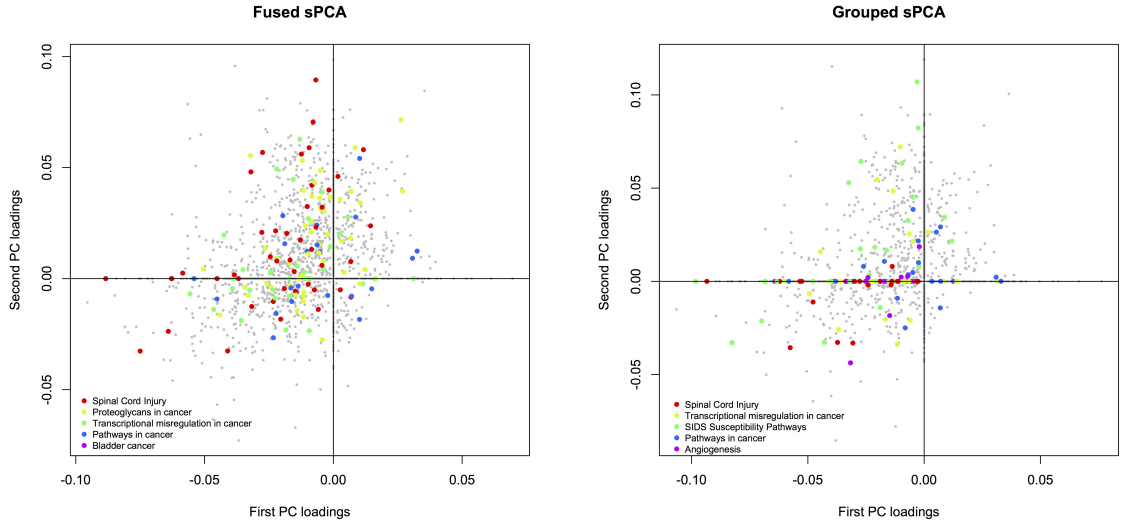


Figure A.3: Loading plots of the first two PCs by Fused and Grouped sPCA. Colored points are genes enriched in Glioblastoma related pathways found by the proposed methods but not found by existing methods.

