TOWARD ROBUST GROUP-WISE EQTL MAPPING VIA INTEGRATING MULTI-DOMAIN HETEROGENEOUS DATA

Wei Cheng

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2015

Approved by:

Wei Wang

Wei Sun

Marc Niethammer

Leonard McMillan

Patrick Sullivan

Xiang Zhang

ABSTRACT

Wei Cheng: Toward Robust Group-Wise eQTL Mapping via Integrating Multi-Domain Heterogeneous Data
(Under the direction of Wei Wang)

As a promising tool for dissecting the genetic basis of common diseases, expression quantitative trait loci (eQTL) study has attracted increasing research interest. Traditional eQTL methods focus on testing the associations between individual single-nucleotide polymorphisms (SNPs) and gene expression traits. A major drawback of this approach is that it cannot model the joint effect of a set of SNPs on a set of genes, which may correspond to biological pathways. This thesis studies the problem of identifying group-wise associations in eQTL mapping. Based on the intuition of group-wise association, we examine how the integration of heterogeneous prior knowledge on the correlation structures between SNPs, and between genes can improve the robustness and the interpretability of eQTL mapping. To obtain a more accurate knowledgebase on the interactions among SNPs and genes, we developed a robust and flexible approach that can incorporate multiple data sources and automatically identify noisy sources. Extensive experiments demonstrate the effectiveness of the proposed algorithms.

To my family and advisor, I couldn't have done this without you.
Thank you for all of your support along the way.

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratefulness to my advisor, Dr. Wei Wang, for her continuous guidance and support. I feel especially fortunate to have worked closely with Dr. Patrick Sullivan and Dr. Xiang Zhang, who encouraged me to work persistently and greatly helped me to improve my critical thinking ability and writing skill. Special thanks go to Dr. Leonard McMillan who chaired my committee and provided assistance to me. I would also like to thank Dr. Wei Sun and Dr. Marc Niethammer, who served on my committee and devoted a lot of effort to my study.

My special thanks also go to members of CompGen Lab, including Eric Yi Liu, Zhaojun Zhang, Shunping Huang, Weibo Wang, and others, for their thoughtful discussions on the problems in the research. I would like to thank all fellow persons I met in the Computer Science department who provided all kinds of help to me during my entire pursuit of PhD. I would also like to thank all the warm-hearted persons I met in the past few years for helping me to live in Chapel Hill, the beautiful town in North Carolina.

Finally, I am also deeply thankful to my wife and my parents. They stand steadily after me and encourage me to overcome the difficulties I have faced in my pursuit of PhD. Without their endless support, I have absolutely no chance to finish this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

The most abundant sources of genetic variations in modern organisms are single nucleotide polymorphisms (SNPs). A SNP is a DNA sequence variation occurring when a single nucleotide (A, T, G, or C) in the genome differs between individuals of a species. For inbred diploid organisms, such as inbred mice, a SNP usually shows variation between only two of the four possible nucleotide types (Ideraabdullah et al., 2004), which allows us to represent it by a binary variable. The binary representation of a SNP is also referred to as the *genotype* of the SNP. The genotype of an organism is the genetic code in its cells. This genetic constitution of an individual influences, but is not solely responsible for, many of its traits. A *phenotype* is an observable trait or characteristic of an individual. The phenotype is the visible, or expressed trait, such as hair color. The phenotype depends upon the genotype but can also be influenced by environmental factors. Phenotypes can be either quantitative or binary.

Driven by the advancement of cost-effective and high-throughput genotyping technologies, genome-wide association studies (GWAS) have revolutionized the field of genetics by providing new ways to identify genetic factors that influence phenotypic traits. Typically, GWAS focus on associations between SNPs and traits like major diseases. As an important subsequent analysis, quantitative trait locus (QTL) analysis is aiming at to detect the associations between two types of information–quantitative phenotypic data (trait measurements) and genotypic data (usually SNPs)–in an attempt to explain the genetic basis of variation in complex traits. QTL analysis allows researchers in fields as diverse as agriculture, evolution, and medicine to link certain complex phenotypes to specific regions of chromosomes.

Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product, such as proteins. It is the most fundamental level at which the genotype gives rise to the phenotype. Gene expression profile is the quantitative measurement of

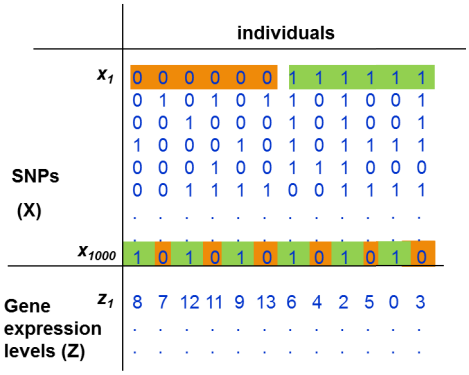


Figure 1.1: An example dataset in eQTL mapping

the activity of thousands of genes at once. The gene expression levels can be represented by continuous variables. Figure 1.1 shows an example dataset consisting of 1000 SNPs $\{x_1, x_2, \cdots, x_{1000}\}$ and a gene expression level z_1 for 12 individuals.

1.1 eQTL Mapping

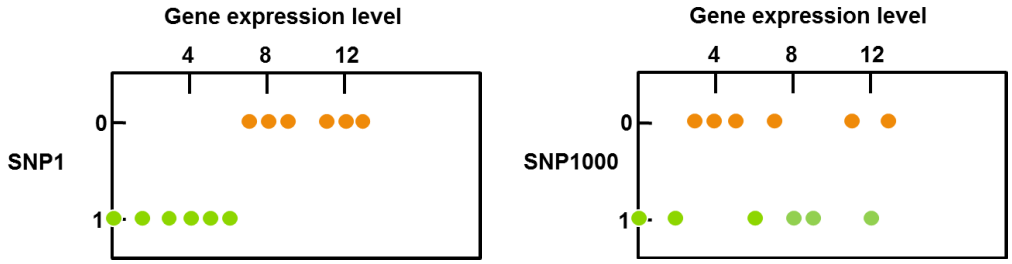
For a QTL analysis, if the phenotype to be analyzed is the gene expression level data, then the analysis is referred to as the expression quantitative trait loci (eQTL) mapping. It aims to identify SNPs that influence the expression level of genes. It has been widely applied to dissect the genetic basis of gene expression and molecular mechanisms underlying complex traits (Bochner, 2003; Rockman and Kruglyak, 2006; Michaelson et al., 2009a). More formally, let $\mathbf{X} = \{\mathbf{x}_d | 1 \leq d \leq D\} \in \mathbb{R}^{K \times D}$ be the SNP matrix denoting genotypes of K SNPs of D individuals and $\mathbf{Z} = \{\mathbf{z}_d | 1 \leq d \leq D\} \in \mathbb{R}^{N \times D}$ be the gene expression matrix denoting phenotypes of N gene expression levels of the same set of D individuals. Each column of $\mathbf{X}$ and $\mathbf{Z}$ stands for one individual. The goal of eQTL mapping is to find SNPs in $\mathbf{X}$, that are highly associated with genes in $\mathbf{Z}$.

Various statistics, such as the ANOVA (analysis of variance) test and the chi-square test, can be applied to measure the association between SNPs and the gene expression level of interest. Sparse feature selection methods, e.g., Lasso (Tibshirani, 1996), are also widely used for eQTL mapping problems. Here, we take Lasso as an example. Lasso is a method for estimating the

regression coefficients $\mathbf{W}$ using ℓ_1 penalty. The objective function of Lasso is

$$\min_{\mathbf{W}} \frac{1}{2}||\mathbf{Z} - \mathbf{W}\mathbf{X}||_F^2 + \eta||\mathbf{W}||_1 \tag{1.1}$$

where $||\cdot||_F$ denotes the Frobenius norm, $||\cdot||_1$ is the ℓ_1-norm. η is the empirical parameter for the ℓ_1 penalty. $\mathbf{W}$ is the parameter (also called weight) matrix setting the limits for the space of linear functions mapping from $\mathbf{X}$ to $\mathbf{Z}$. Each element of $\mathbf{W}$ is the effect size of corresponding SNP and expression level. Lasso uses the least squares method with ℓ_1 penalty. ℓ_1-norm sets many non-significant elements of $\mathbf{W}$ to be exactly zero, since many SNPs have no associations to a given gene. Lasso works even when the number of SNPs is significantly larger than the sample size ($K \gg D$) under the sparsity assumption.



(a) Strong association (b) No association

Figure 1.2: Examples of associations between a gene expression level and two different SNPs

Using the dataset shown in Figure 1.1, Figure 1.2 (a) shows an example of strong association between gene expression z_1 and SNP x_1. 0 and 1 on the y-axis represent the binary SNP genotype and the x-axis represents the gene expression level. Each point in the figure represents an individual. It is clear from the figure that the gene expression level values are partitioned into two groups with distinct means, hence indicating a strong association between the gene expression and the SNP. On the other hand, if the genotype of a SNP partitions the gene expression level values into groups as shown in Figure 1.2 (b), the gene expression and the SNP are not associated with each other. An illustration result of Lasso is shown in Figure 1.3. $\mathbf{W}_{ij} = 0$ means no association between j-th SNP and i-th gene expression. $\mathbf{W}_{ij} \neq 0$ means there exists an association between the j-th SNP and the i-th gene expression.

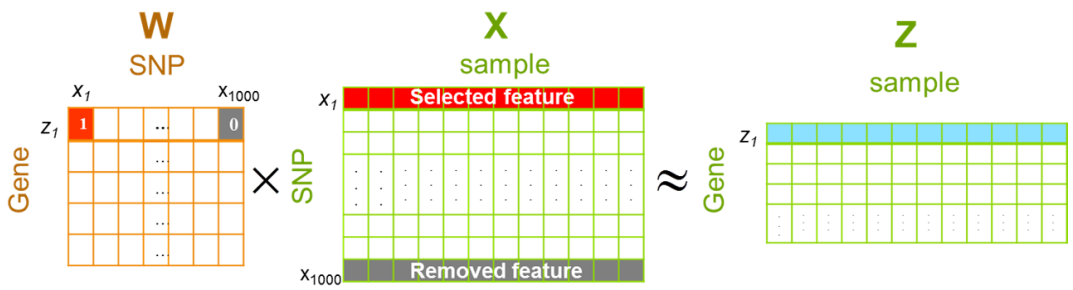


Figure 1.3: Association weights estimated by Lasso on the example data

1.2 Group-Wise eQTL Mapping and Challenges

In a typical eQTL study, the association between each expression trait and each SNP is assessed separately (Cheung et al., 2005; Zhu et al., 2008; Tibshirani, 1996). This approach does not consider the interactions among SNPs and among genes. However, multiple SNPs may jointly influence the phenotypes (Lander, 2011), and genes in the same biological pathway are often co-regulated and may share a common genetic basis (Musani et al., 2007b; Pujana et al., 2007).

To better elucidate the genetic basis of gene expression, it is highly desirable to develop efficient methods that can automatically infer associations between a group of SNPs and a group of genes. We refer to the process of identifying such associations as *group-wise* eQTL mapping. In contrast, we refer to those associations between individual SNPs and individual genes as *individual* eQTL mapping. An example is shown in Figure 1.4. Note that an ideal model should allow overlaps between SNP sets and between gene sets; that is, a SNP or gene may participate in multiple individual and group-wise associations. This is because genes and the SNPs influencing them may play different roles in multiple biological pathways (Lander, 2011).

Besides, advanced bio-techniques are generating a large volume of heterogeneous datasets, such as protein-protein interaction (PPI) networks (Asur et al., 2007), and genetic interaction networks (Cordell, 2009). These datasets describe the partial relationships between SNPs and relationships between genes. Because SNPs and genes are not independent of each other, and there exist group-wise associations, the integration of these multi-domain heterogeneous data sets is able to improve the accuracy of eQTL mapping since more domain

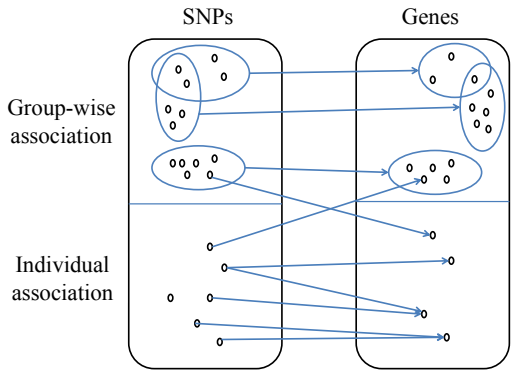


Figure 1.4: An illustration of individual and group-wise associations.

knowledge can be integrated. In literature, several methods based on Lasso have been proposed (Biganzoli et al., 2006; Kim and Xing, 2012; Lee and Xing, 2012; Lee et al., 2010) to leverage the network prior knowledge (Biganzoli et al., 2006; Kim and Xing, 2012; Lee et al., 2010; Lee and Xing, 2012; Jenatton et al., 2011). However, these methods suffer from poor quality or incompleteness of this prior knowledge.

In summary, there are several issues that greatly limit the applicability of current eQTL mapping approaches.

1. It is a crucial challenge to understand *how multiple, modestly-associated SNPs interact to influence the phenotypes* (Lander, 2011). However, little prior work has studied the group-wise eQTL mapping problem.
2. The prior knowledge about the relationships between SNPs and between genes is often partial and usually includes noise.
3. Confounding factors such as expression heterogeneity may result in spurious associations and mask real signals (Michaelson et al., 2009b; Stegle et al., 2008; Gilad et al., 2008).

1.3 Thesis Statement

This thesis systematically studies the group-wise eQTL mapping problem and determines that effective algorithms can be designed for group-wise eQTL mapping. Extensive experimental results demonstrate that the algorithms proposed in this dissertation are able to integrate

multi-domain heterogeneous data and can effectively detect group-wise associations for eQTL mapping.

1.4 Overview of the Developed Algorithms

This thesis proposes and studies the problem of group-wise eQTL mapping. We can decouple the problem into the following sub-problems.

- How can we detect group-wise eQTL associations with eQTL data only, i.e., with SNPs and gene expression profile data?
- How can we prepare more accurate prior knowledge about the relationships between SNPs and between genes by integrating multi-domain heterogeneous data?
- How can we incorporate the prior interaction structures between SNPs and between genes into eQTL mapping to improve the robustness of the model and the interpretability of the results?

To address the first sub-problem, the thesis proposes three approaches based on sparse linear-Gaussian graphical models to infer novel associations between SNP sets and gene sets. In literature, many efforts have focused on single-locus eQTL mapping. However, a multi-locus study dramatically increases the computation burden. The existing algorithms cannot be applied on a genome-wide scale. In order to accurately capture possible interactions between multiple genetic factors and their joint contribution to a group of phenotypic variations, we propose three algorithms. The first algorithm, SET-eQTL, makes use of a three-layer sparse linear-Gaussian model. The upper layer nodes correspond to the set of SNPs in the study. The middle layer consists of a set of hidden variables. The hidden variables are used to model both the joint effect of a set of SNPs and the effect of confounding factors. The lower layer nodes correspond to the genes in the study. The nodes in different layers are connected via arcs. SET-eQTL can help unravel true functional components in existing pathways. The results could provide new insights on how genes act and coordinate with each other to achieve certain biological functions. We further extend the approach to be able to consider confounding factors and decouple *individual* associations and *group-wise* associations for eQTL mapping.

For the second sub-problem, this thesis presents a flexible and robust algorithm, CGC, to integrate heterogeneous graph data for clustering. Graphs (also called networks, but for the purpose of this thesis, we will maintain consistency by using the term "graphs".) are widely used in representing relationships between instances, in which each node corresponds to an instance and each edge depicts the relationship between a pair of instances. Much prior knowledge about the relationships between SNPs and relationships between genes can be modeled as graphs. Biologists believe that a set of SNPs may play joint roles in a disease. Such interactions between SNPs can be modeled by a SNP interaction network. Even though the underlying biological processes are complex and only partially understood, it is well established that SNPs may alter the expression levels of related genes which may in turn have a cascading effect on other genes, e.g., in the same biological pathways (Michaelson et al., 2009c). The interactions between genes can be measured by correlations of gene expressions and represented by a gene interaction network. These two networks are heavily related because of the complicated relationships between SNPs and genes, as demonstrated in many expression quantitative trait loci (eQTL) studies (Lee and Xing, 2012). It is evident that a joint analysis becomes essential in these related domains. Multiple domain data, such as SNP-SNP interaction network, PPI network, and gene co-expression network, are able to provide more accurate prior knowledge about the grouping information of SNPs and genes. Data collected from different sources provide complimentary predictive powers, and combining their information can resolve ambiguity, thus helping to obtain a more accurate knowledge base. This thesis investigates the problem of clustering multiple heterogeneous data sets, where the cross-domain instance relationship is "*many-to-many*". This problem has a wide range of applications and poses new technical challenges that cannot be directly tackled by traditional "*multi-view*" graph clustering methods (Kumar et al., 2011; Chaudhuri et al., 2009; Kumar and III, 2011). Based on the clustering consensus for different domains, we developed a robust and flexible approach that can incorporate multiple sources to enhance graph clustering performance. The proposed approach is robust even when the cross-domain relationships based on prior knowledge are noisy. Besides, the model provides

users with the extent to which the cross-domain instance relationship violates the in-domain clustering structure, and thus enables users to re-evaluate the consistency of the relationship. The thesis further studies the trustworthiness of multi-source data, and extends the approach to enable it to automatically identify noisy domains and assign smaller weights to them for integration.

To address the third sub-problem, this thesis presents an algorithm, Graph-regularized Dual Lasso (GDL), to simultaneously learn the association between SNPs and genes and refine the prior networks. Traditional sparse regression problems in data mining and machine learning consider both predictor variables and response variables individually, such as sparse feature selection using Lasso. In the eQTL mapping application, both predictor variables and response variables are not independent of each other, and we may be interested in the joint effects of multiple predictors to a group of response variables. In some cases, we may have partial prior knowledge, such as the correlation structures between predictors, and correlation structures between response variables. This thesis shows how prior graph information would help improve eQTL mapping accuracy and how refinement of prior knowledge would further improve the mapping accuracy. In addition, other different types of prior knowledge, *e.g.*, location information of SNPs and genes, as well as pathway information, can also be integrated for the graph refinement.

1.5 Thesis Outline

The thesis is organized as follows:

- The algorithms to detect group-wise eQTL associations with eQTL data only (SET-eQTL, etc.) are presented in Chapter 2.
- The algorithm (CGC) to integrate heterogenous graph data for clustering is presented in Chapter 3.
- The algorithm (GDL) to incorporate the prior interaction structures or grouping information of SNPs or genes into eQTL mapping is presented in Chapter 4.
- Chapter 5 concludes the thesis work.

CHAPTER 2: GROUP-WISE EQTL MAPPING

2.1 Introduction

A biological pathway is a series of actions among molecules in a cell that leads to a certain product or a change in a cell. For example, a pathway can trigger the assembly of new molecules, such as a fat or protein. Pathways play a key role in advanced studies of Genomics. In genetics, genes in the same biological pathway are often co-regulated and may share a common genetic basis (Musani et al., 2007b; Pujana et al., 2007). Consequently, it is crucial to understand how multiple modestly associated SNPs interact to influence the phenotypes (Lander, 2011). To address this issue, several approaches have been proposed to study the joint effect of multiple SNPs by testing the association between a set of SNPs and a gene expression trait. A straightforward approach is to follow the gene set enrichment analysis (GESA) (Holden et al., 2008). Wu et al. proposed variance component models for SNP set testing (Wu et al., 2011). Aggregation-based approaches such as collapsing SNPs are investigated (Braun and Buetow, 2011). Listgarten et al. took confounding factors into consideration (Listgarten et al., 2013).