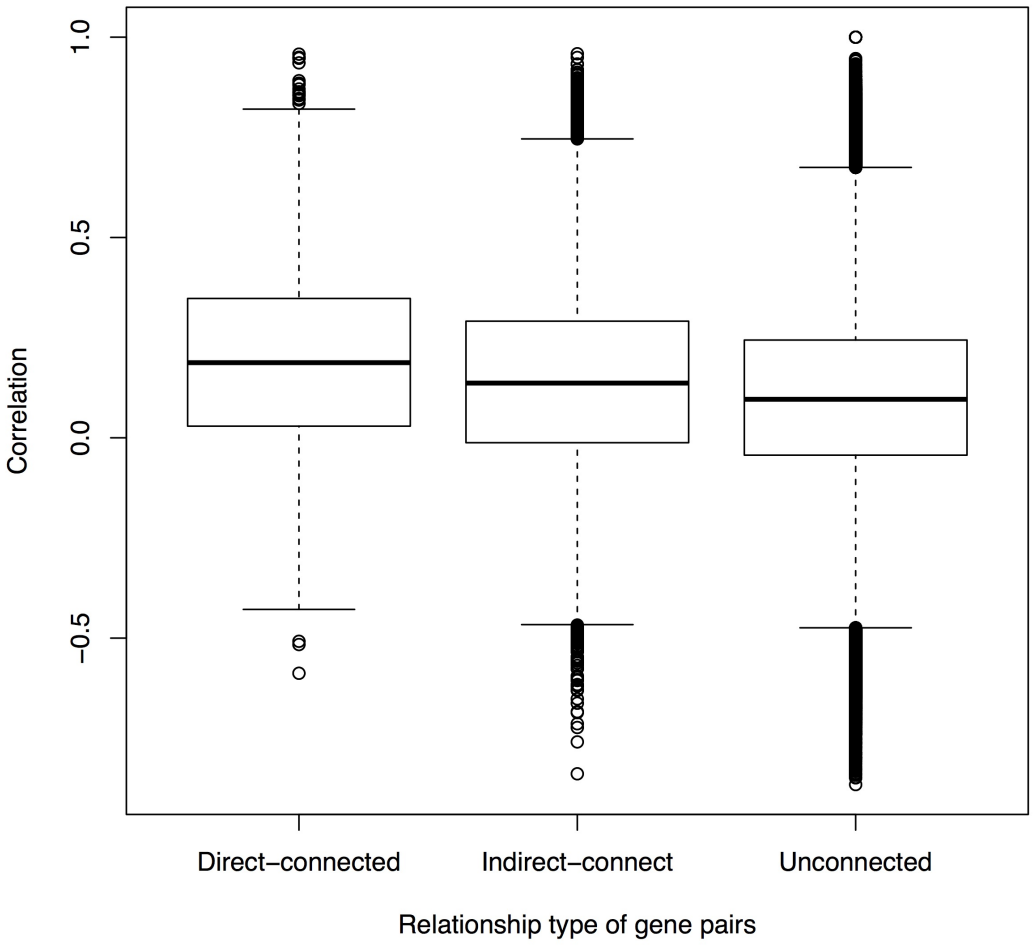


Figure A.4: Correlation of gene pairs by relationship types

Appendix B

Appendix for Chapter 3

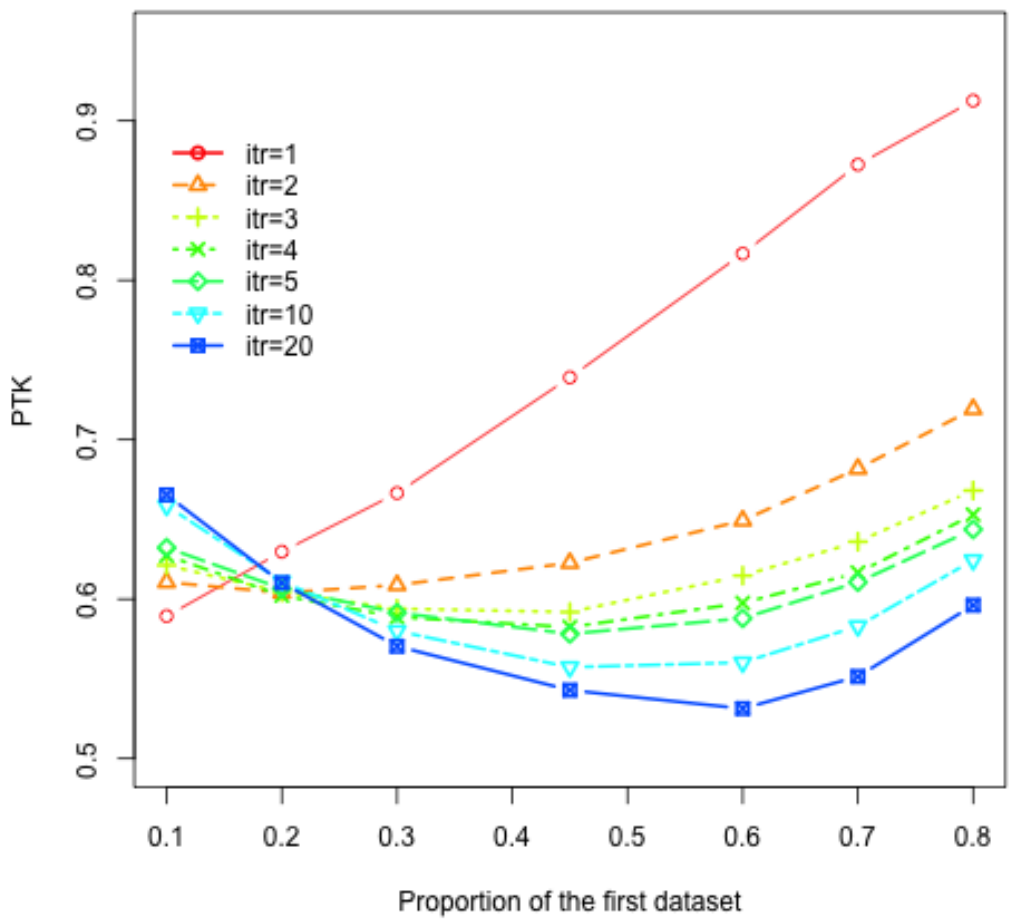


Figure B.1: Precision-Top-K versus K by Distributed NCE using different training sets partitions for different number of iterations.

	Naive updates		Dropout udpates		Distributed		Global	
	PTK	*Avg_AUC*	PTK	*Avg_AUC*	PTK	*Avg_AUC*	PTK	*Avg_AUC*
10:80:10	0.476 (4e-3)	0.772 (8e-3)	0.133 (2e-3)	0.720 (5e-3)	0.610 (4e-3)	0.773 (8e-3)	0.686 (3e-3)	0.774 (8e-3)
20:70:10	0.526 (4e-3)	0.773 (8e-3)	0.168 (3e-3)	0.721 (6e-3)	0.608 (3e-3)	0.774 (8e-3)		
30:60:10	0.526 (4e-3)	0.773 (8e-3)	0.188 (2e-3)	0.722 (7e-3)	0.608 (4e-3)	0.774 (8e03)		
45:45:10	0.510 (3e-3)	0.773 (8e-3)	0.225 (3e-3)	0.723 (7e-3)	0.572 (2e-3)	0.774 (8e-3)		
60:30:10	0.491 (3e-3)	0.774 (8e-3)	0.266 (4e-3)	0.724 (5e-3)	0.572 (3e-3)	0.773 (8e-3)		
70:20:10	0.484 (3e-3)	0.776 (7e-3)	0.310 (4e-3)	0.727 (6e-3)	0.581 (2e-3)	0.774 (7e-3)		
80:10:10	0.480 (4e-3)	0.776 (7e-3)	0.383 (8e-3)	0.736 (7e-3)	0.609 (6e-3)	0.774 (7e-3)		

Table B.1: Simulation results of all methods using $Skip-Gram$ model. Results are summarized over 10-folds cross validation. Distributed NCE is Distributed Noise Contrastive Estimation. PTK is Precision-Top-K. Avg-AUC is averaged Area-Under-Curve. $n_{t1} : n_{t2} : n_{test}$ means that the two training datasets are $n_{t1}\%$ and $n_{t2}\%$ of total data. Testing dataset is $n_{test}\%$ of total data. Global model uses all $n_{t1} + n_{t2}\%$ data as training data.

Bibliography

Abadie, A. and Imbens, G. W. (2006), 'Large sample properties of matching estimators for average treatment effects', Econometrica **74**(1), 235–267.

Adibi, S. (2014), mHealth multidisciplinary verticals, CRC Press.

Allen, G. I., Grosenick, L. and Taylor, J. (2014), 'A generalized least-square matrix decomposition', Journal of the American Statistical Association **109**(505), 145–159.

Anderson, T. W. (1962), An introduction to multivariate statistical analysis, Technical report, Wiley New York.

Anderson, T. W. (1984), An Introduction to Multivariate Statistical Analysis, Wiley Series in Probability and Mathematical Statistics.

Andrew, G., Arora, R., Bilmes, J. A. and Livescu, K. (2013), Deep canonical correlation analysis., in 'ICML (3)', pp. 1247–1255.

Balabin, R. M. and Lomakina, E. I. (2009), 'Neural network approach to quantum-chemistry data: Accurate prediction of density functional theory energies', The journal of chemical physics **131**(7), 074104.

Batal, I., Fradkin, D., Harrison, J., Moerchen, F. and Hauskrecht, M. (2012), Mining recent temporal patterns for event detection in multivariate time series data, in 'Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 280–288.

Ben-Dor, A., Chor, B., Karp, R. and Yakhini, Z. (2003), 'Discovering local structure in gene expression data: the order-preserving submatrix problem', Journal of computational biology **10**(3-4), 373–384.

Bergmann, S., Ihmels, J. and Barkai, N. (2003), 'Iterative signature algorithm for the analysis of large-scale gene expression data', Physical review E **67**(3), 031902.

Bowman, F. D. (2014), 'Brain imaging analysis', Annual review of statistics and its application **1**, 61.

Cadima, J. and Jolliffe, I. T. (1995a), 'Loading and correlations in the interpretation of principle compenents', Journal of Applied Statistics **22**(2), 203–214.

Cadima, J. and Jolliffe, I. T. (1995b), 'Loading and correlations in the interpretation of principle compenents', Journal of Applied Statistics **22**(2), 203–214.

Cai, T. and Liu, W. (2011), 'A direct estimation approach to sparse linear discriminant analysis', Journal of the American Statistical Association **106**(496), 1566–1577.

Cai, T., Ma, Z. and Wu, Y. (2013), 'Sparse pca: Optimal rates and adaptive estimation', The Annals of Statistics pp. 3074–3110.

Caldas, J. and Kaski, S. (2008), Bayesian biclustering with the plaid model, in 'Machine Learning for Signal Processing, 2008. MLSP 2008. IEEE Workshop on', IEEE, pp. 291–296.

Callister, S. J., Barry, R. C., Adkins, J. N., Johnson, E. T., Qian, W.-j., Webb-Robertson, B.-J. M., Smith, R. D. and Lipton, M. S. (2006), 'Normalization approaches for removing systematic biases associated with mass spectrometry and label-free proteomics', Journal of proteome research **5**(2), 277–286.

Candes, E. and Tao, T. (2007), 'The Dantzig selector: Statistical estimation when p is much larger than n', The Annals of Statistics **35**(6), 2313–2351.

CBP (2010), 'Top 10 largest databases in the world'.
URL: *http://www.comparebusinessproducts.com/fyi/10-largest-databases-in-the-world*

Chalise, P. and Fridley, B. L. (2012), 'Comparison of penalty functions for sparse canonical correlation analysis', Computational Statistics and Data Analysis **56**, 245–254.

Chang, C., Kundu, S. and Long, Q. (2016), 'Scalable bayesian variable selection for structured high-dimensional data', arXiv preprint arXiv:1604.07264 .

Chang, C. and Tsay, R. S. (2010), 'Estimation of covariance matrix via the sparse cholesky factor with lasso', Journal of Statistical Planning and Inference **140**(12), 3858–3873.

Che, Z., Kale, D., Li, W., Bahadori, M. T. and Liu, Y. (2015), Deep computational phenotyping, in 'Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 507–516.

Chen, D. and Manning, C. D. (2014), A fast and accurate dependency parser using neural networks., in 'EMNLP', pp. 740–750.

Chen, J., Bardes, E. E., Aronow, B. J. and Jegga, A. G. (2009), 'Toppgene suite for gene list enrichment analysis and candidate gene prioritization', Nucleic acids research **37**(suppl 2), W305–W311.

Chen, J., Bushman, F. D., Lewis, J. D., Wu, G. D. and Li, H. (2013), 'Structure-constrained sparse canonical correlation analysis with an application to microbiome data analysis', Biostatistics **14**(2), 244–258.

Cheng, Y. and Church, G. M. (2000), Biclustering of expression data., in 'Ismb', Vol. 8, pp. 93–103.

Cheng, Y., Wang, F., Zhang, P. and Hu, J. (2016+), 'Risk prediction with electronic health records: A deep learning approach'.

Choi, E., Bahadori, M. T., Searles, E., Coffey, C. and Sun, J. (2016), 'Multi-layer representation learning for medical concepts', arXiv preprint arXiv:1602.05568 .

Choi, E., Schuetz, A., Stewart, W. F. and Sun, J. (2016a), 'Medical concept representation learning from electronic health records and its application on heart failure prediction', arXiv preprint arXiv:1602.03686 .

Choi, E., Schuetz, A., Stewart, W. F. and Sun, J. (2016b), 'Using recurrent neural network models for early detection of heart failure onset', Journal of the American Medical Informatics Association p. ocw112.

Choi, Y. (n.d.), 'Learning low-dimensional representations of medical concepts'.

Collins, F. S. and Varmus, H. (2015), 'A new initiative on precision medicine', New England Journal of Medicine **372**(9), 793–795.

Cooper, L., Gutman, D. A., Long, Q., Johnson, B. A., Cholleti, S. R., Kurc, T., Saltz, J. H., Brat, D. J. and Moreno, C. S. (2010), 'The proneural molecular signature is enriched in oligodendrogliomas and predicts improved survival among diffuse gliomas', PloS one **5**(9), e12548.

Costello, S., Cockburn, M., Bronstein, J., Zhang, X. and Ritz, B. (2009), 'Parkinsons disease and residential exposure to maneb and paraquat from agricultural applications in the central valley of california', American Journal of Epidemiology **169**(8), 919–926.

Cox, J., Neuhauser, N., Michalski, A., Scheltema, R. A., Olsen, J. V. and Mann, M. (2011), 'Andromeda: a peptide search engine integrated into the maxquant environment', Journal of proteome research **10**(4), 1794–1805.

CVX Research, I. (2012), 'CVX: Matlab software for disciplined convex programming, version 2.0', `http://cvxr.com/cvx`.

de Aguiar, C. B. M., Garcez, R. C., Alvarez-Silva, M. and Trentin, A. G. (2002), 'Undersulfation of proteoglycans and proteins alter c6 glioma cells proliferation, adhesion and extracellular matrix organization', International journal of developmental neuroscience **20**(7), 563–571.

De Vine, L., Zuccon, G., Koopman, B., Sitbon, L. and Bruza, P. (2014), Medical semantic similarity with a neural language model, in 'Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management', ACM, pp. 1819–1822.