Despite their successes, these methods have two common limitations. First, they only study the association between a set of SNPs and a single expression trait, thus overlooking the joint effect of a set of SNPs on the activities of a set of genes, which may act and interact with each other to achieve certain biological function. Second, the SNP sets used in these methods are usually taken from known pathways. However, the existing knowledge on biological pathways is far from being complete. These methods cannot identify unknown associations between SNP sets or gene sets.

To address these limitations, a method is developed to identify cliques in a bipartite graph derived from the eQTL data (Huang et al., 2009b). Cliques are used to model the hidden

correlations between SNP sets and gene sets. However, this method needs the progeny strain information, which is used as a bridge for modeling the eQTL association graphs. A two-graph-guided multi-task Lasso approach was developed in (Chen et al., 2012). This method needs to calculate gene co-expression network and SNP correlation network first. Errors and noises in these two networks may introduce bias in the final results. Note that all these methods do not consider confounding factors.

To better elucidate the genetic basis of gene expression and understand the underlying biology pathways, it is desirable to develop methods that can automatically infer associations between a group of SNPs and a group of genes. We refer to the process of identifying such associations as *group-wise* eQTL mapping. In contrast, we refer to the process of identifying associations between individual SNPs and genes as *individual* eQTL mapping. In this chapter, we propose several algorithms to detect group-wise associations. The first algorithm, SET-eQTL, makes use of a three-layer sparse linear-Gaussian model. It is able to identify novel associations between sets of SNPs and sets of genes. The results could provide new insights on how genes act and coordinate with each other to achieve certain biological functions. We further propose a fast and robust approach that is able to consider confounding factors and decouple *individual* associations and *group-wise* associations for eQTL mapping. The model is a multi-layer linear-Gaussian model and uses two different types of hidden variables: one capturing group-wise associations and the other capturing confounding factors (Gao et al., 2013; Leek and Storey, 2007; Joo et al., 2014; Fusi et al., 2012; Listgarten et al., 2013; Carlos M. Carvalhoa and West, 2008). We apply an ℓ_1-norm on the parameters (Lee et al., 2009; Tibshirani, 1996), which yields a sparse network with a large number of association weights being zero (Ng, 2004). We develop an efficient optimization procedure that makes this approach suitable for large-scale studies. Extensive experimental evaluations using both simulated and real datasets demonstrate that the proposed methods can effectively capture both group-wise and individual associations and significantly outperforms the state-of-the-art eQTL mapping methods.

2.2 Related Work

Recently, various analytic methods have been developed to address the limitations of the traditional single-locus approach. Epistasis detection methods aim to find the interaction between SNP-pairs (Hoh and Ott, 2003; Hirschhorn and Daly, 2005; Balding, 2006; Musani et al., 2007a). The computational burden of epistasis detection is usually very high due to the large number of interactions that need to be examined (Nelson et al., 2001; Ritchie et al., 2001). Filtering-based approaches (Evans et al., 2006; Hoh et al., 2000; Yang et al., 2009), which reduce the search space by selecting a small subset of SNPs for interaction study, may miss important interactions in the SNPs that have been filtered out.

Statistical graphical models and Lasso-based methods (Tibshirani, 1996) have been applied to eQTL study. A tree-guided group lasso has been proposed in (Kim and Xing, 2012). This method directly combines statistical strength across multiple related genes in gene expression data to identify SNPs with pleiotropic effects by leveraging the hierarchical clustering tree over genes. Bayesian methods have also been developed (Leopold Parts1, 2011; Stegle et al., 2010). Confounding factors may greatly affect the results of the eQTL study. To model confounders, a two-step approach can be applied (Stegle et al., 2010; Jeffrey T. Leek, 2007). These methods first learn the confounders that may exhibit broad effects to the gene expression traits. The learned confounders are then used as covariates in the subsequent analysis. Statistical models that incorporate confounders have been proposed (Nicolo Fusi and Lawrence, 2012). However, none of these methods are specifically designed to find novel associations between SNP sets and gene sets.

Pathway analysis methods have been developed to aggregate the association signals by considering a set of SNPs together (Cantor et al., 2010; Elbers et al., 2009; Torkamani et al., 2008; Perry et al., 2009). A pathway consists of a set of genes that coordinate to achieve a specific cell function. This approach studies a set of known pathways to find the ones that are highly associated with the phenotype (Wang et al., 2010). Although appealing, this approach is

limited to the priori knowledge on the predefined gene sets/pathways. On the other hand, the current knowledgebase on the biological pathways is still far from being complete.

A method is proposed to identify eQTL association cliques that expose the hidden structure of genotype and expression data (Huang et al., 2009b). By using the cliques identified, this method can filter out SNP-gene pairs that are unlikely to have significant associations. It models the SNP, progeny and gene expression data as an eQTL association graph, and thus depends on the availability of the progeny strain data as a bridge for modeling the eQTL association graph.

2.3 The Problem

Symbols	Description
K	number of SNPs
N	number of genes
D	number of samples
M	number of group-wise associations
H	number of confounding factors
$\mathbf{x}$	random variables of K SNPs
$\mathbf{z}$	random variables of N genes
$\mathbf{y}$	latent variables to model group-wise associaiton
$\mathbf{X} \in \mathbb{R}^{K \times H}$	SNP matrix data
$\mathbf{Z} \in \mathbb{R}^{N \times H}$	gene expression matrix data
$\mathbf{A} \in \mathbb{R}^{M \times K}$	group-wise association coefficient matrix between $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{B} \in \mathbb{R}^{N \times M}$	group-wise association coefficient matrix between $\mathbf{y}$ and $\mathbf{z}$
$\mathbf{C} \in \mathbb{R}^{N \times K}$	individual association coefficient matrix between $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{P} \in \mathbb{R}^{N \times H}$	coefficient matrix of confounding factors
λ, γ	regularization parameters

Table 2.1: Summary of Notations

Important notations used in this chapter are listed in Table 2.1. Throughout the chapter, we assume that, for each sample, the SNPs and genes are represented by column vectors. Let $\mathbf{x} = [x_1, x_2, \ldots, x_K]^{\mathrm{T}}$ represent the K SNPs in the study, where $x_i \in \{0, 1, 2\}$ is a random variable corresponding to the i-th SNP. For example, 0, 1, 2 may encode the homozygous major allele, heterozygous allele, and homozygous minor allele, respectively. Let $\mathbf{z} = [z_1, z_2, \ldots, z_N]^{\mathrm{T}}$ represent the N genes in the study, where z_j is a continuous random variable corresponding to the j-th gene.

The traditional linear regression model for association mapping between $\mathbf{x}$ and $\mathbf{z}$ is

$$\mathbf{z} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{2.1}$$

where $\mathbf{z}$ is a linear function of $\mathbf{x}$ with coefficient matrix $\mathbf{W}$. $\boldsymbol{\mu}$ is an $N \times 1$ translation factor vector. $\boldsymbol{\epsilon}$ is the additive noise of Gaussian distribution with zero-mean and variance $\psi\mathbf{I}$, where ψ is a scalar. That is, $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \psi\mathbf{I})$.

The question now is how to define an appropriate objective function to decompose $\mathbf{W}$ which (1) can effectively detect both individual and group-wise eQTL associations, and (2) is efficient to compute so that it is suitable for large-scale studies. In the next, we will propose a group-wise eQTL detection method first, and then improve it to capture both individual and group-wise associations. Finally, we will discuss how to boost the computational efficiency.

2.4 Detecting Group-Wise Associations

2.4.1 SET-eQTL Model

To infer associations between SNP sets and gene sets, we propose a graphical model as shown in Figure 2.3, which is able to capture any potential confounding factors in a natural way. This model is a two-layer linear Gaussian model. The hidden variables in the middle layer are used to capture the group-wise association between SNP sets and gene sets. These latent variables are presented as $\mathbf{y} = [y_1, y_2, \ldots, y_M]^{\mathrm{T}}$, where M is the total number of latent variables bridging SNP sets and gene sets. Each hidden variable may represent a latent factor regulating a set of genes, and its associated genes may correspond to a set of genes in the same pathway or participating in certain biological function. Note that this model allows a SNP or gene to participate in multiple (SNP set, gene set) pairs. This is reasonable because SNPs and genes may play different roles in multiple biology pathways. Since the model bridges SNP sets and gene sets, we refer this method as SET-eQTL.

The exact role of these latent factors can be inferred from the network topology of the resulting sparse graphical model learned from the data (by imposing ℓ_1-norm on the likelihood function, which will be discussed later in this section). Figure 2.2 shows an example of the

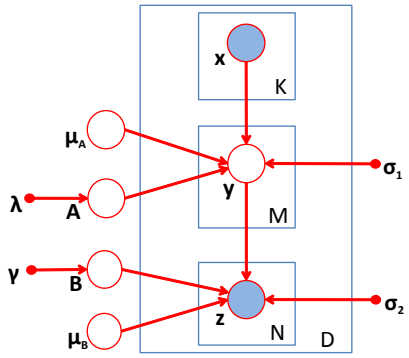


Figure 2.1: The proposed graphical model with hidden variables

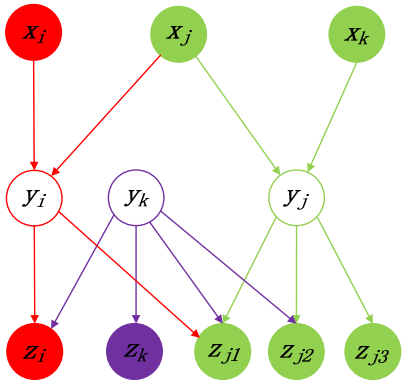


Figure 2.2: An example of the inferred sparse graphical model

resulting graphical model. There are two types of hidden variables. One type consists of hidden variables with zero in-degree (i.e., no connections with the SNPs). These hidden variables correspond to the confounding factors. Other types of hidden variables serve as bridges connecting SNP sets and gene sets. In Figure 2.2, y_k is a hidden variable modeling confounding effects. y_i and y_j are bridge nodes connecting the SNPs and genes associated with them. Note that this model allows overlaps between different (SNP set, gene set) pairs. It is reasonable because SNPs and genes may play multiple roles in different biology pathways.

2.4.2 Objective Function

From the probability theory, we have that the joint probability of $\mathbf{x}$ and $\mathbf{z}$ is

$$p(\mathbf{x}, \mathbf{z}) = \int_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}, \mathbf{z}) \mathrm{d}\mathbf{y}. \tag{2.2}$$

From the factorization properties of the joint distribution for a directed graphical model, we have

$$p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{z}|\mathbf{y})p(\mathbf{x}). \tag{2.3}$$

Thus, we have

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})} = \int_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})p(\mathbf{z}|\mathbf{y})\mathrm{d}\mathbf{y}. \tag{2.4}$$

We assume that the two conditional probabilities follow normal distributions:

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \boldsymbol{\mu}_{\mathbf{A}}, \sigma_{\mathbf{1}}^{\mathbf{2}}\mathbf{I}_{\mathbf{M}}),$$

and

$$\mathbf{z}|\mathbf{y} \sim \mathcal{N}(\mathbf{z}|\mathbf{B}\mathbf{y} + \boldsymbol{\mu}_{\mathbf{B}}, \sigma_{\mathbf{2}}^{\mathbf{2}}\mathbf{I}_{\mathbf{N}}),$$

where $\mathbf{A} \in \mathbb{R}^{M \times K}$ is the coefficient matrix between $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{B} \in \mathbb{R}^{N \times M}$ is the coefficient matrix between $\mathbf{y}$ and $\mathbf{z}$. $\boldsymbol{\mu}_{\mathbf{A}} \in \mathbb{R}^{M \times 1}$ and $\boldsymbol{\mu}_{\mathbf{B}} \in \mathbb{R}^{N \times 1}$ are the translation factor vectors, of which $\sigma_1^2\mathbf{I}_M$ and $\sigma_2^2\mathbf{I}_N$ are their variances respectively (σ_1 and σ_2 are constant scalars and $\mathbf{I}_M$ and $\mathbf{I}_N$ are identity matrices).

To impose sparsity, we assume that entries of $\mathbf{A}$ and $\mathbf{B}$ follow Laplace distributions:

$$\mathbf{A} \sim \textbf{Laplace}(\mathbf{0}, 1/\lambda),$$

and

$$\mathbf{B} \sim \textbf{Laplace}(\mathbf{0}, 1/\gamma).$$

λ and γ are parameters of the ℓ_1-regularization penalty on the objective function. This model is a two-layer linear model and $p(\mathbf{y}|\mathbf{x})$ serves as the conjugate prior of $p(\mathbf{z}|\mathbf{y})$. Thus we have

$$\boldsymbol{\beta} \cdot \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_\mathbf{y}, \boldsymbol{\Sigma}_\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \boldsymbol{\mu}_\mathbf{A}, \sigma_\mathbf{1}^\mathbf{2}\mathbf{I}_\mathbf{M}) \cdot \mathcal{N}(\mathbf{z}|\mathbf{B}\mathbf{y} + \boldsymbol{\mu}_\mathbf{B}, \sigma_\mathbf{2}^\mathbf{2}\mathbf{I}_\mathbf{N}) \tag{2.5}$$

where $\boldsymbol{\beta}$ is a scalar, $\boldsymbol{\mu}_\mathbf{y}$ and $\boldsymbol{\Sigma}_\mathbf{y}$ are the mean and variance of a new normal distribution respectively.

From Equations 2.4 and 2.5, we have that

$$p(\mathbf{z}|\mathbf{x}) = \int_\mathbf{y} \boldsymbol{\beta} \cdot \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_\mathbf{y}, \boldsymbol{\Sigma}_\mathbf{y})\mathrm{d}\mathbf{y} = \boldsymbol{\beta} \tag{2.6}$$

Thus, maximizing $p(\mathbf{z}|\mathbf{x})$ is equivalent to maximizing $\boldsymbol{\beta}$. Next, we show the derivation of $\boldsymbol{\beta}$. We first derive the value of $\boldsymbol{\mu}_\mathbf{y}$ and $\boldsymbol{\Sigma}_\mathbf{y}^{-1}$ by comparing the exponential terms on both sides of Equation 2.5.

$$\begin{aligned}
&\mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \boldsymbol{\mu}_\mathbf{A}, \sigma_\mathbf{1}^\mathbf{2}\mathbf{I}_\mathbf{M}) \cdot \mathcal{N}(\mathbf{z}|\mathbf{B}\mathbf{y} + \boldsymbol{\mu}_\mathbf{B}, \sigma_\mathbf{2}^\mathbf{2}\mathbf{I}_\mathbf{N}) \\
&= \frac{1}{(2\pi)^{\frac{M+N}{2}}\sigma_1^M\sigma_2^N} \exp\{-\frac{1}{2}[\frac{1}{\sigma_1^2}(\mathbf{y} - \mathbf{A}\mathbf{x} - \boldsymbol{\mu}_\mathbf{A})^\mathrm{T}(\mathbf{y} - \mathbf{A}\mathbf{x} - \boldsymbol{\mu}_\mathbf{A}) \\
&\qquad + \frac{1}{\sigma_2^2}(\mathbf{z} - \mathbf{B}\mathbf{y} - \boldsymbol{\mu}_\mathbf{B})^\mathrm{T}(\mathbf{z} - \mathbf{B}\mathbf{y} - \boldsymbol{\mu}_\mathbf{B})]\}
\end{aligned} \tag{2.7}$$

The exponential term in Equation 2.7 can be expanded as

$$\begin{aligned}
\boldsymbol{\Psi} &= -\frac{1}{2}[\frac{1}{\sigma_1^2}(\mathbf{y} - \mathbf{A}\mathbf{x} - \boldsymbol{\mu}_\mathbf{A})^\mathrm{T}(\mathbf{y} - \mathbf{A}\mathbf{x}) \\
&\qquad + \sigma_2^2(\mathbf{z} - \mathbf{B}\mathbf{y} - \boldsymbol{\mu}_\mathbf{B})^\mathrm{T}(\mathbf{z} - \mathbf{B}\mathbf{y})] \\
&= -\frac{1}{2}[\frac{1}{\sigma_1^2}(\mathbf{y}^\mathrm{T}\mathbf{y} - \mathbf{y}^\mathrm{T}\mathbf{A}\mathbf{x} - \mathbf{y}^\mathrm{T}\boldsymbol{\mu}_\mathbf{A} - \mathbf{x}^\mathrm{T}\mathbf{A}^\mathrm{T}\mathbf{y} + \mathbf{x}^\mathrm{T}\mathbf{A}^\mathrm{T}\mathbf{A}\mathbf{x} \\
&+ \mathbf{x}^\mathrm{T}\mathbf{A}^\mathrm{T}\boldsymbol{\mu}_\mathbf{A} - \boldsymbol{\mu}_\mathbf{A}^\mathrm{T}\mathbf{y} + \boldsymbol{\mu}_\mathbf{A}^\mathrm{T}\mathbf{A}X + \boldsymbol{\mu}_\mathbf{A}^\mathrm{T}\boldsymbol{\mu}_\mathbf{A}) + \frac{1}{\sigma_2^2}(\mathbf{z}^\mathrm{T}\mathbf{z} - \mathbf{z}^\mathrm{T}\mathbf{B}\mathbf{y} \\
&- \mathbf{z}^\mathrm{T}\boldsymbol{\mu}_\mathbf{B} - \mathbf{y}^\mathrm{T}\mathbf{B}^\mathrm{T}\mathbf{z} + \mathbf{y}^\mathrm{T}\mathbf{B}^\mathrm{T}\mathbf{B}\mathbf{y} + \mathbf{y}^\mathrm{T}\mathbf{B}^\mathrm{T}\boldsymbol{\mu}_\mathbf{B} - \boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\mathbf{z} + \boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\mathbf{B}\mathbf{y} \\
&+ \boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\boldsymbol{\mu}_\mathbf{B})] \\
&= -\frac{1}{2}[\mathbf{y}^\mathrm{T}(\frac{1}{\sigma_1^2}\mathbf{I}_M + \frac{1}{\sigma_2^2}\mathbf{B}^\mathrm{T}\mathbf{B})\mathbf{y} - \frac{2}{\sigma_1^2}(\mathbf{x}^\mathrm{T}\mathbf{A}^\mathrm{T}\mathbf{y} + \boldsymbol{\mu}_\mathbf{A}^\mathrm{T}\mathbf{y}) \\
&- \frac{2}{\sigma_2^2}(\mathbf{z}^\mathrm{T}\mathbf{B}\mathbf{y} - \boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\mathbf{B}\mathbf{y}) + \frac{1}{\sigma_1^2}(\mathbf{x}^\mathrm{T}\mathbf{A}^\mathrm{T}\mathbf{A}\mathbf{x} + 2\boldsymbol{\mu}_\mathbf{A}^\mathrm{T}\mathbf{A}\mathbf{x} + \boldsymbol{\mu}_\mathbf{A}^\mathrm{T}\boldsymbol{\mu}_\mathbf{A}) \\
&+ \frac{1}{\sigma_2^2}(\mathbf{z}^\mathrm{T}\mathbf{z} - 2\boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\mathbf{z} + \boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\boldsymbol{\mu}_\mathbf{B})]
\end{aligned} \tag{2.8}$$

Thus, by comparing the exponential terms on both sides of Equation 2.5, we get

$$\boldsymbol{\Sigma}_{\mathbf{y}}^{-1} = \frac{1}{\sigma_1^2}\mathbf{I}_M + \frac{1}{\sigma_2^2}\mathbf{B}^{\mathrm{T}}\mathbf{B}, \tag{2.9}$$

$$\boldsymbol{\mu}_{\mathbf{y}}^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1} = \frac{1}{\sigma_1^2}(\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} + \boldsymbol{\mu}_{\mathbf{A}}^{\mathrm{T}}) + \frac{1}{\sigma_2^2}(\mathbf{z}^{\mathrm{T}}\mathbf{B} - \boldsymbol{\mu}_{\mathbf{B}}^{\mathrm{T}}\mathbf{B}). \tag{2.10}$$