Deng, L. and Yu, D. (2014), 'Deep learning', Signal Processing **7**, 3–4.

Diaz, K. M., Krupka, D. J., Chang, M. J., Peacock, J., Ma, Y., Goldsmith, J., Schwartz, J. E. and Davidson, K. W. (2015), 'Fitbit®: An accurate and reliable device for wireless physical activity tracking.', International journal of cardiology **185**, 138–140.

Eigen, M. (1971), 'Selforganization of matter and the evolution of biological macromolecules', Naturwissenschaften **58**(10), 465–523.

Eren, K., Deveci, M., Küçüktunç, O. and Çatalyürek, Ü. V. (2012), 'A comparative analysis of biclustering algorithms for gene expression data', Briefings in bioinformatics **14**(3), 279–292.

Fan, J. and Li, R. (2001), 'Variable selection via nonconcave penalized likelihood and its oracle properties', Journal of the American Statistical Association **96**(456), 1348–1360.

Farhan, W., Wang, Z., Huang, Y., Wang, S., Wang, F. and Jiang, X. (2016+), From symbolic to semantic: a sequential predictive model for medical events, LWW.

Friedman, J., Hastie, T. and Tibshirani, R. (2008), 'Regularization paths for generalized linear models via coordinate descent', Journal of Statistical Software **33**, 1–22.

Furnari, F. B., Fenton, T., Bachoo, R. M., Mukasa, A., Stommel, J. M., Stegh, A., Hahn, W. C., Ligon, K. L., Louis, D. N., Brennan, C. et al. (2007a), 'Malignant astrocytic glioma: genetics, biology, and paths to treatment', Genes & developmeMcLendon:2008nt **21**(21), 2683–2710.

Furnari, F. B., Fenton, T., Bachoo, R. M., Mukasa, A., Stommel, J. M., Stegh, A., Hahn, W. C., Ligon, K. L., Louis, D. N., Brennan, C. et al. (2007b), 'Malignant astrocytic glioma: genetics, biology, and paths to treatment', Genes & development **21**(21), 2683–2710.

Gates, S. C. and Sweeley, C. C. (1978), 'Quantitative metabolic profiling based on gas chromatography.', Clinical chemistry **24**(10), 1663–1673.

Gingras, M.-c., Roussel, E., Bruner, J. M., Branch, C. D. et al. (1995), 'Comparison of cell adhesion molecule expression between glioblastoma multiforme and autologous normal brain tissue', Journal of neuroimmunology **57**(1), 143–153.

Grant, M. and Boyd, S. (2008), Graph implementations for nonsmooth convex programs, in V. Blondel, S. Boyd and H. Kimura, eds, 'Recent Advances in Learning and Control', Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, pp. 95–110.

Gu, J. and Liu, J. S. (2008), 'Bayesian biclustering of gene expression data', BMC genomics **9**(1), S4.

Gulcehre, C. (2016), 'Deep learning'.
URL: *http://deeplearning.net/*

Gunter, T. D. and Terry, N. P. (2005), 'The emergence of national electronic health record architectures in the united states and australia: models, costs, and questions', Journal of medical Internet research **7**(1), e3.

Hall, N. (2007), 'Advanced sequencing technologies and their wider impact in microbiology', Journal of Experimental Biology **210**(9), 1518–1525.

Hammerla, N. Y., Fisher, J., Andras, P., Rochester, L., Walker, R. and Plötz, T. (2015), Pd disease state assessment in naturalistic environments using deep learning., in 'AAAI', pp. 1742–1748.

Hancock, D. B., Martin, E. R., Mayhew, G. M., Stajich, J. M., Jewett, R., Stacy, M. A., Scott, B. L., Vance, J. M. and Scott, W. K. (2008), 'Pesticide exposure and risk of parkinson's disease: A family-based case-control study', BMC Neurology **8**.

Hartigan, J. A. (1972), 'Direct clustering of a data matrix', Journal of the american statistical association **67**(337), 123–129.

Hartzband, D. (2011), 'Using ultra-large data sets in healthcare'.
URL: *http://www.e−healthpolicy.org/docs/2011_sessions/20111005_hartzband_bigdata.pdf*

Hastie, T. and Tibshirani, R. (2004), 'Efficient quadratic regularization for expression arrays', Biostatistics **5**(3), 329–340.

HealthIT (2014), 'What is an electronic medical record (emr)?'.
URL: *https://www.healthit.gov/providers-professionals/electronic-medical-records-emr*

Hecht-Nielsen, R. (1989), Theory of the backpropagation neural network, in 'Neural Networks, 1989. IJCNN., International Joint Conference on', IEEE, pp. 593–605.

Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., Khamiakova, T., Van Sanden, S., Lin, D., Talloen, W. et al. (2010), 'Fabia: factor analysis for bicluster acquisition', Bioinformatics **26**(12), 1520–1527.

Hodge Jr, J. G., Gostin, L. O. and Jacobson, P. D. (1999), 'Legal issues concerning electronic health information: privacy, quality, and liability', Jama **282**(15), 1466–1471.

Holland, E. C. (2000), 'Glioblastoma multiforme: the terminator', Proceedings of the National Academy of Sciences **97**(12), 6242–6244.

Hoshida, Y., Brunet, J.-P., Tamayo, P., Golub, T. R. and Mesirov, J. P. (2007), 'Subclass mapping: identifying common subtypes in independent disease data sets', PloS one **2**(11), e1195.

Hotelling, H. (1933), 'Analysis of a complex of statistical variables into principal components.', Journal of educational psychology **24**(6), 417.

Hotelling, H. (1936), 'Relations between two sets of variables', Biometrika pp. 312–377.

Hripcsak, G. and Albers, D. J. (2013), 'Next-generation phenotyping of electronic health records', Journal of the American Medical Informatics Association **20**(1), 117–121.

Hsiao, C.-J., Hing, E., Socey, T. C. and Cai, B. (2011), 'Electronic health record systems and intent to apply for meaningful use incentives among office-based physician practices: United states, 2001–2011', system **18**(17.3), 17–3.

Huang, D. W., Sherman, B. T. and Lempicki, R. A. (2009), 'Systematic and integrative analysis of large gene lists using david bioinformatics resources', Nature protocols **4**(1), 44–57.

Huda, S. B. and Noureen, N. (2016), Mtbgd: Mutli type biclustering for genomic data, in 'Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on', IEEE, pp. 1113–1119.

Ideker, T., Thorsson, V., Ranish, J. A., Christmas, R., Buhler, J., Eng, J. K., Bumgarner, R., Goodlett, D. R., Aebersold, R. and Hood, L. (2001), 'Integrated genomic and proteomic analyses of a systematically perturbed metabolic network', Science **292**(5518), 929–934.

Ihmels, J., Bergmann, S. and Barkai, N. (2004), 'Defining transcription modules using large-scale gene expression data', Bioinformatics **20**(13), 1993–2003.

Illumina (2016), 'Infinium omniexpress-24 v1.2 beadchip'.
URL: *http://www.illumina.com/content/dam/illumina-marketing/documents/products/datasheets/datasheet_human_omni_express.pdf*

James, P. (1997), 'Protein identification in the post-genome era: the rapid rise of proteomics', Quarterly reviews of biophysics **30**(04), 279–331.

Jenatton, R., Audibert, J.-Y. and Bach, F. (2011), 'Structured variable selection with sparsity-inducing norms', The Journal of Machine Learning Research **12**, 2777–2824.

Jenatton, R., Obozinski, G. and Bach, F. (2009), 'Structured sparse principal component analysis', arXiv preprint arXiv:0909.1440 .

Jenatton, R., Obozinski, G. and Bach, F. R. (2010), Structured sparse principal component analysis., in 'AISTATS', pp. 366–373.

Jensen, P. B., Jensen, L. J. and Brunak, S. (2012), 'Mining electronic health records: towards better research applications and clinical care', Nature Reviews Genetics **13**(6), 395–405.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A. and Mark, R. G. (2016), 'Mimic-iii, a freely accessible critical care database', Scientific data **3**.

Jolliffe, I. T., Trendafilov, N. T. and Uddin, M. (2003a), 'A modified principal component technique based on the lasso', Journal of computational and Graphical Statistics **12**(3), 531–547.

Jolliffe, I. T., Trendafilov, N. T. and Uddin, M. (2003b), 'A modified principal component technique based on the lasso', Journal of computational and Graphical Statistics pp. 531–547.

Jones, G., Machado, J. and Merlo, A. (2001), 'Loss of focal adhesion kinase (fak) inhibits epidermal growth factor receptor-dependent migration and induces aggregation of nh2-terminal fak in the nuclei of apoptotic glioblastoma cells', Cancer research **61**(13), 4978–4981.

Kanehisa, M. and Goto, S. (2000), 'Kegg: kyoto encyclopedia of genes and genomes', Nucleic acids research **28**(1), 27–30.

Kawashima, E., Farinelli, L. and Mayer, P. (2005), 'Method of nucleic acid amplification'. US Patent App. 10/449,010.
URL: *https://www.google.com/patents/US20050100900*

Keshava Prasad, T., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., Telikicherla, D., Raju, R., Shafreen, B., Venugopal, A. et al. (2008), 'Human protein reference database2009 update', Nucleic acids research **37**(suppl_1), D767–D772.

Kim, S. and Xing, E. P. (2013), 'Statistical estimation of correlated genome associations to a quantitative trait network', PLOS Genetics **5**.

Knuth, D. E. (1968), 'Semantics of context-free languages', Mathematical Systems Theory **2**(2), 127–145.

Koboldt, D. (2014), 'Brace yourself for large-scale whole genome sequencing'.
URL: *http://massgenomics.org/2014/11/brace-yourself-for-large-scale-whole-genome-sequencing.html*

Lasko, T. A., Denny, J. C. and Levy, M. A. (2013), 'Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data', PloS one **8**(6), e66341.

Lazzeroni, L. and Owen, A. (2002), 'Plaid models for gene expression data', Statistica sinica pp. 61–86.

Le, Q. V. and Mikolov, T. (2014), Distributed representations of sentences and documents., in 'ICML', Vol. 14, pp. 1188–1196.

Lefranc, F., Brotchi, J. and Kiss, R. (2005), 'Possible future issues in the treatment of glioblastomas: special emphasis on cell migration and the resistance of migrating glioblastoma cells to apoptosis', Journal of clinical oncology **23**(10), 2411–2422.

Li, C. and Li, H. (2008a), 'Network-constrained regularization and variable selection for analysis of genomic data', Bioinformatics **24**(9), 1175–1182.

Li, C. and Li, H. (2008b), 'Network-constrained regularization and variable selection for analysis of genomic data', Bioinformatics **24**(9), 1175–1182.

Li, S., Park, Y., Duraisingham, S., Strobel, F. H., Khan, N., Soltow, Q. A., Jones, D. P. and Pulendran, B. (2013), 'Predicting network activity from high throughput metabolomics', PlOS Computational Biology **9**.

Li, Z., Safo, S. E. and Long, Q. (2017), 'Incorporating biological information in sparse principal component analysis with application to genomic data', BMC bioinformatics **18**(1), 332.

Lipinski, C. A., Tran, N. L., Bay, C., Kloss, J., McDonough, W. S., Beaudry, C., Berens, M. E. and Loftus, J. C. (2003), 'Differential role of proline-rich tyrosine kinase 2 and focal adhesion kinase in determining glioblastoma migration and proliferation1 1 national institutes of health grants hl67938 (jcl) and ns42262 (meb).', Molecular Cancer Research **1**(5), 323–332.

LISA−lab (2016), 'Deep learning tutorial'.
URL: *http://deeplearning.net/tutorial/*

Liu, C., Wang, F., Hu, J. and Xiong, H. (2015), Temporal phenotyping from longitudinal electronic health records: A graph based framework, in 'Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 705–714.