Further, we have

$$\boldsymbol{\mu}_{\mathbf{y}} = \boldsymbol{\Sigma}_{\mathbf{y}}[\frac{1}{\sigma_1^2}(\mathbf{A}\mathbf{x} + \boldsymbol{\mu}_{\mathbf{A}}) + \frac{1}{\sigma_2^2}(\mathbf{B}^{\mathrm{T}}\mathbf{z} - \mathbf{B}^{\mathrm{T}}\boldsymbol{\mu}_{\mathbf{B}})]. \tag{2.11}$$

With $\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}$ and $\boldsymbol{\mu}_{\mathbf{y}}$, we can derive the explicit form of $\boldsymbol{\beta}$ easily by setting $\mathbf{y} = \mathbf{0}$, which leads to the equation below:

$$\begin{aligned} &\boldsymbol{\beta} \cdot \tfrac{1}{(2\pi)^{\frac{M}{2}}|\boldsymbol{\Sigma}_{\mathbf{y}}|^{\frac{1}{2}}} \exp\{-\tfrac{1}{2}\boldsymbol{\mu}_{\mathbf{y}}^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\boldsymbol{\mu}_{\mathbf{y}}\} \\ &= \tfrac{1}{(2\pi)^{\frac{M+N}{2}}\sigma_1^M\sigma_2^N} \exp\{\boldsymbol{\Psi}_{\mathbf{y}=\mathbf{0}}\}, \end{aligned} \tag{2.12}$$

where $\boldsymbol{\Psi}_{\mathbf{y}=\mathbf{0}}$ is the value of $\boldsymbol{\Psi}$ when $\mathbf{y} = \mathbf{0}$, and thereby

$$\begin{aligned} &\boldsymbol{\Psi}_{\mathbf{y}=\mathbf{0}} = -\tfrac{1}{2}[\tfrac{1}{\sigma_1^2}(\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} + 2\boldsymbol{\mu}_{\mathbf{A}}^{\mathrm{T}}\mathbf{A}\mathbf{x} + \boldsymbol{\mu}_{\mathbf{A}}^{\mathrm{T}}\boldsymbol{\mu}_{\mathbf{A}}) \\ &+\tfrac{1}{\sigma_2^2}(\mathbf{z}^{\mathrm{T}}\mathbf{z} - 2\boldsymbol{\mu}_{\mathbf{B}}^{\mathrm{T}}\mathbf{z} + \boldsymbol{\mu}_{\mathbf{B}}^{\mathrm{T}}\boldsymbol{\mu}_{\mathbf{B}})] \end{aligned} \tag{2.13}$$

Thus, we get the explicit form of $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta} = \tfrac{|\boldsymbol{\Sigma}_{\mathbf{y}}|^{\frac{1}{2}}}{(2\pi)^{\frac{N}{2}}\sigma_1^M\sigma_2^N} \exp\{\boldsymbol{\Psi}_{\mathbf{y}=\mathbf{0}} + \tfrac{1}{2}(\boldsymbol{\mu}_{\mathbf{y}}^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\boldsymbol{\mu}_{\mathbf{y}})\}. \tag{2.14}$$

Here, $\boldsymbol{\beta} = p(\mathbf{z}|\mathbf{x}, \mathbf{A}, \mathbf{B}, \boldsymbol{\mu}_{\mathbf{A}}, \boldsymbol{\mu}_{\mathbf{B}}, \sigma_1, \sigma_2)$ is the likelihood function for one data point $\mathbf{x}$. Let $\mathbf{X} = \{\mathbf{x}_d\}$ and $\mathbf{Z} = \{\mathbf{z}_d\}$ be the sets of D observed data points (genotype and the gene expression profiles for the samples in the study). To maximize $\boldsymbol{\beta}_d$, we can minimize the negative

log-likelihood of $\boldsymbol{\beta}_d$. Thus, our loss function is

$$\begin{aligned}\mathcal{J} &= -\log \prod_{d=1}^{D} p(\mathbf{z}_d|\mathbf{x}_d) \\ &= -\sum_{d=1}^{D} \log p(\mathbf{z}_d|\mathbf{x}_d) \\ &= -\sum_{d=1}^{D} \log \boldsymbol{\beta}_d\end{aligned} \tag{2.15}$$

Substituting Equation 2.14 into Equation 2.15, the expanded form of the loss function is

$$\begin{aligned}&\mathcal{J}(\mathbf{A}, \mathbf{B}, \boldsymbol{\mu}_{\mathbf{A}}, \boldsymbol{\mu}_{\mathbf{B}}, \sigma_1, \sigma_2) \\ &= \tfrac{D \cdot N}{2} \ln(2\pi) + D \cdot M \ln(\sigma_1) + D \cdot N \ln(\sigma_2) + \tfrac{D}{2} \ln |\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}| \\ &+ \tfrac{1}{2} \sum_{d=1}^{D} \{ \tfrac{1}{\sigma_1^2} (\mathbf{x}_d^{\mathrm{T}} \mathbf{A}^{\mathrm{T}} \mathbf{A} \mathbf{x}_d + 2\boldsymbol{\mu}_{\mathbf{A}}^{\mathrm{T}} \mathbf{A} \mathbf{x}_d + \boldsymbol{\mu}_{\mathbf{A}}^{\mathrm{T}} \boldsymbol{\mu}_{\mathbf{A}}) \\ &+ \tfrac{1}{\sigma_2^2} (\mathbf{z}_d^{\mathrm{T}} \mathbf{z}_d - 2\boldsymbol{\mu}_{\mathbf{B}}^{\mathrm{T}} \mathbf{z}_d + \boldsymbol{\mu}_{\mathbf{B}}^{\mathrm{T}} \boldsymbol{\mu}_{\mathbf{B}}) - [\tfrac{1}{\sigma_1^2} (\mathbf{x}_d^{\mathrm{T}} \mathbf{A}^{\mathrm{T}} + \boldsymbol{\mu}_A^{\mathrm{T}}) \\ &+ \tfrac{1}{\sigma_2^2} (\mathbf{z}_d^{\mathrm{T}} \mathbf{B} - \boldsymbol{\mu}_{\mathbf{B}}^{\mathrm{T}} \mathbf{B})] \boldsymbol{\Sigma}_{\mathbf{y}} [\tfrac{1}{\sigma_1^2} (\mathbf{A} \mathbf{x}_d + \boldsymbol{\mu}_A) + \tfrac{1}{\sigma_2^2} (\mathbf{B}^{\mathrm{T}} \mathbf{z}_d - \mathbf{B}^{\mathrm{T}} \boldsymbol{\mu}_{\mathbf{B}})] \}\end{aligned} \tag{2.16}$$

Taking into account the prior distributions of **A** and **B**, we have that

$$\begin{aligned}&p(\mathbf{z}, \mathbf{A}, \mathbf{B}|\mathbf{x}, \boldsymbol{\mu}_{\mathbf{A}}, \boldsymbol{\mu}_{\mathbf{B}}, \sigma_1, \sigma_2) \\ &= \boldsymbol{\beta} \cdot \mathbf{Laplace}(\mathbf{A}|\mathbf{0}, 1/\lambda) \cdot \mathbf{Laplace}(\mathbf{B}|\mathbf{0}, 1/\gamma)\end{aligned} \tag{2.17}$$

Thus, we can have the ℓ_1-regularized objective function

$$\max_{\mathbf{A}, \mathbf{B}, \boldsymbol{\mu}_{\mathbf{A}}, \boldsymbol{\mu}_{\mathbf{B}}, \sigma_1, \sigma_2} \log \prod_{d=1}^{D} p(\mathbf{z}_d, \mathbf{A}, \mathbf{B}|\mathbf{x}_d, \boldsymbol{\mu}_{\mathbf{A}}, \boldsymbol{\mu}_{\mathbf{B}}, \sigma_1, \sigma_2),$$

which is identical to

$$\min_{\mathbf{A}, \mathbf{B}, \boldsymbol{\mu}_{\mathbf{A}}, \boldsymbol{\mu}_{\mathbf{B}}, \sigma_1, \sigma_2} [\mathcal{J} + D \cdot (\lambda ||\mathbf{A}||_1 + \gamma ||\mathbf{B}||_1)], \tag{2.18}$$

where $|| \cdot ||_1$ is the ℓ_1-norm. λ and γ are the *precision* of the prior Laplace distributions of **A** and **B** respectively, serving as the regularization parameters which can be determined by cross or holdout validation.

The gradient of the loss function $\mathcal{J}$ with respect to $\mathbf{A}$, $\mathbf{B}$, $\boldsymbol{\mu}_\mathbf{A}$, $\boldsymbol{\mu}_\mathbf{B}$, σ_1, and σ_2 are:

$$\begin{aligned}\nabla_\mathbf{A}\mathcal{J} = &\textstyle\sum_{d=1}^{D}(\frac{1}{\sigma_1^2}\mathbf{A}\mathbf{x}_d\mathbf{x}_d^\mathrm{T} - \frac{1}{\sigma_1^4}\boldsymbol{\Sigma}_\mathbf{y}\mathbf{A}\mathbf{x}_d\mathbf{x}_d^\mathrm{T} - \frac{1}{\sigma_1^2\sigma_2^2}\boldsymbol{\Sigma}_\mathbf{y}\mathbf{B}^\mathrm{T}\mathbf{z}_d\mathbf{x}_d^\mathrm{T}\\ &+\frac{1}{\sigma_1^2}\boldsymbol{\mu}_\mathbf{A}\mathbf{x}_d^\mathrm{T} - \frac{1}{\sigma_1^4}\boldsymbol{\Sigma}_\mathbf{y}\boldsymbol{\mu}_\mathbf{A}\mathbf{x}_d^\mathrm{T} + \frac{1}{\sigma_1^2\sigma_2^2}\boldsymbol{\Sigma}_\mathbf{y}\mathbf{B}^\mathrm{T}\boldsymbol{\mu}_\mathbf{B}\mathbf{x}_d^\mathrm{T})\end{aligned} \tag{2.19}$$

$$\begin{aligned}\nabla_\mathbf{B}\mathcal{J} = &\tfrac{D}{\sigma_2^2}\mathbf{B}\boldsymbol{\Sigma}_\mathbf{y} + \tfrac{1}{\sigma_2^4}(\tfrac{1}{\sigma_2^2}\mathbf{B}\boldsymbol{\Sigma}_\mathbf{y}\mathbf{B}^\mathrm{T} - \mathbf{I}_N)\textstyle\sum_{d=1}^{D}[(\mathbf{z}_d - \boldsymbol{\mu}_\mathbf{B})\\ &\cdot(\mathbf{z}_d - \boldsymbol{\mu}_\mathbf{B})^\mathrm{T}]\mathbf{B}\boldsymbol{\Sigma}_\mathbf{y} + \tfrac{1}{\sigma_1^2\sigma_2^4}\textstyle\sum_{d=1}^{D}\{\mathbf{B}\boldsymbol{\Sigma}_\mathbf{y}[(\mathbf{A}\mathbf{x}_d + \boldsymbol{\mu}_\mathbf{A})(\mathbf{z}_d - \boldsymbol{\mu}_\mathbf{B})^\mathrm{T}\mathbf{B}\\ &+\mathbf{B}^\mathrm{T}(\mathbf{z}_d - \boldsymbol{\mu}_\mathbf{B})(\mathbf{A}\mathbf{x}_d + \boldsymbol{\mu}_\mathbf{A})^\mathrm{T}]\boldsymbol{\Sigma}_\mathbf{y} - \sigma_2^2(\mathbf{z}_d - \boldsymbol{\mu}_\mathbf{B})(\mathbf{A}\mathbf{x}_d + \boldsymbol{\mu}_\mathbf{A})^\mathrm{T}\boldsymbol{\Sigma}_\mathbf{y}\}\\ &+\tfrac{1}{\sigma_1^4\sigma_2^2}B\boldsymbol{\Sigma}_\mathbf{y}\textstyle\sum_{d=1}^{D}[(\mathbf{A}\mathbf{x}_d + \boldsymbol{\mu}_\mathbf{A})(\mathbf{x}_d^\mathrm{T}\mathbf{A}^\mathrm{T} + \boldsymbol{\mu}_\mathbf{A}^\mathrm{T})]\boldsymbol{\Sigma}_\mathbf{y}\end{aligned} \tag{2.20}$$

$$\nabla_{\boldsymbol{\mu}_\mathbf{A}}\mathcal{J} = \tfrac{1}{2}\textstyle\sum_{d=1}^{D}[\frac{2}{\sigma_1^2}(\mathbf{A}\mathbf{x}_d + \boldsymbol{\mu}_\mathbf{A}) - \frac{2}{\sigma_1^4}\boldsymbol{\Sigma}_\mathbf{y}(\boldsymbol{\mu}_\mathbf{A} + \mathbf{A}\mathbf{x}_d) - \frac{2}{\sigma_1^2\sigma_2^2}\boldsymbol{\Sigma}_\mathbf{y}(\mathbf{B}^\mathrm{T}\mathbf{z}_d - \mathbf{B}^\mathrm{T}\boldsymbol{\mu}_\mathbf{B})] \tag{2.21}$$

$$\nabla_{\boldsymbol{\mu}_\mathbf{B}}\mathcal{J} = \tfrac{1}{2}\textstyle\sum_{d=1}^{D}[\frac{2}{\sigma_1^2}(-\mathbf{z}_d + \boldsymbol{\mu}_\mathbf{B}) + \frac{2}{\sigma_1^4}\mathbf{B}\boldsymbol{\Sigma}_\mathbf{y}\mathbf{B}^\mathrm{T}(\mathbf{z}_d - \boldsymbol{\mu}_\mathbf{B}) + \frac{2}{\sigma_1^2\sigma_2^2}\mathbf{B}\boldsymbol{\Sigma}_\mathbf{y}(\mathbf{A}\mathbf{x}_d + \boldsymbol{\mu}_\mathbf{A})] \tag{2.22}$$

$$\begin{aligned}\nabla_{\sigma_1}\mathcal{J} = &\tfrac{D\cdot M}{\sigma_1} - \tfrac{D\cdot\mathrm{tr}(\Sigma_\mathbf{y})}{\sigma_1^3} + \textstyle\sum_{d=1}^{D}[-\frac{\mathbf{x}_d^\mathrm{T}\mathbf{A}^\mathrm{T}\mathbf{A}\mathbf{x}_d + 2\boldsymbol{\mu}_\mathbf{A}^\mathrm{T}\mathbf{A}\mathbf{x}_d + \boldsymbol{\mu}_\mathbf{A}^\mathrm{T}\boldsymbol{\mu}_\mathbf{A}}{\sigma_1^3}\\ &+\tfrac{2(\mathbf{x}_d^\mathrm{T}\mathbf{A}^\mathrm{T} + \boldsymbol{\mu}_\mathbf{A}^\mathrm{T})\boldsymbol{\Sigma}_\mathbf{y}(\mathbf{A}\mathbf{x}_d + \boldsymbol{\mu}_\mathbf{A})}{\sigma_1^5} - \tfrac{(\mathbf{x}_d^\mathrm{T}\mathbf{A}^\mathrm{T} + \boldsymbol{\mu}_\mathbf{A}^\mathrm{T})\boldsymbol{\Sigma}_\mathbf{y}^2(\mathbf{A}\mathbf{x}_d + \boldsymbol{\mu}_\mathbf{A})}{\sigma_1^7}\\ &+\tfrac{2(\mathbf{x}_d^\mathrm{T}\mathbf{A}^\mathrm{T} + \boldsymbol{\mu}_\mathbf{A}^\mathrm{T})\boldsymbol{\Sigma}_\mathbf{y}(\mathbf{B}^\mathrm{T}\mathbf{z}_d - \mathbf{B}^\mathrm{T}\boldsymbol{\mu}_\mathbf{B})}{\sigma_1^3\sigma_2^2} - \tfrac{2(\mathbf{x}_d^\mathrm{T}\mathbf{A}^\mathrm{T} + \boldsymbol{\mu}_\mathbf{A}^\mathrm{T})\boldsymbol{\Sigma}_\mathbf{y}^2(\mathbf{B}^\mathrm{T}\mathbf{z}_d - \mathbf{B}^\mathrm{T}\boldsymbol{\mu}_\mathbf{B})}{\sigma_1^5\sigma_2^2}\\ &-\tfrac{(\mathbf{z}_d^\mathrm{T}\mathbf{B} - \boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\mathbf{B})\boldsymbol{\Sigma}_\mathbf{y}^2(\mathbf{B}^\mathrm{T}\mathbf{z}_d - \mathbf{B}^\mathrm{T}\boldsymbol{\mu}_\mathbf{B})}{\sigma_1^3\sigma_2^4}]\end{aligned} \tag{2.23}$$

$$\begin{aligned}\nabla_{\sigma_2}\mathcal{J} = &\tfrac{D\cdot N}{\sigma_2} - \tfrac{D\cdot\mathrm{tr}(\Sigma_\mathbf{y}\mathbf{B}^\mathrm{T}\mathbf{B})}{\sigma_2^3} + \textstyle\sum_{d=1}^{D}[-\frac{\mathbf{z}_d^\mathrm{T}\mathbf{z}_d - 2\boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\mathbf{z}_d + \boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\boldsymbol{\mu}_\mathbf{B}}{\sigma_2^3}\\ &+\tfrac{2(\mathbf{z}_d^\mathrm{T}\mathbf{B} - \boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\mathbf{B})\boldsymbol{\Sigma}_\mathbf{y}(\mathbf{B}^\mathrm{T}\mathbf{z}_d - \mathbf{B}^\mathrm{T}\boldsymbol{\mu}_\mathbf{B})}{\sigma_2^5} - \tfrac{(\mathbf{z}_d^\mathrm{T}\mathbf{B} - \boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\mathbf{B})\boldsymbol{\Sigma}_\mathbf{y}\mathbf{B}^\mathrm{T}\mathbf{B}\boldsymbol{\Sigma}_\mathbf{y}(\mathbf{B}^\mathrm{T}\mathbf{z}_d - \mathbf{B}^\mathrm{T}\boldsymbol{\mu}_\mathbf{B})}{\sigma_2^7}\\ &+\tfrac{2(\mathbf{z}_d^\mathrm{T}\mathbf{B} - \boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\mathbf{B})\boldsymbol{\Sigma}_\mathbf{y}(\mathbf{A}\mathbf{x}_d + \boldsymbol{\mu}_\mathbf{A})}{\sigma_1^2\sigma_2^3} - \tfrac{2(\mathbf{z}_d^\mathrm{T}\mathbf{B} - \boldsymbol{\mu}_\mathbf{B}^\mathrm{T}\mathbf{B})\boldsymbol{\Sigma}_\mathbf{y}\mathbf{B}^\mathrm{T}\mathbf{B}\boldsymbol{\Sigma}_\mathbf{y}(\mathbf{A}\mathbf{x}_d + \boldsymbol{\mu}_\mathbf{A})}{\sigma_1^2\sigma_2^5}\\ &-\tfrac{(\mathbf{x}_d^\mathrm{T}\mathbf{A}^\mathrm{T} + \boldsymbol{\mu}_\mathbf{A}^\mathrm{T})\boldsymbol{\Sigma}_\mathbf{y}\mathbf{B}^\mathrm{T}\mathbf{B}\boldsymbol{\Sigma}_\mathbf{y}(\mathbf{A}\mathbf{x}_d + \boldsymbol{\mu}_\mathbf{A})}{\sigma_1^4\sigma_2^3}]\end{aligned} \tag{2.24}$$

2.5 Considering Confounding Factors

To infer associations between SNP sets and gene sets while taking into consideration confounding factors, we further propose a graphical model as shown in Figure 2.3. Different from the previous model, a new type of hidden variable, $\mathbf{s} = [s_1, s_2, \ldots, s_H]^{\mathrm{T}}$, is used to model confounding factors. For simplicity, we refer to this model as *Model 1*. The objective function of this model can be derivated using similar strategy as SET-eQTL.