Liu, Y., Gu, Q., Hou, J. P., Han, J. and Ma, J. (2014), 'A network-assisted co-clustering algorithm to discover cancer subtypes based on gene expression', BMC bioinformatics **15**(1), 37.

MacKay, D. J. (2003), Information theory, inference and learning algorithms, Cambridge university press.

Madhavan, S., Zenklusen, J.-C., Kotliarov, Y., Sahni, H., Fine, H. A. and Buetow, K. (2009), 'Rembrandt: helping personalized medicine become a reality through integrative translational research', Molecular Cancer Research **7**(2), 157–167.

Mardia, K. V., Kent, J. T. and Bibby, J. M. (2003), Multivariate Analysis, Acadmeic Press.

Markoff, J. (2006), 'Taking spying to higher level, agencies look for more ways to mine data'.

URL: *http://www.nytimes.com/2006/02/25/technology/taking-spying-to-higher-level-agencies-look-for-more-ways-to.html*

Martín-Guerrero, J. D., Gomez, F., Soria-Olivas, E., Schmidhuber, J., Climente-Martí, M. and Jiménez-Torres, N. V. (2009), 'A reinforcement learning approach for individualizing erythropoietin dosages in hemodialysis patients', Expert Systems with Applications **36**(6), 9737–9742.

McLendon, R., Friedman, A., Bigner, D., Van Meir, E. G., Brat, D. J., Mastrogianakis, G. M., Olson, J. J., Mikkelsen, T., Lehman, N., Aldape, K. et al. (2008), 'Comprehensive genomic characterization defines human glioblastoma genes and core pathways', Nature **455**(7216), 1061–1068.

Mi, H., Poudel, S., Muruganujan, A., Casagrande, J. T. and Thomas, P. D. (2015), 'Panther version 10: expanded protein families and functions, and analysis tools', Nucleic acids research **44**(D1), D336–D342.

Mi, H., Poudel, S., Muruganujan, A., Casagrande, J. T. and Thomas, P. D. (2016), 'Panther version 10: expanded protein families and functions, and analysis tools', Nucleic acids research **44**(D1), D336–D342.

Mikolov, T. (2012), 'Statistical language models based on neural networks', Presentation at Google, Mountain View, 2nd April .

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013), 'Efficient estimation of word representations in vector space', arXiv preprint arXiv:1301.3781 .

Mikolov, T., Deoras, A., Povey, D., Burget, L. and Černockỳ, J. (2011), Strategies for training large scale neural network language models, in 'Automatic Speech

Recognition and Understanding (ASRU), 2011 IEEE Workshop on', IEEE, pp. 196–201.

Mikolov, T., Kombrink, S., Burget, L., Černockỳ, J. and Khudanpur, S. (2011), Extensions of recurrent neural network language model, in '2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 5528–5531.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J. (2013), Distributed representations of words and phrases and their compositionality, in 'Advances in neural information processing systems', pp. 3111–3119.

Misra, S. (2015), 'New report finds more than 165,000 mobile health apps now available, takes close look at characteristics & use'.
URL: *http://www.imedicalapps.com/2015/09/ims-health-apps-report/*

Mnih, A. and Kavukcuoglu, K. (2013), Learning word embeddings efficiently with noise-contrastive estimation, in 'Advances in Neural Information Processing Systems', pp. 2265–2273.

Mulkar-Mehta, R. (2015a), 'Online word2vec for gensim'.
URL: *http://rutumulkar.com/blog/2015/word2vec*

Mulkar-Mehta, R. (2015b), 'Online word2vec for gensim'.
URL: *http://rutumulkar.com/blog/2015/word2vec/*

Murali, T. and Kasif, S. (2002), Extracting conserved gene expression motifs from gene expression data, in 'Biocomputing 2003', World Scientific, pp. 77–88.

NCHS (2009), 'Health, united states'.

Network, C. G. A. R. et al. (2011), 'Integrated genomic analyses of ovarian carcinoma', Nature **474**(7353), 609–615.

Padilha, V. A. and Campello, R. J. (2017), 'A systematic comparative evaluation of biclustering techniques', BMC bioinformatics **18**(1), 55.

Pan, W., Xie, B. and Shen, X. (2010a), 'Incorporating predictor network in penalized regression', Biometrics **66**(2), 474–484.

Pan, W., Xie, B. and Shen, X. (2010b), 'Incorporating predictor network in penalized regression with application to microarray data', Biometrics **66**(2), 474–484.

Park, M.-J., Kim, M.-S., Park, I.-C., Kang, H.-S., Yoo, H., Park, S. H., Rhee, C. H., Hong, S.-I. and Lee, S.-H. (2002), 'Pten suppresses hyaluronic acid-induced matrix metalloproteinase-9 expression in u87mg glioblastoma cells through focal adhesion kinase dephosphorylation', Cancer research **62**(21), 6318–6322.

Parkhomenko, E., Tritchler, D. and Beyene, J. (2009), 'Sparse canonical correlation analysis with application to genomic data integration', Statistical Applications in Genetics and Molecular Biology **8**.

Patrikainen, A. and Meila, M. (2006), 'Comparing subspace clusterings', IEEE Transactions on Knowledge and Data Engineering **18**(7), 902–916.

Peason, K. (1901), 'On lines and planes of closest fit to systems of point in space', Philosophical Magazine **2**, 559–572.

Pen, A., Moreno, M. J., Martin, J. and Stanimirovic, D. B. (2007), 'Molecular markers of extracellular matrix remodeling in glioblastoma vessels: Microarray study of laser-captured glioblastoma vessels', Glia **55**(6), 559–572.

Peri, S., Navarro, J. D., Amanchy, R., Kristiansen, T. Z., Jonnalagadda, C. K., Surendranath, V., Niranjan, V., Muthusamy, B., Gandhi, T., Gronborg, M. et al. (2003a), 'Development of human protein reference database as an initial platform for approaching systems biology in humans', Genome research **13**(10), 2363–2371.