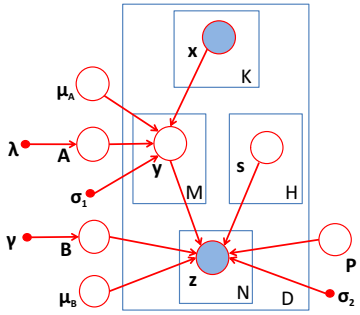


Figure 2.3: Graphical model with two types of hidden variables

2.6 Incorporating Individual Effect

In the graphical model shown in Figure 2.3, we use a hidden variable y as a bridge between a SNP set and a gene set to capture the group-wise effect. In addition, individual effects may exist as well (Listgarten et al., 2013). An example is shown in Figure 1.4. Note that an ideal model should allow overlaps between SNP sets and between gene sets; that is, a SNP or gene may participate in multiple individual and group-wise associations. To incorporate both individual and group-wise effects, we extend the model in Figure 2.3 and add one edge between $\mathbf{x}$ and $\mathbf{z}$ to capture individual associations as shown in Figure 2.4. We will show that this refinement will significantly improve the accuracy of model and enhance its computational efficiency. For simplicity, we refer to the new model that considers both individual and group-wise associations as *Model 2*.

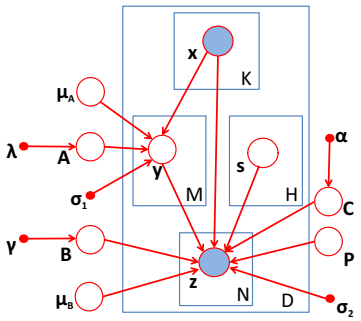


Figure 2.4: Refined graphical model to capture both individual and group-wise associations.

2.6.1 Objective Function

Next, we give the derivation of the objective function for the model in Figure 2.4. We assume that the two conditional probabilities follow normal distributions:

$$\mathbf{y}|\mathbf{x} \sim N(\mathbf{y}|\mathbf{A}\mathbf{x} + \boldsymbol{\mu}_\mathbf{A}, \sigma_\mathbf{1}^\mathbf{2}\mathbf{I}_\mathbf{M}), \tag{2.25}$$

and

$$\mathbf{z}|\mathbf{y}, \mathbf{x} \sim N(\mathbf{z}|\mathbf{B}\mathbf{y} + \mathbf{C}\mathbf{x} + \mathbf{P}\mathbf{s} + \boldsymbol{\mu}_\mathbf{B}, \sigma_\mathbf{2}^\mathbf{2}\mathbf{I}_\mathbf{N}), \tag{2.26}$$

where $\mathbf{A} \in \mathbb{R}^{M\times K}$ is the coefficient matrix between $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{B} \in \mathbb{R}^{N\times M}$ is the coefficient matrix between $\mathbf{y}$ and $\mathbf{z}$, $\mathbf{C} \in \mathbb{R}^{N\times K}$ is the coefficient matrix between $\mathbf{x}$ and $\mathbf{z}$ to capture the individual associations, $\mathbf{P} \in \mathbb{R}^{N\times H}$ is the coefficient matrix of confounding factors. $\boldsymbol{\mu}_\mathbf{A} \in \mathbb{R}^{M\times 1}$ and $\boldsymbol{\mu}_\mathbf{B} \in \mathbb{R}^{N\times 1}$ are the translation factor vectors, $\sigma_1^2\mathbf{I}_M$ and $\sigma_2^2\mathbf{I}_N$ are the variances of the two conditional probabilities respectively (σ_1 and σ_2 are constant scalars and $\mathbf{I}_M$ and $\mathbf{I}_N$ are identity matrices).

Since the expression level of a gene is usually affected by a small fraction of SNPs, we impose sparsity on $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. We assume that the entries of these matrices follow Laplace distributions: $\mathbf{A}_{i,j} \sim \textbf{Laplace}(0, 1/\lambda)$, $\mathbf{B}_{i,j} \sim \textbf{Laplace}(0, 1/\gamma)$, and $\mathbf{C}_{i,j} \sim \textbf{Laplace}(0, 1/\alpha)$. λ, γ and α will be used as parameters in the objective function. The probability density function of $\textbf{Laplace}(\mu, b)$ distribution is $f(x|\mu, b) = \frac{1}{2b}\exp(-\frac{|x-\mu|}{b})$.

Thus, we have

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\mu}_{\mathbf{A}} + \boldsymbol{\epsilon}_1, \tag{2.27}$$

$$\mathbf{z} = \mathbf{B}\mathbf{y} + \mathbf{C}\mathbf{x} + \mathbf{P}\mathbf{s} + \boldsymbol{\mu}_{\mathbf{B}} + \boldsymbol{\epsilon}_2, \tag{2.28}$$

where $\boldsymbol{\epsilon}_1 \sim N(\mathbf{0}, \sigma_1^2\mathbf{I}_M)$,$\boldsymbol{\epsilon}_2 \sim N(\mathbf{0}, \sigma_2^2\mathbf{I}_N)$. From Eq. (2.25) we have

$$\mathbf{B}\mathbf{y}|\mathbf{x} \sim N(\mathbf{B}\mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\mu}_{\mathbf{A}}, \sigma_1^2\mathbf{B}\mathbf{B}^{\mathrm{T}}), \tag{2.29}$$

Assuming that the confounding factors follow normal distribution (Listgarten et al., 2013), $\mathbf{s} \sim N(\mathbf{0}, \mathbf{I}_H)$, then we have

$$\mathbf{P}\mathbf{s} \sim N(\mathbf{0}, \mathbf{P}\mathbf{P}^{\mathrm{T}}). \tag{2.30}$$

We substitute Eq. (2.29), (2.30) into Eq. (2.28), and get

$$\mathbf{z}|\mathbf{x} \sim N(\mathbf{B}\mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\mu}_{\mathbf{A}} + \mathbf{C}\mathbf{x} + \boldsymbol{\mu}_{\mathbf{B}}, \sigma_1^2\mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{P}\mathbf{P}^{\mathrm{T}} + \sigma_2^2\mathbf{I}_N).$$

From the formula above, we observe that the summand $\mathbf{B}\boldsymbol{\mu}_{\mathbf{A}}$ can also be integrated in $\boldsymbol{\mu}_{\mathbf{B}}$. Thus to simplify the model, we set $\boldsymbol{\mu}_{\mathbf{A}} = \mathbf{0}$ and obtain

$$\mathbf{z}|\mathbf{x} \sim N(\mathbf{B}\mathbf{A}\mathbf{x} + \mathbf{C}\mathbf{x} + \boldsymbol{\mu}_{\mathbf{B}}, \sigma_1^2\mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{P}\mathbf{P}^{\mathrm{T}} + \sigma_2^2\mathbf{I}_N).$$

To learn the parameters, we can use MLE (Maximize Likelihood Estimation) or MAP (Maximum a posteriori). Then, we get the likelihood function as $p(\mathbf{z}|\mathbf{x}) = \prod_{d=1}^{D} p(\mathbf{z}_d|\mathbf{x}_d)$. Maximizing the likelihood function is identical to minimizing the negative log-likelihood. Here,

the negative log-likelihood (loss function) is

$$\begin{aligned}
&\mathcal{J} = \sum_{d=1}^{D} \mathcal{J}_d \\
&= -1 \cdot \log \prod_{d=1}^{D} p(\mathbf{z}_d|\mathbf{x}_d) \\
&= \sum_{d=1}^{D} (-1) \cdot \log p(\mathbf{z}_d|\mathbf{x}_d) \\
&= \frac{D \cdot N}{2} \log(2\pi) + \frac{D}{2} \log |\mathbf{\Sigma}| + \frac{1}{2} \sum_{d=1}^{D} [(\mathbf{z}_d - \boldsymbol{\mu}_d)^{\mathbf{T}} \mathbf{\Sigma}^{-1} (\mathbf{z}_d - \boldsymbol{\mu}_d)],
\end{aligned} \tag{2.31}$$

where

$$\boldsymbol{\mu}_d = \mathbf{B}\mathbf{A}\mathbf{x}_d + \mathbf{C}\mathbf{x}_d + \boldsymbol{\mu}_{\mathbf{B}},$$

$$\mathbf{\Sigma} = \sigma_1^2 \mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma_2^2 \mathbf{I}_N.$$

Moreover, taking into account the prior distributions of **A**, **B** and **C**, we have

$$\begin{aligned}
&p(\mathbf{z}_d, \mathbf{A}, \mathbf{B}, \mathbf{C}|\mathbf{x}_d, \mathbf{P}, \sigma_1, \sigma_2) = \\
&\exp(-\mathcal{J}_d) \cdot \tfrac{\lambda}{2} \textstyle\prod_{i,j} \exp(-\lambda|\mathbf{A}_{i,j}|) \cdot \tfrac{\gamma}{2} \prod_{i,j} \exp(-\gamma|\mathbf{B}_{i,j}|) \cdot \tfrac{\alpha}{2} \prod_{i,j} \exp(-\alpha|\mathbf{C}_{i,j}|).
\end{aligned} \tag{2.32}$$

Thus, we have the ℓ_1-regularized objective function

$$\max_{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{P},\sigma_1,\sigma_2} \log \prod_{d=1}^{D} p(\mathbf{z}_d, \mathbf{A}, \mathbf{B}, \mathbf{C}|\mathbf{x}_d, \mathbf{P}, \sigma_1, \sigma_2),$$

which is identical to

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{P},\sigma_1,\sigma_2} [\mathcal{J} + D \cdot (\lambda||\mathbf{A}||_1 + \gamma||\mathbf{B}||_1 + \alpha||\mathbf{C}||_1)], \tag{2.33}$$

where $||\cdot||_1$ is the ℓ_1-norm. λ, γ and α are the *precision* of the prior Laplace distributions of **A**, **B**, and **C** respectively. They serve as the regularization parameters and can be determined by cross or holdout validation.

The explicit expression of $\boldsymbol{\mu}_{\mathbf{B}}$ can be derived as follows. When $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are fixed, we have

$\mathcal{J} = \frac{D \cdot N}{2} \log(2\pi) + \frac{D}{2} \log |\mathbf{\Sigma}| + \frac{1}{2} \sum_{d=1}^{D} [(\mathbf{z}_d - \mathbf{B}\mathbf{A}\mathbf{x}_d - \mathbf{C}\mathbf{x}_d - \boldsymbol{\mu}_{\mathbf{B}})^{\mathrm{T}} \mathbf{\Sigma}^{-1} (\mathbf{z}_d - \mathbf{B}\mathbf{A}\mathbf{x}_d - \mathbf{C}\mathbf{x}_d - \boldsymbol{\mu}_{\mathbf{B}})]$.

When $D = 1$, this is a classic maximum likelihood estimation problem, and we have

$\boldsymbol{\mu}_{\mathbf{B}} = \mathbf{z}_d - \mathbf{B}\mathbf{A}\mathbf{x}_d - \mathbf{C}\mathbf{x}_d$. When $D > 1$, leveraging the fact that $\boldsymbol{\Sigma}^{-1}$ is symmetric, we convert the problem into a least-square problem, which leads to

$$\boldsymbol{\mu}_{\mathbf{B}} = \frac{1}{D}\sum_{d=1}^{D}(\mathbf{z}_d - \mathbf{B}\mathbf{A}\mathbf{x}_d - \mathbf{C}\mathbf{x}_d).$$

Substituting it into Eq. (2.31), we have

$$\begin{aligned}\mathcal{J} = &\tfrac{D\cdot N}{2}\log(2\pi) + \tfrac{D}{2}\log|\boldsymbol{\Sigma}| + \tfrac{1}{2}\textstyle\sum_{d=1}^{D}\{[(\mathbf{z}_d - \bar{\mathbf{z}}) \\ &-(\mathbf{B}\mathbf{A}+\mathbf{C})(\mathbf{x}_d - \bar{\mathbf{x}})]^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}[(\mathbf{z}_d - \bar{\mathbf{z}}) - (\mathbf{B}\mathbf{A}+\mathbf{C})(\mathbf{x}_d - \bar{\mathbf{x}})]\},\end{aligned} \tag{2.34}$$

where

$$\bar{\mathbf{x}} = \frac{1}{D}\sum_{d=1}^{D}\mathbf{x}_d, \qquad \bar{\mathbf{z}} = \frac{1}{D}\sum_{d=1}^{D}\mathbf{z}_d.$$

The gradient of the loss function, which (without detailed derivation) is given in the below.

1). Derivative with respect to σ_1

$$\nabla_{\sigma_1}\mathcal{O} = 2\sigma_1\sum_{d=1}^{D}\{\mathrm{tr}[\boldsymbol{\Psi}_d]\mathbf{B}\mathbf{B}^{\mathrm{T}}\}. \tag{2.35}$$

2). Derivative with respect to σ_2

$$\nabla_{\sigma_2}\mathcal{O} = 2\sigma_2\sum_{d=1}^{D}\{\mathrm{tr}[\boldsymbol{\Psi}_d]\}. \tag{2.36}$$

3). Derivative with respect to $\mathbf{A}$

$$\nabla_{\mathbf{A}}\mathcal{O} = -\sum_{d=1}^{D}[\mathbf{B}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{t}_d(\mathbf{x}_d - \bar{\mathbf{x}})^{\mathrm{T}}]. \tag{2.37}$$

4). Derivative with respect to $\mathbf{B}$

$$\nabla_{\mathbf{B}}\mathcal{O} = \boldsymbol{\Xi}_{\mathbf{1}} + \boldsymbol{\Xi}_{\mathbf{2}}, \tag{2.38}$$

where

$$\boldsymbol{\Xi}_{\mathbf{1}} = -\sum_{d=1}^{D}[\boldsymbol{\Sigma}^{-1}\mathbf{t}_d(\mathbf{x}_d - \bar{\mathbf{x}})^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}], \tag{2.39}$$

$$(\boldsymbol{\Xi}_{\mathbf{2}})_{ij} = \sigma_1^2\sum_{d=1}^{D}\{\mathrm{tr}[\boldsymbol{\Psi}_d(\mathbf{E}_{ij}\mathbf{B}^{\mathrm{T}} + \mathbf{B}\mathbf{E}_{ji})]\}. \tag{2.40}$$

($\mathrm{tr}[\cdot]$ stands for trace; $\mathbf{E}_{ij}$ is the single-entry matrix: 1 at (i, j) and 0 elsewhere.)

We speed up this calculation by exploiting sparsity of $\boldsymbol{E}_{ij}$ and $\mathrm{tr}[\cdot]$. (The following equation uses *Einstein summation convention* to better illustrate the idea.)

$$\begin{aligned}
(\boldsymbol{\Xi}_2)_{ij} &= \sigma_1^2 \sum_{d=1}^{D} \{\mathrm{tr}[\boldsymbol{\Psi}_d(\boldsymbol{E}_{ij}\boldsymbol{B}^T + \boldsymbol{B}\boldsymbol{E}_{ji})]\} \\
&= \sigma_1^2 \sum_{d=1}^{D} \{\mathrm{tr}[\boldsymbol{\Psi}_d\boldsymbol{E}_{ij}\boldsymbol{B}^T + \boldsymbol{\Psi}_d\boldsymbol{B}\boldsymbol{E}_{ji}]\} \\
&= \sigma_1^2 \sum_{d=1}^{D} \{\mathrm{tr}[(\boldsymbol{\Psi}_d)_l^k(\boldsymbol{E}_{ij})_m^l(\boldsymbol{B}^T)_n^m + (\boldsymbol{\Psi}_d)_l^k(\boldsymbol{B})_m^l(\boldsymbol{E}_{ji})_n^m]\} \\
&= \sigma_1^2 \sum_{d=1}^{D} \{(\boldsymbol{\Psi}_d)_l^k(\boldsymbol{E}_{ij})_m^l(\boldsymbol{B}^T)_k^m + (\boldsymbol{\Psi}_d)_l^k(\boldsymbol{B})_m^l(\boldsymbol{E}_{ji})_k^m\} \\
&= \sigma_1^2 \sum_{d=1}^{D} \{(\boldsymbol{\Psi}_d)_i^k(\boldsymbol{B}^T)_k^j + (\boldsymbol{\Psi}_d)_l^i(\boldsymbol{B})_j^l\} \\
&= \sigma_1^2 \sum_{d=1}^{D} \{\sum_{k=1}^{N}[(\boldsymbol{\Psi}_d)_{k,i}(\boldsymbol{B}^T)_{j,k}] + \sum_{l=1}^{N}[(\boldsymbol{\Psi}_d)_{i,l}(\boldsymbol{B})_{l,j}]\} \\
&= \sigma_1^2 \sum_{d=1}^{D} \{\sum_{k=1}^{N}[(\boldsymbol{B}^T)_{j,k}(\boldsymbol{\Psi}_d)_{k,i}] + \sum_{l=1}^{N}[(\boldsymbol{\Psi}_d)_{i,l}(\boldsymbol{B})_{l,j}]\}.
\end{aligned} \tag{2.41}$$

Therefore,

$$\begin{aligned}
\boldsymbol{\Xi}_2 &= \sigma_1^2 \sum_{d=1}^{D} [(\boldsymbol{B}^T\boldsymbol{\Psi}_d)^T + \boldsymbol{\Psi}_d\boldsymbol{B}] \\
&= \sigma_1^2 \sum_{d=1}^{D} [\boldsymbol{\Psi}_d^T\boldsymbol{B} + \boldsymbol{\Psi}_d\boldsymbol{B}] \\
&= 2\sigma_1^2 \sum_{d=1}^{D} \boldsymbol{\Psi}_d\boldsymbol{B}.
\end{aligned} \tag{2.42}$$

5). Derivative with respect to $\mathbf{C}$

$$\nabla_{\mathbf{C}}\mathcal{O} = -\sum_{d=1}^{D} [\boldsymbol{\Sigma}^{-1}\mathbf{t}_d(\mathbf{x}_d - \bar{\mathbf{x}})^{\mathrm{T}}]. \tag{2.43}$$

6). Derivative with respect to $\mathbf{P}$

$$\nabla_{\mathbf{P}}\mathcal{O} = \sum_{d=1}^{D} \{\mathrm{tr}[\boldsymbol{\Psi}_d(\mathbf{E}_{ij}\mathbf{P}^{\mathrm{T}} + \mathbf{P}\mathbf{E}_{ji})]\} = 2\sum_{d=1}^{D} \boldsymbol{\Psi}_d\mathbf{P}. \tag{2.44}$$

2.6.2 Increasing Computational Speed

In this section, we discuss how to increase the speed of the optimization process for the proposed model. In the previous section, we have shown that $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{P}$, σ_1, and σ_2 are the parameters to be solved. Here, we first derive an updating scheme for σ_2 when other parameters

are fixed by following a similar technique as discussed in (Kang et al., 2008). For other parameters, we develop an efficient method for calculating the inverse of the covariance matrix which is the main bottleneck of the optimization process.