Peri, S., Navarro, J. D., Amanchy, R., Kristiansen, T. Z., Jonnalagadda, C. K., Surendranath, V., Niranjan, V., Muthusamy, B., Gandhi, T., Gronborg, M. et al. (2003b), 'Development of human protein reference database as an initial platform for approaching systems biology in humans', Genome research **13**(10), 2363–2371.

Polson, N. G., Scott, J. G. and Windle, J. (2013), 'Bayesian inference for logistic models using pólya–gamma latent variables', Journal of the American statistical Association **108**(504), 1339–1349.

Pontes, B., Giráldez, R. and Aguilar-Ruiz, J. S. (2015), 'Biclustering on expression data: A review', Journal of biomedical informatics **57**, 163–180.

Prasad, T. K., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., Telikicherla, D., Raju, R., Shafreen, B., Venugopal, A. et al. (2009), 'Human protein reference database2009 update', Nucleic acids research **37**(suppl 1), D767–D772.

Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L. and Zitzler, E. (2006), 'A systematic comparison and evaluation of biclustering methods for gene expression data', Bioinformatics **22**(9), 1122–1129.

Ribeiro, L. F., Saverese, P. H. and Figueiredo, D. R. (2017), struc2vec: Learning node representations from structural identity, in 'Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 385–394.

Riesz, F. (1910), 'Untersuchungen über systeme integrierbarer funktionen', Mathematische Annalen **69**(4), 449–497.

Ritz, B. and Yu, F. (2000), 'Parkinsons disease mortality and pesticide exposure in california from 1984-1994', International Journal of Epidemiology **29**(2), 323–329.

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

Rodriguez-Baena, D. S., Perez-Pulido, A. J. and Aguilar-Ruiz, J. S. (2011), 'A biclustering algorithm for extracting bit-patterns from binary datasets', Bioinformatics **27**(19), 2738–2745.

Roede, J. R., Uppal, K., Park, Y., Tran, V. and Jones, D. P. (2014), 'Trancriptome-metabolome wide association study (tmwas) of maneb and paraquet neurotoxicity reveals network level interactions in toxiocologic mechanism', Toxicology Reports **1**, 435–444.

Rong, X. (2014), 'word2vec parameter learning explained', arXiv preprint arXiv:1411.2738 .

Rosenwald, A., Wright, G., Chan, W. C., Connors, J. M., Campo, E., Fisher, R. I., Gascoyne, R. D., Muller-Hermelink, H. K., Smeland, E. B., Giltnane, J. M. et al. (2002), 'The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma', New England Journal of Medicine **346**(25), 1937–1947.

Safo, S. and Ahn, J. (2014), Sparse Analysis for High Dimensional Data, PhD thesis, University of Georgia.

Safo, S. E., Li, S. and Long, Q. (2017), 'Integrative analysis of transcriptomic and metabolomic data via sparse canonical correlation analysis with incorporation of biological information', Biometrics .

Sanger, F. (1981), 'Determination of nucleotide sequences in dna', Bioscience reports **1**(1), 3–18.

Schulz, S. and Martínez-Costa, C. (2013), How ontologies can improve semantic interoperability in health care., in 'KR4HC/ProHealth', Springer, pp. 1–10.

Serin, A. and Vingron, M. (2011), 'Debi: Discovering differentially expressed biclusters using a frequent itemset approach', Algorithms for Molecular Biology **6**(1), 18.

Sheng, Q., Moreau, Y. and De Moor, B. (2003), 'Biclustering microarray data by gibbs sampling', Bioinformatics **19**(suppl 2), ii196–ii205.

Silver, D. and Hassabis, D. (2016), 'Alphago: Mastering the ancient game of go with machine learning'.

URL: *https://research.googleblog.com/2016/01/alphago-mastering-ancient-game-of-go.html*

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. et al. (2016), 'Mastering the game of go with deep neural networks and tree search', Nature **529**(7587), 484–489.

Simard, P. Y., Steinkraus, D. and Platt, J. C. (2003), Best practices for convolutional neural networks applied to visual document analysis., in 'ICDAR', Vol. 3, pp. 958–962.

Sparkman, O. D. (2000), 'Mass spectrometry desk reference', Journal of the American Society for Mass Spectrometry **11**(12), 1144.

Stang, P. E., Ryan, P. B., Racoosin, J. A., Overhage, J. M., Hartzema, A. G., Reich, C., Welebob, E., Scarnecchia, T. and Woodcock, J. (2010), 'Advancing the science for active surveillance: rationale and design for the observational medical outcomes partnership', Annals of internal medicine **153**(9), 600–606.

Stanley, K. O., Bryant, B. D. and Miikkulainen, R. (2005), 'Evolving neural network agents in the nero video game', Proceedings of the IEEE pp. 182–189.

Steinbach, M., Karypis, G., Kumar, V. et al. (2000), A comparison of document clustering techniques, in 'KDD workshop on text mining', Vol. 400, Boston, pp. 525–526.

Su, A. I., Cooke, M. P., Ching, K. A., Hakak, Y., Walker, J. R., Wiltshire, T., Orth, A. P., Vega, R. G., Sapinoso, L. M., Moqrich, A. et al. (2002), 'Large-scale analysis of the human and mouse transcriptomes', Proceedings of the National Academy of Sciences **99**(7), 4465–4470.

Tanay, A., Sharan, R., Kupiec, M. and Shamir, R. (2004), 'Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data', Proceedings of the National Academy of Sciences of the United States of America **101**(9), 2981–2986.

Tanay, A., Sharan, R. and Shamir, R. (2002), 'Discovering statistically significant biclusters in gene expression data', Bioinformatics **18**(suppl_1), S136–S144.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J. and Mei, Q. (2015), Line: Large-scale information network embedding, in 'Proceedings of the 24th International Conference on World Wide Web', International World Wide Web Conferences Steering Committee, pp. 1067–1077.

Tenenbaum, D. (2013), 'Keggrest: Client-side rest access to kegg', R package version **1**(1).