2.6.2.1 Updating σ_2

When all other parameters are fixed, using spectral decomposition on $(\sigma_1^2\mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{W}\mathbf{W}^{\mathrm{T}})$, we have

$$\begin{aligned}\mathbf{\Sigma} &= (\sigma_1^2\mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{W}\mathbf{W}^{\mathrm{T}}) + \sigma_2^2\mathbf{I}_N \\ &= [\mathbf{U}, \mathbf{V}]\,\mathrm{diag}(\lambda_1 + \sigma_2^2, ..., \lambda_{N-q} + \sigma_2^2, 0, ..., 0)[\mathbf{U}, \mathbf{V}]^{\mathrm{T}} \\ &= \mathbf{U}\,\mathrm{diag}(\lambda_1 + \sigma_2^2, ..., \lambda_{N-q} + \sigma_2^2)\mathbf{U}^{\mathrm{T}},\end{aligned} \tag{2.45}$$

where $\mathbf{U}$ is an $N \times (N-q)$ eigenvector matrix corresponding to the nonzero eigenvalues; $\mathbf{V}$ is an $N \times q$ eigenvector matrix corresponding to the zero eigenvalues. A reasonable solution should have no zero eigenvalues in $\mathbf{\Sigma}$, otherwise the loss function would be infinitely big. Therefore, $q = 0$.

Thus

$$\mathbf{\Sigma}^{-1} = \mathbf{U}\,\mathrm{diag}(\frac{1}{\lambda_1 + \sigma_2^2}, ..., \frac{1}{\lambda_N + \sigma_2^2})\mathbf{U}^{\mathrm{T}}.$$

Let $\mathbf{U}^{\mathrm{T}}(\mathbf{z}_d - \mathbf{B}\mathbf{A}\mathbf{x}_d - \mathbf{C}\mathbf{x}_d - \boldsymbol{\mu}_{\mathbf{B}}) =: [\eta_{d,1}, \eta_{d,2}, ..., \eta_{d,N}]^{\mathrm{T}}$. Then solving σ_2 is equivalent to minimizing

$$l(\sigma_2^2) = \frac{D \cdot N}{2}\log(2\pi) + \frac{D}{2}\sum_{s=1}^{N}\log(\lambda_s + \sigma_2^2) + \frac{1}{2}\sum_{d=1}^{D}\sum_{s=1}^{N}\frac{\eta_{d,s}^2}{\lambda_s + \sigma_2^2}, \tag{2.46}$$

whose derivative is

$$l'(\sigma_2^2) = \frac{D}{2}\sum_{s=1}^{N}\frac{1}{\lambda_s + \sigma_2^2} - \frac{1}{2}\sum_{d=1}^{D}\sum_{s=1}^{N}\frac{\eta_{d,s}^2}{(\lambda_s + \sigma_2^2)^2}.$$

This is a 1-dimensional optimization problem that can be solved very efficiently.

2.6.2.2 Efficiently Inverting the Covariance Matrix

From objective function Eq. 2.34 and the gradient of the parameters, the time complexity of each iteration in the optimization procedure is $\mathcal{O}(DN^2M + DN^2H + DN^3 + DNMK)$. Since $M \ll N$ and $H \ll N$, the third term of the time complexity ($\mathcal{O}(DN^3)$) is the bottleneck of

the overall performance. This is for computing the inverse of the covariance matrix

$$\mathbf{\Sigma} = \sigma_1^2 \mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{P}\mathbf{P}^{\mathrm{T}} + \sigma_2^2 \mathbf{I}_N,$$

which is much more time-consuming than other matrix multiplication operations.

We devise an acceleration strategy that calculates $\mathbf{\Sigma}^{-1}$ using formula (2.47) in the following theorem. The complexity of computing the inverse reduces to $\mathcal{O}(M^3 + H^3)$.

Theorem 1. Given $\mathbf{B} \in \mathbb{R}^{N \times M}$, $\mathbf{P} \in \mathbb{R}^{N \times H}$, and

$$\mathbf{\Sigma} = \sigma_2^2 \mathbf{I}_N + \sigma_1^2 \mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{P}\mathbf{P}^{\mathrm{T}}.$$

Then

$$\mathbf{\Sigma}^{-1} = \mathbf{T} - \mathbf{T}\mathbf{P}\mathbf{S}^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{T}, \tag{2.47}$$

where

$$\mathbf{S} = \mathbf{I}_H + \mathbf{P}^{\mathrm{T}}\mathbf{T}\mathbf{P}, \tag{2.48}$$

$$\mathbf{T} = \sigma_2^{-2}(\mathbf{I}_N - \sigma_1^2 \mathbf{B}(\sigma_2^2 \mathbf{I}_M + \sigma_1^2 \mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}). \tag{2.49}$$

The proof of Theorem 1 is provided in the following.

2.6.2.3 Preparation for Derivatives of $\mathcal{O}$ for Model 2

For notational simplicity, we denote

$$\mathbf{t}_d = (\mathbf{z}_d - \bar{\mathbf{z}}) - (\mathbf{B}\mathbf{A} + \mathbf{C})(\mathbf{x}_d - \bar{\mathbf{x}}),$$

$$\mathbf{\Psi}_d = \frac{1}{2}(\mathbf{\Sigma}^{-1} - \mathbf{\Sigma}^{-1}\mathbf{t}_d{\mathbf{t}_d}^{\mathrm{T}}\mathbf{\Sigma}^{-1}).$$

2.6.2.4 Proof of Theorem 1

Before giving the formal proof for Theorem 1, we first introduce Lemma 1, which follows from the definition of matrix inverse.

Lemma 1. For all $\mathbf{U} \in \mathbb{R}^{N\times M}$, if $\mathbf{I}_M + \mathbf{U}^{\mathrm{T}}\mathbf{U}$ is invertible, then

$$(\mathbf{I}_N + \mathbf{U}\mathbf{U}^{\mathrm{T}})^{-1} = \mathbf{I}_N - \mathbf{U}(\mathbf{I}_M + \mathbf{U}^{\mathrm{T}}\mathbf{U})^{-1}\mathbf{U}^{\mathrm{T}}.$$

Here we provide a more general proof, which can be modified to derive more involved cases.

Proof. We denote

$$\mathbf{Q} = \sigma_2^2\mathbf{I}_N + \sigma_1^2\mathbf{B}\mathbf{B}^{\mathrm{T}}, \tag{2.50}$$

that is,

$$\mathbf{\Sigma} = \sigma_2^2\mathbf{I}_N + \sigma_1^2\mathbf{B}\mathbf{B}^{\mathrm{T}} + \mathbf{P}\mathbf{P}^{\mathrm{T}} = \mathbf{Q} + \mathbf{P}\mathbf{P}^{\mathrm{T}}. \tag{2.51}$$

By Lemma 1, we have

$$\mathbf{Q}^{-1} = \mathbf{T} = \sigma_2^{-2}(\mathbf{I}_N - \sigma_1^2\mathbf{B}(\sigma_2^2\mathbf{I}_M + \sigma_1^2\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}).$$

$\mathbf{Q}$ is symmetric positive definite, hence its inverse, $\mathbf{T}$, is symmetric positive definite. Since every symmetric positive definite matrix has exactly one symmetric positive definite square root, we can write

$$\mathbf{T} = \mathbf{R}\mathbf{R},$$

where $\mathbf{R}$ is an $N \times N$ symmetric positive definite matrix.

It is clear that, $\mathbf{Q} = \mathbf{T}^{-1} = (\mathbf{R}\mathbf{R})^{-1} = \mathbf{R}^{-1}\mathbf{R}^{-1}$, which leads to $\mathbf{R}\mathbf{Q}\mathbf{R} = \mathbf{R}\mathbf{R}^{-1}\mathbf{R}^{-1}\mathbf{R} = \mathbf{I}_N$, and therefore

$$\mathbf{R}\mathbf{\Sigma}\mathbf{R} = \mathbf{I}_N + \mathbf{R}\mathbf{P}\mathbf{P}^{\mathrm{T}}\mathbf{R} = \mathbf{I}_N + \mathbf{R}\mathbf{P}\mathbf{P}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}.$$

Note that the above and the following formulas follow the fact that $\mathbf{R}$ is symmetric.

Once again, by Lemma 1, we have

$$(\mathbf{R\Sigma R})^{-1} = \mathbf{I}_N - \mathbf{RPS}^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}},$$

where

$$\mathbf{S} = \mathbf{I}_H + \mathbf{P}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{RP} = \mathbf{I}_H + \mathbf{P}^{\mathrm{T}}\mathbf{TP}.$$

Therefore,

$$\mathbf{\Sigma}^{-1} = \mathbf{R}(\mathbf{R\Sigma R})^{-1}\mathbf{R} = \mathbf{RR} - \mathbf{RRPS}^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{R},$$

and thus

$$\mathbf{\Sigma}^{-1} = \mathbf{T} - \mathbf{TPS}^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{T}$$

□

2.7 Optimization

To optimize the objective function, there are many off-the-shelf ℓ_1-penalized optimization tools. We use the Orthant-Wise Limited-memory Quasi-Newton (OWL-QN) algorithm described in (Andrew and Gao, 2007). The OWL-QN algorithm minimizes functions of the form

$$f(w) = loss(w) + c||w||_1,$$

where *loss*$(\cdot)$ is an arbitrary differentiable loss function, and $||w||_1$ is the ℓ_1-norm of the parameter vector. It is based on the L-BFGS Quasi-Newton algorithm (Nocedal and Wright, 2006), with modifications to deal with the fact that the ℓ_1-norm is not differentiable. The algorithm is proven to converge to a local optimum of the parameter vector. The algorithm is very fast, and capable of scaling efficiently to problems with millions of parameters. Thus it is a good option for our problem where the parameter space is large when dealing with large scale eQTL data.

2.8 Experimental Results

We apply our methods (SET-eQTL, $Model1$, and $Model2$) to both simulation datasets and yeast eQTL datasets (Rachel B. Brem and Kruglyak, 2005) to evaluate its performance. For comparison, we select several recent eQTL methods, including LORS (Yang et al., 2013), MTLasso2G (Chen et al., 2012), FaST-LMM (Listgarten et al., 2013) and Lasso (Tibshirani, 1996). The tuning parameters in the selected methods are learned using cross-validation. All experiments are performed on a PC with 2.20 GHz Intel i7 eight-core CPU and 8 GB memory.

2.8.1 Simulation Study

We first evaluate whether Model 2 can identify both individual and group-wise associations. We adopt a similar setup for simulation study to that in (Lee and Xing, 2012; Yang et al., 2013) and generate synthetic datasets as follows. 100 SNPs are randomly selected from the yeast eQTL dataset (Rachel B. Brem and Kruglyak, 2005). N gene expression profiles are generated by $\mathbf{Z}_{j*} = \beta_{j*}\mathbf{X} + \Xi_{j*} + \mathbf{E}_{j*}$ $(1 \leq j \leq N)$, where $\mathbf{E}_{j*} \sim N(0, \eta I)$ $(\eta = 0.1)$ denotes Gaussian noise. Ξ_{j*} is used to model non-genetic effects, which is drawn from $N(\mathbf{0}, \rho\Lambda)$, where $\rho = 0.1$. Λ is generated by $\mathbf{F}\mathbf{F}^{\mathrm{T}}$, where $\mathbf{F} \in \mathbb{R}^{D\times U}$ and $\mathbf{F}_{ij} \sim N(0, 1)$. U is the number of hidden factors and is set to 10 by default. The association matrix β is shown in the top-left plot in Figure 2.5. The association strength is 1 for all selected SNPs. There are four group-wise associations of different scales in total. The associations on the diagonal are used to represent individual association signals in *cis*-regulation.

The remaining three plots in Figure 2.5 show associations estimated by $Model2$. From the figure, we can see that $Model2$ well captures both individual and group-wise signals. For comparison, Figure 2.6 visualizes the association weights estimated by $Model1$ and $Model2$ when varying the number of hidden variables (M). We observe that for $Model1$, when $M = 20$, most of the individual association signals on the diagonal are not captured. As M increases, more individual association signals are detected by $Model1$. In contrast, $Model2$ recovers both individual and group-wise linkage signals with small M.

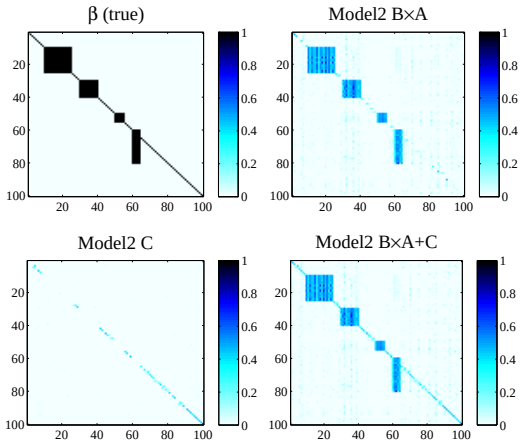


Figure 2.5: Ground truth of β and linkage weights estimated by $Model2$ on simulated data.

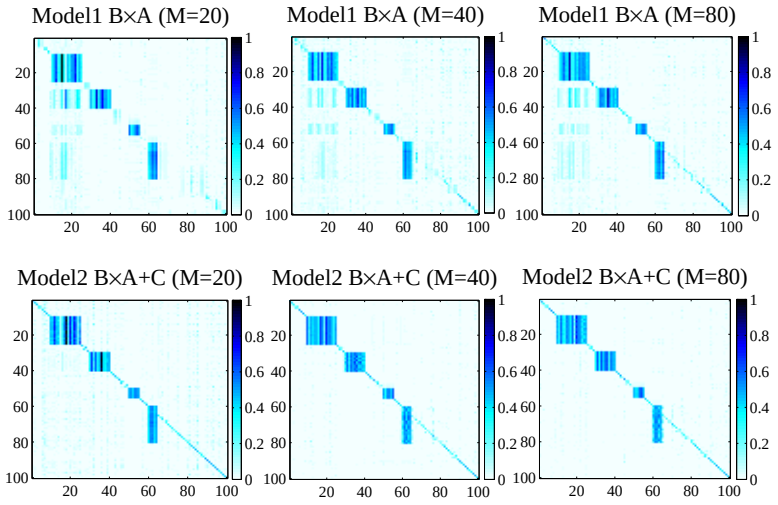
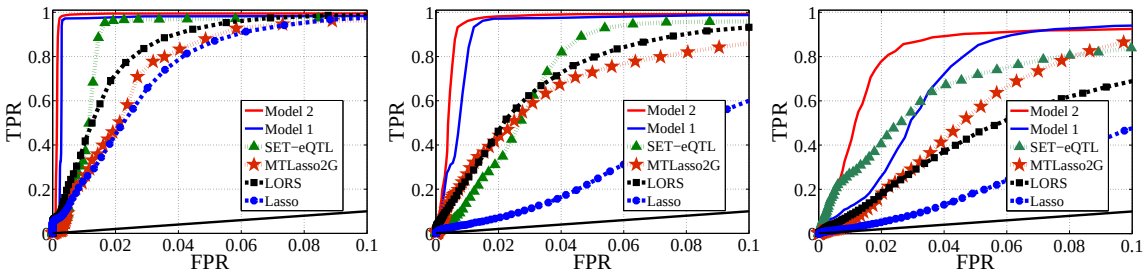


Figure 2.6: Association weights estimated by $Model1$ and $Model2$.

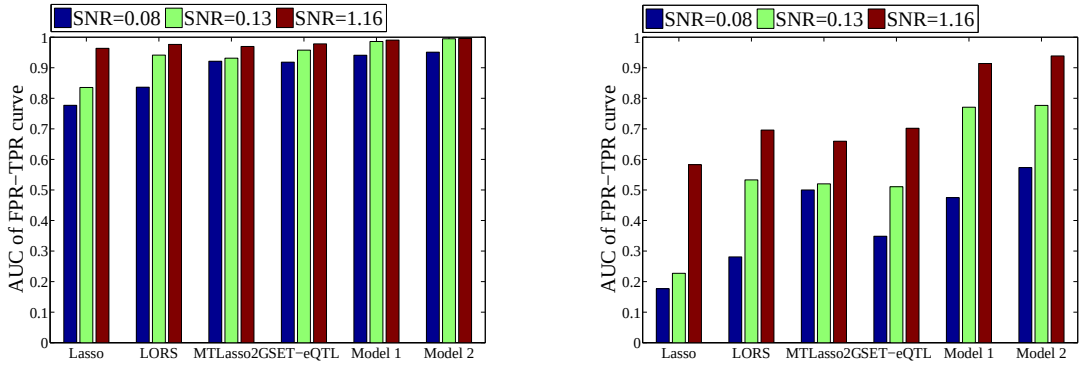
Next, we generate 50 simulated datasets with different signal-to-noise ratios (defined as $SNR = \sqrt{\frac{Var(\beta \mathbf{X})}{Var(\Xi + \mathbf{E})}}$) in the eQTL datasets (Yang et al., 2013) to compare the performance of the selected methods. Here, we fix $H = 10, \rho = 0.1$, and use different η's to control SNR. For each setting, we report the average result from the 50 datasets. For the proposed methods, we use $\mathbf{BA} + \mathbf{C}$ as the overall associations. Since FaST-LMM needs extra information (e.g., the genetic similarities between individuals) and uses PLINK format, we do not list it here and will compare it on the real data set.

Figure 2.7 shows the ROC curves of TPR-FPR for performance comparison. The corresponding areas under the TPR-FPR curve and the areas under the precision-recall curve



(a) ROC curve(SNR=1.16) (b) ROC curve(SNR=0.13) (c) ROC curve(SNR=0.08)

Figure 2.7: The ROC curve of FPR-TPR on simulated data.



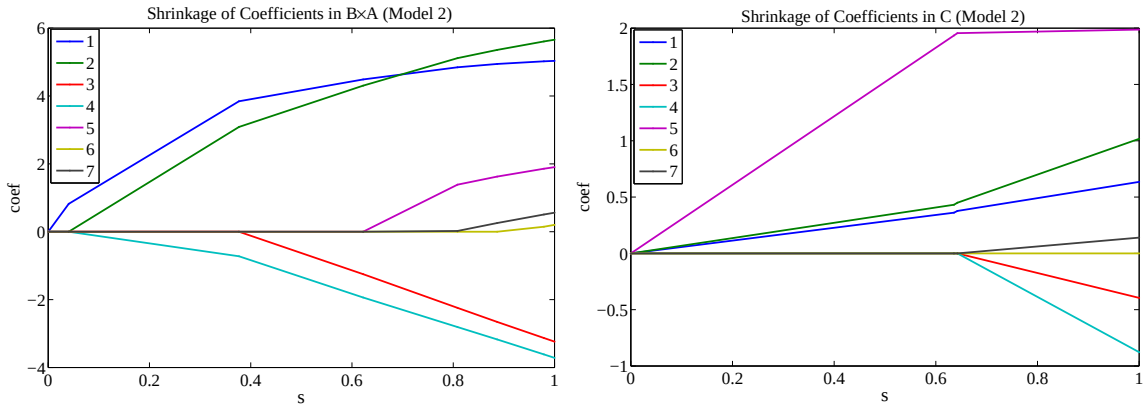
(a) AUC of FPR-TPR curve (b) AUC of precision-recall curve

Figure 2.8: The areas under the precision-recall/FPR-TPR curve (AUCs).

(AUCs) (Chen et al., 2012) are shown in Figure 2.8. It can be seen that $Model2$ outperforms all alternative methods by a large margin. $Model2$ outperforms $Model1$ because it considers both group-wise and individual associations. $Model1$ outperforms SET-eQTL because it considers confounding factors that is not considered by SET-eQTL. SET-eQTL considers all associations as group-wise, thus it may miss some individual associations. MTLasso2G is comparable to LORS because MTLasso2G considers the group-wise associations while neglecting confounding factors. LORS considers the confounding factors, but does not distinguish individual and group-wise associations. LORS outperforms Lasso since confounding factors are not considered in Lasso.

2.8.1.1 Shrinkage of C and B $\times$ A

As discussed in the previous section, the group-wise associations are encoded in $\mathbf{B} \times \mathbf{A}$ and individual associations are encoded in $\mathbf{C}$. To enforce sparsity on $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, we use Laplace



(a) $\mathbf{B} \times \mathbf{A}$ (b) $\mathbf{C}$

Figure 2.9: Model 2 shrinkage of coefficients for $\mathbf{B} \times \mathbf{A}$ and $\mathbf{C}$ respectively.

prior on the elements of these matrices. Thus, it is interesting to study the overall shrinkage of $\mathbf{B} \times \mathbf{A}$ and $\mathbf{C}$. We randomly generate 7 predictors ($\{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_7}\}$) and 1 response ($\mathbf{z}$) with sample size 100. $\mathbf{x_i} \sim N(\mathbf{0}, 0.6 \cdot \mathbf{I})(i \in [1, 7])$. The response vector was generated with the formula: $\mathbf{z} = 5 \cdot (\mathbf{x_1} + \mathbf{x_2}) - 3 \cdot (\mathbf{x_3} + \mathbf{x_4}) + 2 \cdot \mathbf{x_5} + \tilde{\epsilon}$ and $\tilde{\epsilon} \in N(\mathbf{0}, \mathbf{I})$. Thus, there are two groups of predictors ($\{\mathbf{x_1}, \mathbf{x_2}\}$ and $\{\mathbf{x_3}, \mathbf{x_4}\}$) and one individual predictor $\mathbf{x_5}$. Figure 2.9 shows the Model 2 shrinkage of coefficients for $\mathbf{B} \times \mathbf{A}$ and $\mathbf{C}$ respectively. Each curve represents a coefficient as a function of the scaled parameter $s = \frac{|\mathbf{B} \times \mathbf{A}|}{\max |\mathbf{B} \times \mathbf{A}|}$ or $s = \frac{|\mathbf{C}|}{\max |\mathbf{C}|}$. We can see that the two groups of predictors can be identified by $\mathbf{B} \times \mathbf{A}$ as the most important variables, and the individual predictor can be identified by $\mathbf{C}$.

2.8.1.2 Computational Efficiency Evaluation

Scalability is an important issue for eQTL study. To evaluate the techniques for speeding up the computational efficiency, we compare the running time with/without these techniques. Figure 2.10 shows the running time when varying the number of hidden variables (M) and number of traits (N). The results are consistent with the theoretical analysis in previous part that the time complexity is reduced to $\mathcal{O}(M^3 + H^3)$ from $\mathcal{O}(N^3)$ when using the improved method for inverting the covariance matrix. We also observe that $Model2$ uses slightly more time than $Model1$, since it has more parameters to optimize. However, to get similar performance, $Model1$ needs a significantly larger number of hidden variables M. As shown in Figure 2.10 (b), a larger

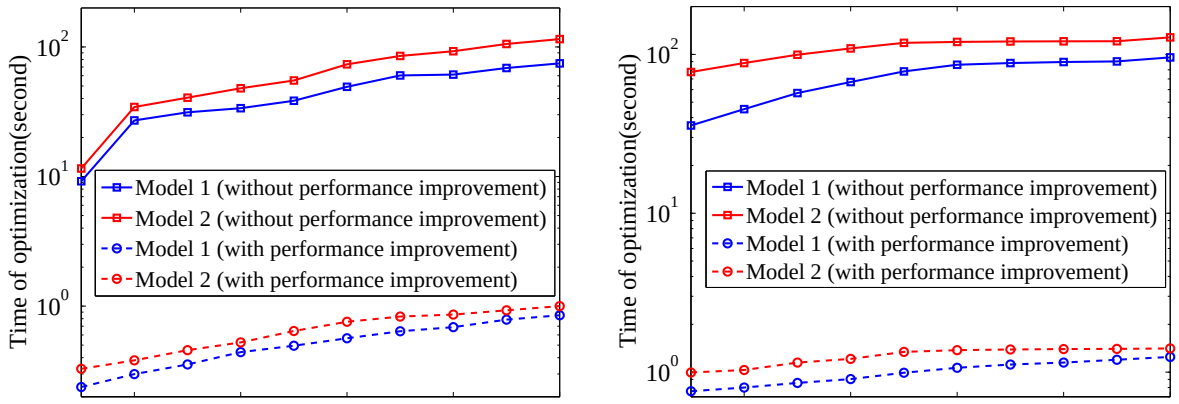


Figure 2.10: Running time performance on simulated data when varying N and M.

M results in a longer running time. In some cases, $Model2$ is actually faster than $Model1$. As an example, to obtain the same performance (i.e., AUC), $Model1$ needs 60 hidden variables (M), while $Model2$ only needs $M = 20$. In this case, from Figure 2.10 (a), we can observe that $Model2$ needs less time than $Model1$ to obtain the same results.

2.8.2 Yeast eQTL Study

We apply the proposed methods to a yeast (Saccharomyces cerevisiae) eQTL dataset of 112 yeast segregants generated from a cross of two inbred strains (Rachel B. Brem and Kruglyak, 2005). The dataset originally includes expression profiles of 6229 gene expression traits and genotype profiles of 2956 SNP markers. After removing SNPs with more than 10% missing values and merging consecutive SNPS with high linkage disequilibrium, we obtain 1017 SNPs with distinct genotypes (Huang et al., 2009a). In total, 4474 expression profiles are selected after removing the ones with missing values. It takes about 5 hours for $Model1$, and 3 hours for $Model2$ to run to completion. The regularization parameters are set by grid search in $\{0.1, 1, 10, 50, 100, 500, 1000, 2000\}$. Specifically, grid search trains the model with each combination of three regularization parameters in the grid and evaluates their performance (by measuring out-of-sample loss function value) for a two-fold cross validation. Finally, the grid search algorithm outputs the settings that achieved the smallest loss in the validation procedure.

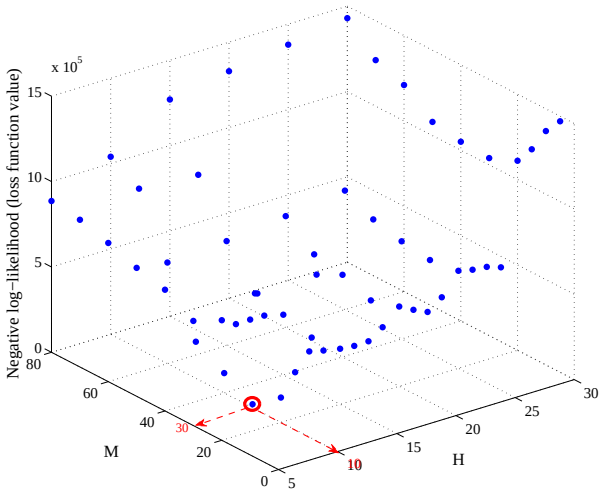
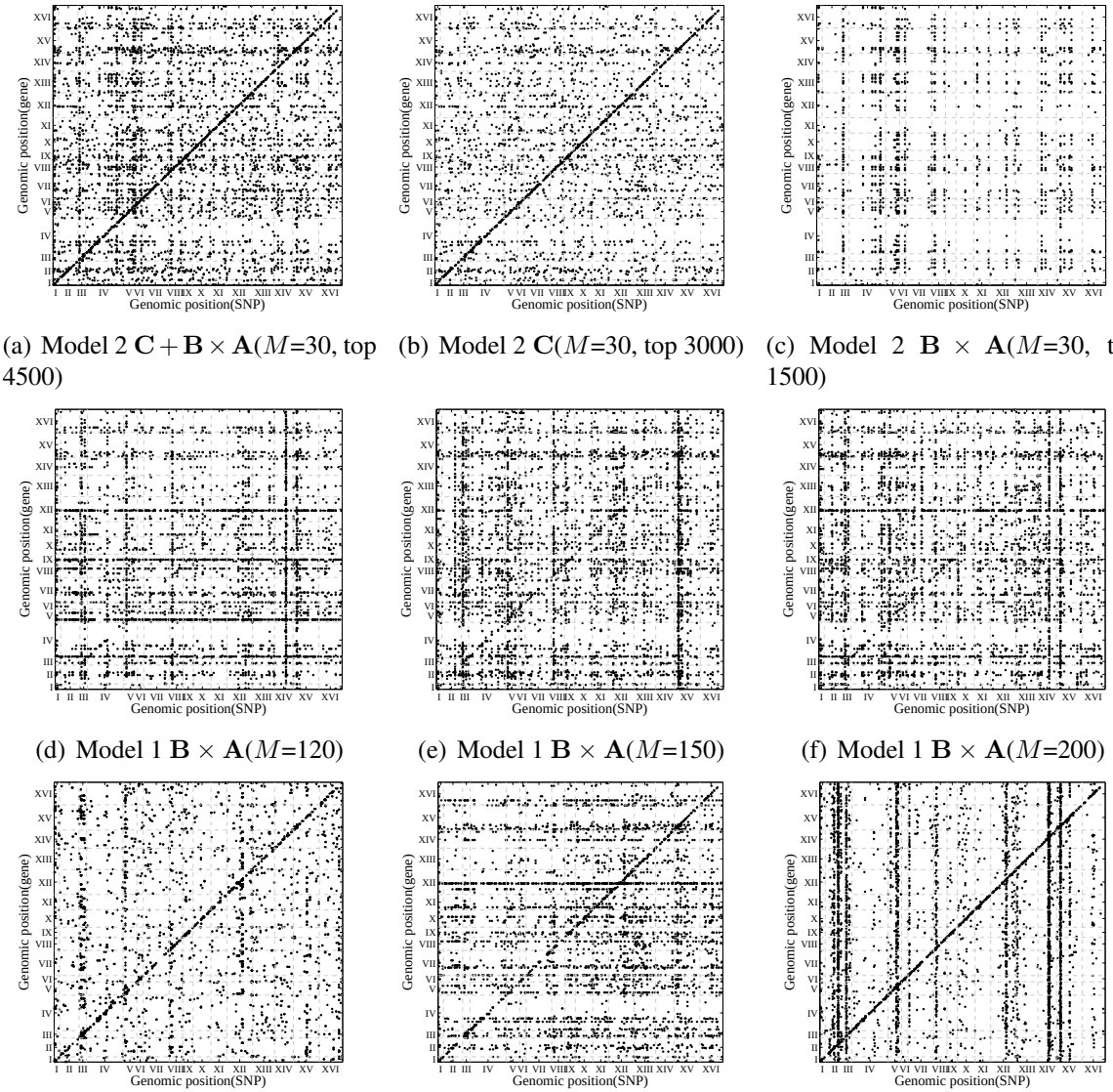


Figure 2.11: Parameter tuning for M and H ($Model2$)

We use hold-out validation to find the optimal number of hidden variables M and H for each model. Specifically, we partition the samples into 2 subsets of equal size. We use one subset as training data and test the learned model using the other subset of samples. By measuring out-of-sample predictions, we can find optimal combination of M and H that avoids over-fitting. For each combination, optimal values for regularization parameters were determined with two-fold cross validation. The loss function values for different $\{M, H\}$ combinations of $Model2$ are shown in Figure 2.11. We find that M=30 and H=10 for $Model2$ delivers the best overall performance. Similarly, we find that the optimal M and H values for $Model1$ are 150 and 10 respectively. The significant associations given by $Model1$, $Model2$, LORS, MTLasso2G and Lasso are shown in Figure 2.12. For $Model2$, we can clearly see that the estimated matrices $\mathbf{C}$ and $\mathbf{B} \times \mathbf{A}$ well capture the non group-wise and group-wise signals respectively. $\mathbf{C} + \mathbf{B} \times \mathbf{A}$ and $\mathbf{C}$ of $Model2$ have stronger *cis*-regulatory signals and weaker *trans*-regulatory bands than that of $Model1$, LORS, and Lasso. $\mathbf{C}$ of $Model2$ has the weakest *trans*-regulatory bands. LORS has weaker *trans*-regulatory bands than Lasso since it considers confounding factors. With more hidden variables (larger M), $Model1$ obtains stronger *cis*-regulatory signals.



(a) Model 2 $\mathbf{C}+\mathbf{B}\times\mathbf{A}$($M$=30, top 4500)

(b) Model 2 $\mathbf{C}$(M=30, top 3000)

(c) Model 2 $\mathbf{B}\times\mathbf{A}$($M$=30, top 1500)

(d) Model 1 $\mathbf{B}\times\mathbf{A}$($M$=120)

(e) Model 1 $\mathbf{B}\times\mathbf{A}$($M$=150)

(f) Model 1 $\mathbf{B}\times\mathbf{A}$($M$=200)

(g) MTLasso2G

(h) LORS

(i) Lasso

Figure 2.12: Significant associations discovered by different methods in yeast.

2.8.2.1 cis- and trans- Enrichment Analysis

In total, the proposed two methods detect about 6000 associations with non-zero weight values ($\mathbf{B}\times\mathbf{A}$ for $Model1$ and $\mathbf{C}+\mathbf{B}\times\mathbf{A}$ for $Model2$). We estimate their FDR values by following the method proposed in (Yang et al., 2013). With FDR ≤ 0.01, both models obtain about 4500 associations. The visualization of significant associations detected by different methods is provided in Figure 2.12.

We apply *cis*- and *trans*-enrichment analysis on the discovered associations. In particular, we follow the standard *cis*-enrichment analysis (Listgarten et al., 2010; McClurg et al., 2007) to compare the performance of two competing models. The intuition behind *cis*-enrichment analysis is that more *cis*-acting SNPs are expected than *trans*-acting SNPs. A two-step procedure is used in the *cis*-enrichment analysis (Listgarten et al., 2010): (1) for each model, we apply a one-tailed Mann-Whitney test on each SNP to test the null hypothesis that the model ranks its *cis* hypotheses (we use <500bp for yeast) no better than its *trans* hypotheses, (2) for each pair of models compared, we perform a two-tailed paired Wilcoxon sign-rank test on the *p*-values obtained from the previous step. The null hypothesis is that the median difference of the *p*-values in the Mann-Whitney test for each SNP is zero. The *trans*-enrichment is implemented using a similar strategy as in (Yvert et al., 2003), in which genes regulated by transcription factors are used as *trans*-acting signals.

The results of pairwise comparison of selected models are shown in Table 2.2. A p-value shows how significant a method on the left column outperforms a method in the top row in terms of *cis*-enrichment or *trans*-enrichment. We observe that the proposed $Model2$ has significantly better *cis*-enrichment scores than other methods. For *trans*-enrichment, $Model2$ is the best, and FaST-LMM comes in second. This is because both $Model2$ and FaST-LMM consider confounding factors (FaST-LMM considers confounders from population structure) and joint effects of SNPs, but only $Model2$ considers grouping of genes. $Model1$ has poor performance because a larger M may be needed for $Model1$ to capture those individual associations.

		FaST-LMM	$\mathbf{C}$ of $Model2$	SET-eQTL	MTLasso2G	$\mathbf{B} \times \mathbf{A}$ of $Model1$	LORS	Lasso
cis-enrichment	$\mathbf{C} + \mathbf{B} \times \mathbf{A}$ of $Model2$	0.4351	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
	FaST-LMM	-	0.2351	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
	$\mathbf{C}$ of $Model2$	-	-	0.0253	0.0221	< 0.0001	< 0.0001	< 0.0001
	SET-eQTL	-	-	-	0.0117	< 0.0001	< 0.0001	< 0.0001
	MTLasso2G	-	-	-	-	< 0.0001	< 0.0001	< 0.0001
	$\mathbf{B} \times \mathbf{A}$ of $Model1$	-	-	-	-	-	< 0.0001	< 0.0001
	LORS	-	-	-	-	-	-	0.0052
		$\mathbf{B} \times \mathbf{A}$ of $Model2$	FaST-LMM	MTLasso2G	LORS	$\mathbf{B} \times \mathbf{A}$ of $Model1$	SET-eQTL	Lasso
trans-enrichment	$\mathbf{C} + \mathbf{B} \times \mathbf{A}$ of $Model2$	0.4245	0.3123	0.0034	0.0029	0.0027	0.0025	0.0023
	$\mathbf{B} \times \mathbf{A}$ of $Model2$	-	0.3213	0.0132	0.0031	0.0028	0.0027	0.0026
	FaST-LMM	-	-	0.0148	0.0033	0.0031	0.003	0.0029
	MTLasso2G	-	-	-	0.0038	0.0037	0.0036	0.0032
	LORS	-	-	-	-	0.0974	0.0387	0.0151
	$\mathbf{B} \times \mathbf{A}$ of $Model1$	-	-	-	-	-	0.0411	0.0563
	SET-eQTL	-	-	-	-	-	-	0.0578

Table 2.2: Pairwise comparison of different models using *cis*- and *trans*- enrichment.

2.8.2.2 Reproducibility of trans Regulatory Hotspots between Studies

We also evaluate the consistency of calling eQTL hotspots between two independent glucose yeast datasets (Smith and Kruglyak, 2008). The glucose environment from Smith et al. (Smith and Kruglyak, 2008) shares a common set of segregants. It includes 5493 probes measured in 109 segregates. Since our algorithm aims at finding group-wise associations, we focus on the consistency of regulatory hotspots.

We examine the reproducibility of *trans* regulatory hotspots based on the following criteria (Fusi et al., 2012; Yang et al., 2013; Joo et al., 2014). For each SNP, we count the number of associated genes from the detected SNP-gene associations. We use this number as the regulatory degree of each SNP. For Model2, LORS, and Lasso, all SNP-Gene pairs with non-zero association weights are defined as associations. Note that Model2 uses $\mathbf{BA} + \mathbf{C}$ as the overall associations. For FaST-LMM, SNP-Gene pairs with a q-value < 0.001 are defined as associations. Note that we also tried different cutoffs for FaST-LMM (from 0.01 to 0.001), the results are similar. SNPs with large regulatory degrees are often referred to as hotspots. We sort SNPs by the extent of *trans* regulation (regulatory degrees) in a descending order. We denote the sorted SNPs lists as S_1 and S_2 for the two yeast datasets. Let S_1^T and S_2^T be the top T SNPs in the sorted SNP lists. The trans calling consistency of detected hotspots is defined as $\frac{|S_1^T \bigcap S_2^T|}{T}$. Figure 2.13

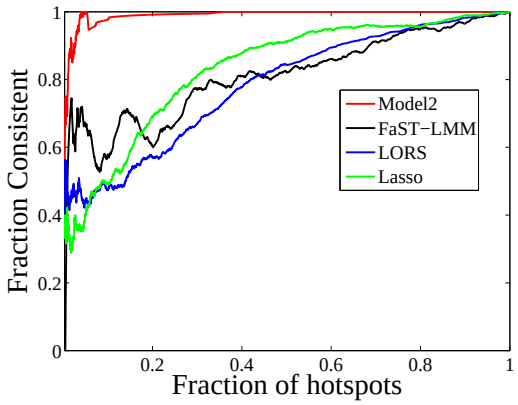


Figure 2.13: Consistency of detected eQTL hotspots

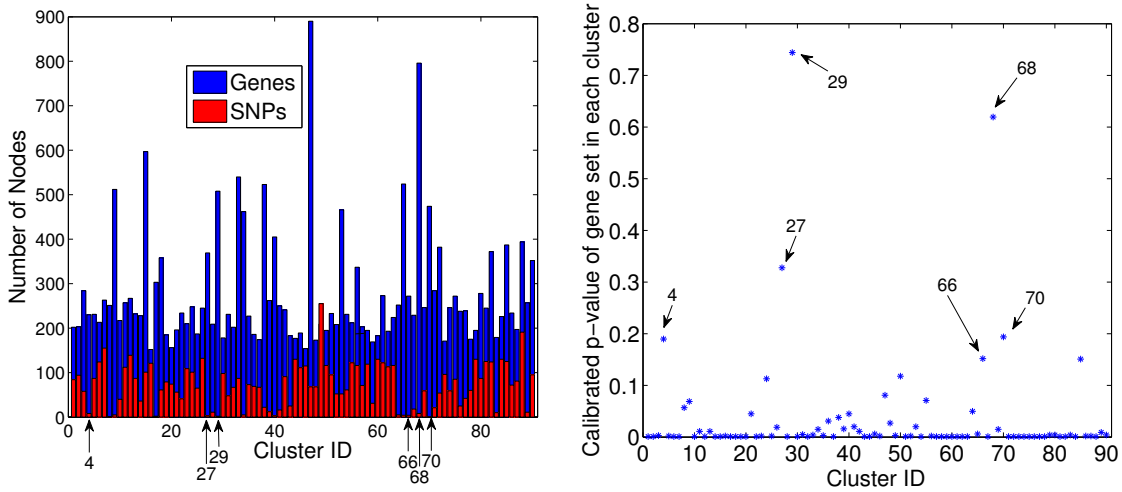
compares the reproducibility of *trans* regulatory hotspots given by different studies. It can be seen

that the proposed Model2 gives much higher consistency than any other competitors do. In particular, the consistency of *trans* hotspots suggests the superiority of Model2 in identifying hotspots that are likely to have a true genetic underpinning.