Tian, L., Fan, C., Ming, Y. and Jin, Y. (2015), Stacked pca network (spcanet): an effective deep learning for face recognition, in '2015 IEEE International Conference on Digital Signal Processing (DSP)', IEEE, pp. 1039–1043.

Tibshirani, R. (1994), 'Regression shrinkage and selection via the lasso', Journal of the Royal Statistical Society, Series B **58**, 267–288.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005), 'Sparsity and smoothness via the fused lasso', Journal of the Royal Statistical Society: Series B (Statistical Methodology) **67**(1), 91–108.

Torgan, C. (2009), 'The mhealth summit: Local & global converge'.
URL: *http://caroltorgan.com/mhealth-summit/*

USDHHS (2006), Health, United States, 2005: With chartbook on trends in the health of Americans, Claitor's Law Books and Publishing Division.

Van't Veer, L. J., Dai, H., Van De Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., Van Der Kooy, K., Marton, M. J., Witteveen, A. T. et al. (2002), 'Gene expression profiling predicts clinical outcome of breast cancer', nature **415**(6871), 530–536.

Verhaak, R. G., Hoadley, K. A., Purdom, E., Wang, V., Qi, Y., Wilkerson, M. D., Miller, C. R., Ding, L., Golub, T., Mesirov, J. P. et al. (2010), 'Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in pdgfra, idh1, egfr, and nf1', Cancer cell **17**(1), 98–110.

Vinod, H. D. (1970), 'Canonical ridge and econometrics of joint production', Journal of Econometrics pp. 147–166.

Waaijenborg, S., de Witt Hamar, P. C. V. and Zwinderman, A. H. (2008), 'Quantifying the association between gene expressions and dna-markers by penalized

canonical correlation analysis', Statistical Applications in Genetics and Molecular Biology **7**.

Wang, A., Costello, S., Cockburn, M., Zhang, X., Bronstein, J. and Ritz, B. (2011), 'Parkinsons disease risk from ambient exposure to pesticides', European Journal of Epidemiology **26**(7), 547–555.

Wang, K., Li, M. and Bucan, M. (2007), 'Pathway-based approaches for analysis of genomewide association studies', The American Journal of Human Genetics **81**(6), 1278–1283.

Wang, W., Baladandayuthapani, V., Morris, J. S., Broom, B. M., Manyam, G. and Do, K.-A. (2012), 'ibag: integrative bayesian analysis of high-dimensional multiplatform genomics data', Bioinformatics **29**(2), 149–159.

Watson, J. D. and Crick, F. H. (1953), The structure of dna, in 'Cold Spring Harbor symposia on quantitative biology', Vol. 18, Cold Spring Harbor Laboratory Press, pp. 123–131.

Weigel, V. B. (2002), Deep Learning for a Digital Age: Technology's Untapped Potential To Enrich Higher Education., ERIC.

WhiteHouse (2015), 'Fact sheet: President obamas precision medicine initiative'.
URL: *https://www.whitehouse.gov/the-press-office/2015/01/30/fact-sheet-president-obama-s-precision-medicine-initiative*

Wikipedia (2016a), 'mhealth'.
URL: *https://en.wikipedia.org/wiki/MHealth#cite_note−caroltorgan*1 − 10

Wikipedia (2016b), 'Omics'.
URL: *https://en.wikipedia.org/wiki/Omics*

Witten, D. M., Tibshirani, R. and Hastie, T. (2009a), 'A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis', Biostatistics p. kxp008.

Witten, D. M. and Tibshirani, R. J. (2009), 'Extensions of sparse canonical correlation analysis with applications to genomic data', Statistical Applications in Genetics and Molecular Biology **8**.

Witten, D. M., Tibshirani, R. J. and Hastie, T. (2009b), 'A penalized matrix decomposition, with applications to sparse prinicial components and canonical correlation analysis', Biostatistics **10**(3), 515–534.

Witten, D., Tibshirani, R., Gross, S. and Narasimhan, B. (2013), 'Package 'pma'', http://cran.r-project.org/web/packages/PMA/PMA.pdf. Version 1.0.9.

Wu, J., Roy, J. and Stewart, W. F. (2010), 'Prediction modeling using ehr data: challenges, strategies, and a comparison of machine learning approaches', Medical care **48**(6), S106–S113.

Xiao, Y., Xiong, L., Fan, L. and Goryczka, S. (2012), 'Dpcube: differentially private histogram release through multidimensional partitioning', arXiv preprint arXiv:1202.5358 .

Yu, G., Yu, X. and Wang, J. (2017), 'Network-aided bi-clustering for discovering cancer subtypes', Scientific Reports **7**(1), 1046.

Zhang, J. D. and Wiemann, S. (2009), 'Kegggraph: a graph approach to kegg pathway in r and bioconductor', Bioinformatics **25**(11), 1470–1471.

Zhao, Y., Chung, M., Johnson, B. A., Moreno, C. S. and Long, Q. (2016), 'Hierarchical feature selection incorporating known and novel biological information:

Identifying genomic features related to prostate cancer recurrence', Journal of the American Statistical Association (in press).

Zhao, Y., Zeng, D., Socinski, M. A. and Kosorok, M. R. (2011), 'Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer', Biometrics **67**(4), 1422–1433.

Zissis, D., Xidias, E. K. and Lekkas, D. (2015), 'A cloud based architecture capable of perceiving and predicting multiple vessel behaviour', Applied Soft Computing **35**, 652–661.

Zou, H. and Hastie, T. (2005a), 'Regularization and variable selection via the elastic net', Journal of the Royal Statistical Society: Series B (Statistical Methodology) **67**(2), 301–320.

Zou, H. and Hastie, T. (2005b), 'Regularization and variable selection via the elastic net', Journal of the Royal Statistical Society, Series B **67**, 301–320.

Zou, H., Hastie, T. and Tibshirani, R. (2006), 'Sparse principal component analysis', Journal of Computational and Graphical Statistics pp. 265–286.