2.8.2.3 Gene Ontology Enrichment Analysis

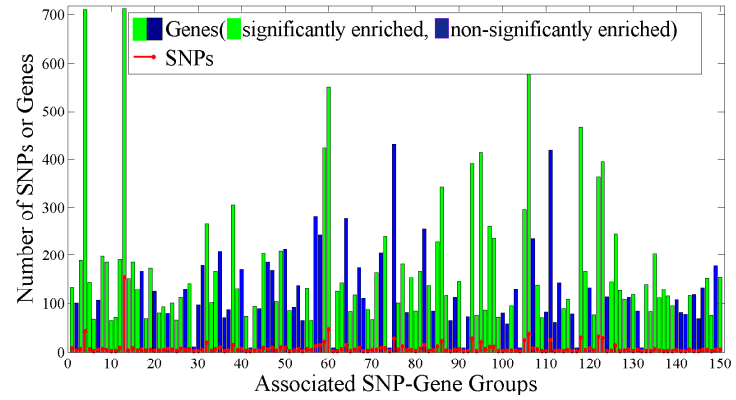
As discussed in previous section, hidden variables $\mathbf{y}$ in the middle layer may model the joint effect of SNPs that have influence on a group of genes. To better understand the learned model, we look for correlations between a set of genes associated with a hidden variable and GO categories (Biological Process Ontology) (The Gene Ontology Consortium, 2000). In particular, for each gene set G, we identify the GO category whose set of genes is most correlated with G. We measure the correlation by a p-value determined by the Fisher's exact test. Since multiple gene sets G need to be examined, the raw p-values need to be calibrated because of the multiple testing problem (Westfall and Young, 1993). To compute the calibrated p-values for each gene set G, we perform a randomization test, wherein we apply the same test to randomly created gene sets that have the same number of genes as G. Specifically, the enrichment test is performed using DAVID (Huang et al., 2009a). And gene sets with calibrated p-values less than 0.01 are considered as significantly enriched. The results from $Model2$ are reported in Table 2.3. Each row of Table 2.3 represents the gene set associated with a hidden variable. All of these detected gene sets are significantly enriched in certain GO categories. The significantly enriched gene sets of $Model1$ and SET-eQTL are included in Table 2.4, Table 2.5, and Table 2.7, respectively. In total, 77 out of 90 gene sets detected by SET-eQTL are significant. For SET-eQTL, Figure 2.14 shows the number of genes and SNPs within each group-wise association and the corresponding calibrated p-value (Fisher's exact test) of each discovered gene set. The hidden variable IDs are used as the cluster IDs. We can observe that for SET-eQTL, the gene sets with large calibrated p-values tend to have a very small SNP set associated with them. Those clusters are labeled in both figures. This is a strong indicator that these hidden variables may correspond to confounding factors.

For comparison, we visualize the number of SNPs and genes in each group-wise association in Figure 2.15. We observe that 90 out of 150 gene sets reported by $Model1$ are

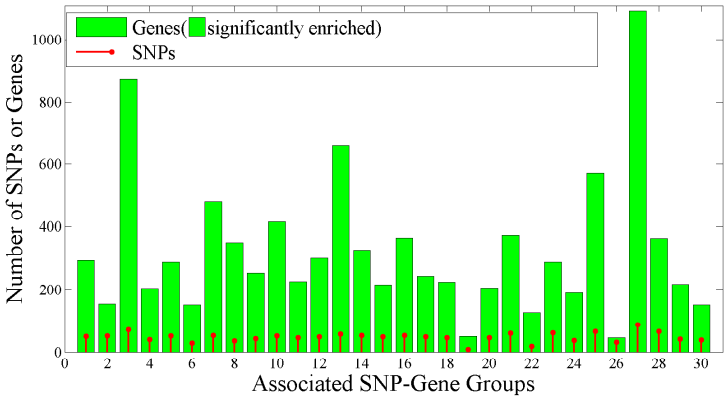


(a) #nodes in each set (SNP/gene) (b) Calibrated p-values of gene sets

Figure 2.14: Number of nodes and calibrated p-values in each group-wise association



(a) $Model1$(150 groups)



(b) $Model2$(30 groups)

Figure 2.15: Number of SNPs and genes in each group-wise association.

[a]Group ID	[b]SNPs set size	[c]gene set size	[d]GO category
1	63	294	oxidation-reduction process*
2	78	153	thiamine biosynthetic process*
3	94	871	rRNA processing***
4	64	204	nucleosome assembly**
5	70	288	ATP synthesis coupled proton transport***
6	43	151	branched chain family amino acid biosynthetic...**
7	76	479	mitochondrial translation***
8	47	349	transmembrane transport**
9	64	253	cytoplasmic translation***
10	72	415	response to stress**
11	64	225	mitochondrial translation*
12	62	301	oxidation-reduction process**
13	83	661	oxidation-reduction process*
14	69	326	cytoplasmic translation*
15	71	216	oxidation-reduction process*
16	66	364	methionine metabolic process*
17	74	243	cellular amino acid biosynthetic process***
18	63	224	transmembrane transport**
19	23	50	de novo' pyrimidine base biosynthetic process*
20	66	205	cellular amino acid biosynthetic process***
21	81	372	oxidation-reduction process**
22	33	126	oxidation-reduction process***
23	81	288	pheromone-dependent signal transduction...**
24	53	190	pheromone-dependent signal transduction...**
25	91	572	oxidation-reduction process***
26	66	46	cellular cell wall organization*
27	111	1091	translation***
28	89	362	cellular amino acid biosynthetic process**
29	62	217	transmembrane transport**
30	71	151	cellular aldehyde metabolic process**

Table 2.3: Summary of all detected groups of genes from $Model2$ on yeast data.

significantly enriched, and all 30 gene sets reported by $Model2$ are significantly enriched. This indicates that $Model2$ is able to detect group-wise linkages more precisely than $Model1$. We also study the hotspots detected by LORS, which affect > 10 gene traits (Lee and Xing, 2012). Specifically, we delve into the top 15 hotspots detected by LORS (ranking by number of associated genes for each SNP), as listed in Table 2.6. We can see that only 9 out of 15 top ranked hotspots are significantly enriched.

2.9 Conclusion

A crucial challenge in eQTL study is to understand how multiple SNPs interact with each other to jointly affect the expression level of genes. In this chapter, we propose three sparse graphical model based approaches to identify novel group-wise eQTL associations. ℓ_1-regularization is applied to learn the sparse structure of the graphical model. The three models incrementally take into consideration more aspects, such as group-wise association, potential confounding factors and the existence of individual associations. We illustrate how each aspect would benefit the eQTL mapping. We also introduce computational techniques to make this

[a]Group ID	[b]SNPs set size	[c]gene set size	[d]GO category
1	8	134	branched chain family amino acid biosynthetic process**
3	6	189	oxidation-reduction process***
4	43	710	cytoplasmic translation***
5	6	144	ion transport*
6	2	69	arginine biosynthetic process*
8	6	197	cellular amino acid biosynthetic process**
9	4	185	transmembrane transport*
10	2	66	cellular response to nitrogen starvation*
11	2	73	cellular response to nitrogen starvation*
12	9	191	pheromone-dependent signal transduction involved in conjugation with cellular fusion*
13	154	712	cytoplasmic translation***
14	3	151	amino acid catabolic process to alcohol via Ehrlich pathway*
15	8	185	oxidation-reduction process**
16	3	130	arginine biosynthetic process*
18	3	70	arginine biosynthetic process*
19	5	173	cellular amino acid biosynthetic process*
21	3	81	cellular aldehyde metabolic process*
22	4	93	cellular amino acid biosynthetic process**
24	5	101	iron ion homeostasis*
25	2	67	cellular amino acid metabolic process**
26	7	112	oxidation-reduction process*
28	6	141	oxidation-reduction process*
32	19	265	cellular amino acid biosynthetic process*
33	3	102	glycogen biosynthetic process*
34	6	166	oxidation-reduction process*
38	15	305	cellular amino acid biosynthetic process***
39	4	131	telomere maintenance via recombination**
41	2	75	cellular response to nitrogen starvation*
43	3	94	cellular response to nitrogen starvation*
45	9	205	cellular amino acid biosynthetic process*
48	3	104	telomere maintenance via recombination*
49	10	210	oxidation-reduction process*
51	2	86	cellular aldehyde metabolic process*
55	6	132	cytogamy*
56	4	66	cellular cell wall organization*
59	21	425	methionine biosynthetic process*
60	46	551	cellular amino acid biosynthetic process**
62	2	124	ion transport**
63	6	143	iron ion homeostasis*
65	2	84	cellular response to nitrogen starvation*
66	5	117	transposition, RNA-mediated*
69	2	88	one-carbon metabolic process*
70	4	68	cellular response to nitrogen starvation*
71	5	164	oxidation-reduction process*
73	8	240	cellular amino acid biosynthetic process**
76	5	101	cellular response to nitrogen starvation*
77	12	181	mitochondrial electron transport, ubiquinol to cytochrome c***
79	4	153	cellular amino acid biosynthetic process**
80	2	85	hexose transport*
81	7	166	oxidation-reduction process**
83	2	137	cellular amino acid biosynthetic process*
85	12	228	cellular amino acid biosynthetic process***
86	22	342	cellular amino acid biosynthetic process*
87	3	116	cellular amino acid biosynthetic process*
90	6	146	hexose transport*

Table 2.4: Summary of detected significantly enriched gene groups from $Model1$ (Part I).

approach suitable for large scale studies. Extensive experimental evaluations using both simulated and real datasets demonstrate that the proposed methods can effectively capture both individual and group-wise signals and significantly outperform the state-of-the-art eQTL mapping methods.

[a]Group ID	[b]SNPs set size	[c]gene set size	[d]GO category
93	28	391	ATP synthesis coupled proton transport**
94	2	76	oxidation-reduction process**
95	20	414	nucleosome assembly*
96	6	87	cellular response to nitrogen starvation*
97	11	260	oxidation-reduction process*
98	11	236	mitochondrial electron transport, ubiquinol to cytochrome c*
99	2	73	cellular response to nitrogen starvation*
102	3	95	cellular aldehyde metabolic process**
105	24	296	cellular amino acid biosynthetic process**
106	37	651	oxidation-reduction process***
108	6	138	oxidation-reduction process*
109	2	72	siderophore transport*
114	2	90	amino acid transmembrane transport*
115	4	108	arginine biosynthetic process*
118	30	467	cellular amino acid biosynthetic process***
119	4	166	methionine biosynthetic process**
121	3	77	iron ion homeostasis*
122	31	364	cellular amino acid biosynthetic process***
123	29	395	cellular amino acid biosynthetic process***
125	3	145	cellular amino acid biosynthetic process**
126	14	244	cellular response to nitrogen starvation*
127	2	126	cellular amino acid biosynthetic process*
128	3	108	telomere maintenance via recombination*
130	2	118	oxidation-reduction process*
133	2	139	cell adhesion*
134	2	84	cell adhesion*
135	6	204	oxidation-reduction process**
136	3	111	arginine biosynthetic process*
137	2	129	response to pheromone**
138	2	115	transmembrane transport*
139	2	95	cellular aldehyde metabolic process*
143	5	116	cellular amino acid biosynthetic process*
147	4	152	mitochondrial electron transport, ubiquinol to cytochrome c**
148	2	76	cellular aldehyde metabolic process**
150	5	154	fermentation*

Table 2.5: Summary of detected significantly enriched gene groups from $Model1$ (Part II).

chr	start	end	size	GO category	adjusted p-value
XII	659357	662627	36	sterol biosynthetic process	7.18E-05
XII	1056097	1056097	31	telomere maintenance via recombination	4.72E-08
XV	**154177**	**154309**	**29**	**amino acid catabolic process to alcohol via Ehrlich pathway**	**0.052947053**
III	201166	201167	23	regulation of mating-type specific transcription, DNA-dependent	0.001998002
XV	**143597**	**150651**	**23**	**response to stress**	**0.672327672**
III	81832	92391	22	pheromone-dependent signal transduction involved in conjugation with cellular fusion	1.76E-03
VIII	111682	111690	22	cell adhesion	0.002947528
IX	139462	139512	21	cellular response to nitrogen starvation	0.00106592
XV	**170945**	**180961**	**20**	**cell adhesion**	**0.053946054**
III	105042	105042	19	branched chain family amino acid biosynthetic process	5.51357E-08
XIII	**46070**	**46084**	**19**	**cell adhesion**	**0.050949051**
XV	563943	563943	19	transport	0.003996004
I	**41483**	**42639**	**18**	**cellular response to nitrogen starvation**	**0.016983017**
III	175799	177850	18	pheromone-dependent signal transduction involved in conjugation with cellular fusion	7.47E-03
I	**36900**	**37068**	**17**	**signal transduction**	**0.547452547**

Table 2.6: Summary of the top 15 detected hotspots by LORS.

[a]Group ID	[b]SNPs set size	[c]gene set size	[d]GO category
75	84	272	cellular amino acid biosynthetic process***
74	94	246	cellular amino acid biosynthetic process***
62	124	193	cellular amino acid biosynthetic process***
17	155	303	oxidation-reduction process***
78	40	175	sterol biosynthetic process***
81	139	245	oxidation-reduction process***
88	36	394	cellular amino acid biosynthetic process***
18	101	358	oxidation-reduction process***
1	2	202	cellular amino acid biosynthetic process***
2	61	203	cellular amino acid biosynthetic process***
76	79	238	oxidation-reduction process***
10	74	217	cellular aldehyde metabolic process***
51	41	233	transmembrane transport***
19	11	185	oxidation-reduction process***
37	98	174	cellular amino acid biosynthetic process***
77	67	239	arginine biosynthetic process***
20	67	156	transmembrane transport***
71	25	284	oxidation-reduction process***
58	130	195	oxidation-reduction process***
61	95	273	oxidation-reduction process***
6	61	213	oxidation-reduction process***
32	71	202	oxidation-reduction process***
30	119	178	arginine biosynthetic process***
67	31	229	response to stress***
43	130	183	arginine biosynthetic process***
22	122	234	lysine biosynthetic process***
7	113	263	cellular amino acid biosynthetic process***
57	116	203	transmembrane transport***
14	18	228	oxidation-reduction process***
54	21	231	oxidation-reduction process***
72	54	382	cellular aldehyde metabolic process***
59	96	169	cellular amino acid biosynthetic process***
44	59	177	pentose-phosphate shunt***
73	85	171	transmembrane transport***
15	25	597	cytoplasmic translation***
12	42	267	cellular amino acid biosynthetic process***
28	60	209	cellular amino acid biosynthetic process***
82	125	372	mitochondrial translation***
60	124	183	cellular amino acid biosynthetic process***
63	130	224	oxidation-reduction process***
84	191	226	cellular amino acid biosynthetic process***
5	87	231	arginine biosynthetic process***
23	121	210	oxidation-reduction process***
86	109	234	telomere maintenance via recombination***
16	65	152	arginine biosynthetic process**
52	115	208	lysine biosynthetic process**
46	52	154	cellular response to nitrogen starvation**
87	116	197	response to stress**
56	72	337	mitochondrial electron transport, ubiquinol to cytochrome c**
25	81	187	telomere maintenance via recombination**
35	58	227	transmembrane transport**
49	73	209	pyrimidine nucleotide biosynthetic process***
3	255	284	telomere maintenance via recombination**
83	87	179	iron ion homeostasis**
33	130	540	translation**
80	87	278	telomere maintenance via recombination**
90	10	352	ion transport**
79	95	195	cellular response to nitrogen starvation**
31	48	231	arginine biosynthetic process***
45	111	189	oxidation-reduction process**
65	4	524	lysine biosynthetic process**
89	11	257	cellular response to nitrogen starvation**
13	112	232	cellular amino acid biosynthetic process**
42	87	241	cellular response to nitrogen starvation**
11	91	257	oxidation-reduction process**
34	5	462	cellular amino acid biosynthetic process**
69	59	246	cellular amino acid biosynthetic process**
39	12	262	response to stress**
26	132	245	cellular amino acid biosynthetic process**
41	16	250	cellular response to nitrogen starvation**
53	52	466	cytoplasmic translation**
48	68	173	cellular aldehyde metabolic process*
36	69	186	oxidation-reduction process*
38	22	523	pheromone-dependent signal transduction involved in conjugation with cellular fusion*
40	56	405	cellular amino acid metabolic process*
21	3	196	arginine biosynthetic process*
64	5	252	one-carbon metabolic process*

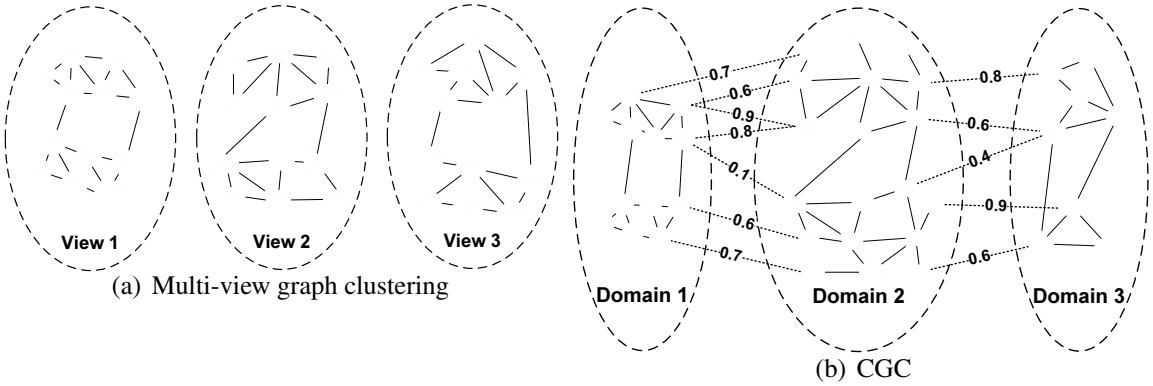
Table 2.7: Summary of detected significantly enriched gene groups from SET-eQTL.

CHAPTER 3: REFINING PRIOR GROUPING INFORMATION

3.1 Introduction

Much prior knowledge about the relationships between SNPs and relationships between genes can be modeled as networks (or graphs). Biologists believe that a set of SNPs may play joint roles in a disease. Such interactions between SNPs can be modeled by a SNP interaction network. Even though the underlying biological processes are complex and only partially solved, it is well established that SNPs may alter the expression levels of related genes which may in turn have a cascading effect to other genes, e.g., in the same biological pathways (Michaelson et al., 2009c). The interactions between genes can be measured by correlations of gene expressions and represented by a gene interaction network. These two networks are heavily related because of the complicated relationships between SNPs and genes, as demonstrated in many expression quantitative trait loci (eQTL) studies. It is evident that a joint analysis becomes essential in these related domains. Conducting graph clustering jointly on the multiple networks, e.g., SNP-SNP interaction network, PPI network, and gene co-expression network, provides more accurate prior knowledge about the grouping information of SNPs and genes. Data collected from different sources provides complimentary predictive powers, and combining expertise from these different sources can resolve ambiguity, thus helping to obtain more *accurate and robust* decisions for establishing knowledge bases.

In literature, the integration of multiple networks for clustering has been well studied. This task is usually referred as *multi-view* graph clustering. By exploiting multi-domain information to refine clustering and resolve ambiguity, multi-view graph clustering methods have the potential to dramatically increase the accuracy of the final results (Bickel and Scheffer, 2004; Kumar et al., 2011; Chaudhuri et al., 2009). The key assumption of these methods is that the



(a) Multi-view graph clustering

(b) CGC

Figure 3.1: Multi-view graph clustering vs co-regularized multi-domain graph clustering (CGC)

same set of data instances may have multiple representations, and different views are generated from the same underlying distribution (Chaudhuri et al., 2009). These views should agree on a consensus partition of the instances that reflects the hidden ground truth (Long et al., 2008). The learning objective is thus to find the most consensus clustering structure across different domains.

Existing multi-view graph clustering methods usually assume that information collected in different domains is for the same set of instances. Thus, the cross-domain instance relationships are strictly *one-to-one*. This also implies that different views are of the same size. For example, Figure 3.1 (a) shows a typical scenario of multi-view graph clustering, where the same set of 12 data instances has 3 different views. Each view gives a different graph representation of the instances.

However, for the eQTL mapping application, it is common to have cross-domain relationships as shown in Figure 3.1 (b). This example illustrates several key properties that are different from the traditional multi-view graph clustering scenario.

- An instance in one domain may be mapped to multiple instances in another domain. For example, in Figure 3.1 (b), instance Ⓐ in domain 1 is mapped to two instances ① and ② in domain 2. The cross-domain relationship is many-to-many rather than one-to-one. For example, if we want to integrate protein-protein interaction (PPI) networks

(Asur et al., 2007), multiple proteins may be synthesized from one gene and one gene may contain many genetic variants.

- Mapping between cross-domain instances may be associated with weights, which is a generalization of a binary relationship. As shown in Figure 3.1 (b), each cross-domain mapping is coupled with a weight. Users may specify these weights based on their prior knowledge.
- The cross-domain instance relationship may be a partial mapping. Graphs in different domains may have different sizes. Some instance in one domain may not have corresponding instance in another. As shown in Figure 3.1 (b), mapping between instances in different domains is not complete.

In this chapter, we propose CGC (Co-regularized Graph Clustering), a flexible and robust approach that is able to incorporate multiple sources to enhance graph clustering performance. The proposed approach is robust even when the cross-domain relationships based on prior knowledge are noisy. Besides, the model provides users with the extent to which the cross-domain instance relationship violates the in-domain clustering structure, and thus enables users to re-evaluate the consistency of the relationship. The chapter further studies the trustworthiness of multi-source data, and extended the approach to enable it to automatically identify noisy domains and assign smaller weights to them for integration.

3.2 The Problem

Suppose that we have d graphs, each from a domain in $\{\mathcal{D}_1,\mathcal{D}_2,...,\mathcal{D}_d\}$. We use n_π to denote the number of instances (nodes) in the graph from domain $\mathcal{D}_\pi$ ($1 \leq \pi \leq d$). Each graph is represented by an affinity (similarity) matrix. The affinity matrix of the graph in domain $\mathcal{D}_\pi$ is denoted as $\mathbf{A}^{(\pi)} \in \mathbb{R}_+^{n_\pi \times n_\pi}$. In this chapter, we follow the convention and assume that $\mathbf{A}^{(\pi)}$ is a symmetric and non-negative matrix (Ng et al., 2001; Kuang et al., 2012). We denote the set of pairwise cross-domain relationships as $\mathcal{I} = \{(i,j)\}$ where i and j are domain indices. For example, $\mathcal{I} = \{(1,3),(2,5)\}$ contains two cross-domain relationships (mappings): the relationship between instances in $\mathcal{D}_1$ and $\mathcal{D}_3$, and the relationship between instances in $\mathcal{D}_2$ and $\mathcal{D}_5$.

Symbols	Description
d	The number of domains
$\mathcal{D}_\pi$	The π-th domain
n_π	The number of instances in the graph from $\mathcal{D}_\pi$
k_π	The number of clusters in $\mathcal{D}_\pi$
$\mathbf{A}^{(\pi)}$	The affinity matrix of graph in $\mathcal{D}_\pi$
$\mathcal{I}$	The set of cross-domain relationships
$\mathbf{S}^{(i,j)}$	The relationship matrix between instances in $\mathcal{D}_i$ and $\mathcal{D}_j$
$\mathbf{W}^{(i,j)}$	The confidence matrix of relationship matrix $\mathbf{S}^{(i,j)}$
$\mathbf{H}^{(\pi)}$	The clustering indicator matrix of $\mathcal{D}_\pi$
α	Confidence threshold of finding the global
c_ϕ	Termination threshold for tabu search
$\boldsymbol{\lambda}$	Weights vector on the R regularizers for related domains
$\boldsymbol{\mu}$	Clustering inconsistency vector

Table 3.1: Summary of symbols and their meanings

Each relationship $(i, j) \in \mathcal{I}$ is coupled with a matrix $\mathbf{S}^{(i,j)} \in \mathbb{R}_+^{n_j \times n_i}$, indicating the (weighted) mapping between instances in $\mathcal{D}_i$ and $\mathcal{D}_j$, where n_i and n_j represent the number of instances in $\mathcal{D}_i$ and $\mathcal{D}_j$ respectively. We use $\mathbf{S}_{a,b}^{(i,j)}$ to denote the weight between the a-th instance in $\mathcal{D}_j$ and the b-th instance in $\mathcal{D}_i$, which can be either binary (0 or 1) or quantitative (any value between [0,1]). Important notations are listed in Table 3.1.

Our goal is to partition each $\mathbf{A}^{(\pi)}$ into k_π clusters while considering the co-regularizing constraints implicitly represented by the cross-domain relationships in $\mathcal{I}$.

3.3 Co-Regularized Multi-Domain Graph Clustering

In this section, we present the Co-regularized Graph Clustering (CGC) method. We model cross-domain graph clustering as a joint matrix optimization problem. The proposed CGC method simultaneously optimizes the empirical likelihood in multiple domains and take into account the cross-domain relationships.

3.3.1 Objective Function

3.3.1.1 Single-Domain Clustering

Graph clustering in a single domain has been extensively studied. We adopt the widely used non-negative matrix factorization (NMF) approach (Lee and Seung, 2000). In particular, we use the symmetric version of NMF (Kuang et al., 2012; Ding et al., 2006) to formulate the

objective of clustering on $\mathbf{A}^{(\pi)}$ as minimizing the objective function:

$$\mathcal{L}^{(\pi)} = ||\mathbf{A}^{(\pi)} - \mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}||_F^2 \tag{3.1}$$

where $||\cdot||_F$ denotes the Frobenius norm, $\mathbf{H}^{(\pi)}$ is a non-negative matrix of size $n_\pi \times k_\pi$, and k_π is the number of clusters requested. We have $\mathbf{H}^{(\pi)} = [\mathbf{h}_{1*}^{(\pi)}, \mathbf{h}_{2*}^{(\pi)}, ..., \mathbf{h}_{n_\pi *}^{(\pi)}]^{\mathrm{T}} \in \mathbb{R}_+^{n_\pi \times k_\pi}$, where each $\mathbf{h}_{a*}^{(\pi)}$ $(1 \leq a \leq n_\pi)$ represents the cluster assignment (distribution) of the a-th instance in domain $\mathcal{D}_\pi$. For hard clustering, $\mathrm{argmax}_j\, \mathbf{h}_{aj}^{(\pi)}$ is often used as the cluster assignment.

3.3.1.2 Cross-Domain Co-Regularization

To incorporate the cross-domain relationship, the key idea is to add pairwise co-regularizers to the single-domain clustering objective function. We develop two loss functions to regularize the cross-domain clustering structure. Both loss functions are designed to penalize cluster assignment inconsistency with the given cross-domain relationships. The *residual sum of squares (RSS) loss* requires that graphs in different domains are partitioned into the same number of clusters. The *clustering disagreement loss* has no such restriction.

A). Residual sum of squares (RSS) loss function

We first consider the case where the number of clusters is the same in different domains, i.e. $k_1 = k_2 = ... = k_d = k$. For simplicity, we denote the instances in domain $\mathcal{D}_\pi$ as $\{x_1^{(\pi)}, x_2^{(\pi)}, ..., x_{n_\pi}^{(\pi)}\}$. If an instance $x_a^{(i)}$ in $\mathcal{D}_i$ is mapped to an instance $x_b^{(j)}$ in $\mathcal{D}_j$, then the clustering assignments $\mathbf{h}_{a*}^{(i)}$ and $\mathbf{h}_{b*}^{(j)}$ should be similar. We now generalize the relationship to many-to-many. We use $\mathcal{N}^{(i,j)}(x_b^{(j)})$ to denote the set of indices of instances in $\mathcal{D}_i$ that are mapped to $x_b^{(j)}$ with positive weights, and $|\mathcal{N}^{(i,j)}(x_b^{(j)})|$ represents its cardinality. To penalize the inconsistency of cross-domain cluster partitions, for the l-th cluster in $\mathcal{D}_i$, the loss function (residual) for the b-th instance is

$$\mathcal{J}_{b,l}^{(i,j)} = (\mathbb{M}^{(i,j)}(x_b^{(j)}, l) - \mathbf{h}_{b,l}^{(j)})^2 \tag{3.2}$$

where

$$\mathbb{M}^{(i,j)}(x_b^{(j)}, l) = \frac{1}{|\mathcal{N}^{(i,j)}(x_b^{(j)})|} \sum_{a \in \mathcal{N}^{(i,j)}(x_b^{(j)})} \mathbf{S}_{b,a}^{(i,j)} \mathbf{h}_{a,l}^{(i)} \tag{3.3}$$

is the weighted mean of cluster assignment of instances mapped to $x_b^{(j)}$, for the l-th cluster.

We assume every non-zero row of $\mathbf{S}^{(i,j)}$ is normalized. By summing up Eq. (3.2) over all instances in $\mathcal{D}_j$ and k clusters, we have the following residual of sum of squares loss function

$$\mathcal{J}_{RSS}^{(i,j)} = \sum_{l=1}^{k}\sum_{b=1}^{n_j} \mathcal{J}_{b,l}^{(i,j)} = ||\mathbf{S}^{(i,j)}\mathbf{H}^{(i)} - \mathbf{H}^{(j)}||_F^2 \tag{3.4}$$

B). Clustering disagreement (CD) loss function

When the number of clusters in different domains varies, we can no longer use the RSS loss to quantify the inconsistency of cross-domain partitions. From the previous discussion, we observe that $\mathbf{S}^{(i,j)}\mathbf{H}^{(i)}$ in fact serves as a weighted projection of instances in domain $\mathcal{D}_i$ to instances in domain $\mathcal{D}_j$. For simplicity, we denote the matrix $\widetilde{\mathbf{H}}^{(i\to j)} = \mathbf{S}^{(i,j)}\mathbf{H}^{(i)}$. Recall that $\mathbf{h}_{a*}^{(j)}$ represents a cluster assignment over k_j clusters for the a-th instance in $\mathcal{D}_j$. Then $\widetilde{\mathbf{H}}_{a*}^{(i\to j)}$ corresponds to $\mathbf{H}_{a*}^{(j)}$ for the a-th instance in domain $\mathcal{D}_j$. The previous RSS loss compares them directly to measure the clustering inconsistency. However, it is inapplicable to the case where different domains have different numbers of clusters. To tackle this problem, we first measure the similarity between $\widetilde{\mathbf{H}}_{a*}^{(i\to j)}$ and $\widetilde{\mathbf{H}}_{b*}^{(i\to j)}$, and the similarity between $\mathbf{H}_{a*}^{(j)}$ and $\mathbf{H}_{b*}^{(j)}$. Then we measure the difference between these two similarity values. Taking Figure 3.1 (b) as an example. Note that Ⓐ and Ⓑ in domain 1 are mapped to ② in domain 2, and Ⓒ is mapped to ④. Intuitively, if the similarity between clustering assignments for ② and ④ is small, the similarity of clustering assignments between Ⓐ and Ⓒ and the similarity between Ⓑ and Ⓒ should also be small. Note that symmetric NMF can handle both linearity and nonlinearity (Kuang et al., 2012). Thus in this chapter, we choose a linear kernel to measure the in-domain cluster assignment similarity, i.e., $K(\mathbf{h}_{a*}^{(j)}, \mathbf{h}_{b*}^{(j)}) = \mathbf{h}_{a*}^{(j)}(\mathbf{h}_{b*}^{(j)})^{\mathrm{T}}$. The cross-domain clustering disagreement loss function is thus defined as

$$\begin{aligned}\mathcal{J}_{CD}^{(i,j)} &= \sum_{a=1}^{n_j}\sum_{b=1}^{n_j}\Big(K(\widetilde{\mathbf{H}}_{a*}^{(i\to j)}, \widetilde{\mathbf{H}}_{b*}^{(i\to j)}) - K(\mathbf{h}_{a*}^{(j)}, \mathbf{h}_{b*}^{(j)})\Big)^2 \\ &= ||\mathbf{S}^{(i,j)}\mathbf{H}^{(i)}(\mathbf{S}^{(i,j)}\mathbf{H}^{(i)})^{\mathrm{T}} - \mathbf{H}^{(j)}(\mathbf{H}^{(j)})^{\mathrm{T}}||_F^2\end{aligned} \tag{3.5}$$

3.3.1.3 Joint Matrix Optimization

We can integrate the domain-specific objective and the loss function quantifying the inconsistency of cross-domain partitions into a unified objective function

$$\min_{\mathbf{H}^{(\pi)}\geq 0(1\leq\pi\leq d)} \mathcal{O} = \sum_{i=1}^{d} \mathcal{L}^{(i)} + \sum_{(i,j)\in\mathcal{I}} \lambda^{(i,j)} \mathcal{J}^{(i,j)} \tag{3.6}$$

where $\mathcal{J}^{(i,j)}$ can be either $\mathcal{J}_{RSS}^{(i,j)}$ or $\mathcal{J}_{CD}^{(i,j)}$. $\lambda^{(i,j)} \geq 0$ is a tuning parameter balancing between in-domain clustering objective and cross-domain regularizer. When all $\lambda^{(i,j)}$= 0, Eq. (3.6) degenerates to d independent graph clusterings. Intuitively, the more reliable the prior cross-domain relationship, the larger the value of $\lambda^{(i,j)}$.

3.3.2 Learning Algorithm

In this section, we present an alternating scheme to optimize the objective function in Eq. (3.6); that is, we optimize the objective with respect to one variable while fixing others. This procedure continues until convergence. The objective function is invariant under these updates if and only if $\mathbf{H}^{(\pi)}$'s are at a stationary point (Lee and Seung, 2000). Specifically, the solution to the optimization problem in Eq. (3.6) is based on the following two theorems, which are derived from the Karush-Kuhn-Tucker (KKT) complementarity condition (Boyd and Vandenberghe, 2004). Detailed theoretical analysis of the optimization procedure will be presented in the next section.

Theorem 2. For RSS loss, updating $\mathbf{H}^{(\pi)}$ according to Eq. (3.7) will monotonically decrease the objective function in Eq. (3.6) until convergence.

$$\mathbf{H}^{(\pi)} \leftarrow \mathbf{H}^{(\pi)} \circ \left(\frac{\boldsymbol{\Psi}'(\mathbf{H}^{(\pi)})}{\boldsymbol{\Xi}'(\mathbf{H}^{(\pi)})} \right)^{\frac{1}{4}} \tag{3.7}$$

where

$$\begin{aligned} \boldsymbol{\Psi}'(\mathbf{H}^{(\pi)}) = \mathbf{A}^{(\pi)}\mathbf{H}^{(\pi)} &+ \sum_{(i,\pi)\in\mathcal{I}} \frac{\lambda^{(i,\pi)}}{2} \mathbf{S}^{(i,\pi)}\mathbf{H}^{(i)} \\ &+ \sum_{(\pi,j)\in\mathcal{I}} \frac{\lambda^{(\pi,j)}}{2} (\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{H}^{(j)} \end{aligned} \tag{3.8}$$

and

$$\begin{aligned} \boldsymbol{\Xi}'(\mathbf{H}^{(\pi)}) = \mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} &+ \sum_{(i,\pi)\in\mathcal{I}} \frac{\lambda^{(i,\pi)}}{2} \mathbf{H}^{(\pi)} \\ &+ \sum_{(\pi,j)\in\mathcal{I}} \frac{\lambda^{(\pi,j)}}{2} (\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)} \end{aligned} \tag{3.9}$$

Theorem 3. For CD loss, updating $\mathbf{H}^{(\pi)}$ according to Eq. (3.10) will monotonically decrease the objective function in Eq. (3.6) until convergence.

$$\mathbf{H}^{(\pi)} \leftarrow \mathbf{H}^{(\pi)} \circ \left(\frac{\mathbf{\Psi}(\mathbf{H}^{(\pi)})}{\mathbf{\Xi}(\mathbf{H}^{(\pi)})} \right)^{\frac{1}{4}} \tag{3.10}$$

where

$$\begin{aligned} \mathbf{\Psi}(\mathbf{H}^{(\pi)}) = & \mathbf{A}^{(\pi)}\mathbf{H}^{(\pi)} \\ & + \sum_{(i,\pi)\in\mathcal{I}} \lambda^{(i,\pi)}\mathbf{S}^{(i,\pi)}\mathbf{H}^{(i)}(\mathbf{H}^{(i)})^{\mathrm{T}}(\mathbf{S}^{(i,\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} \\ & + \sum_{(\pi,j)\in\mathcal{I}} \lambda^{(\pi,j)}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{H}^{(j)}(\mathbf{H}^{(j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)} \end{aligned} \tag{3.11}$$

and

$$\begin{aligned} \mathbf{\Xi}(\mathbf{H}^{(\pi)}) = & \mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} \\ & + \sum_{(i,\pi)\in\mathcal{I}} \lambda^{(i,\pi)}\mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} \\ & + \sum_{(\pi,j)\in\mathcal{I}} \lambda^{(\pi,j)}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)} \end{aligned} \tag{3.12}$$

where $\circ$, $\frac{[\cdot]}{[\cdot]}$ and $(\cdot)^{\frac{1}{4}}$ are element-wise operators.

Based on Theorem 2 and Theorem 3, we develop the iterative multiplicative updating algorithm for optimization and summarize it in Algorithm 1.

3.3.3 Theoretical Analysis

3.3.3.1 Derivation

We derive the solution to Eq. (3.6) following the constrained optimization theory (Boyd and Vandenberghe, 2004). Since the objective function is not jointly convex, we adopt an effective alternating minimization algorithm to find a locally optimal solution. We prove Theorem 3 in the following. The proof of Theorem 2 is similar and hence omitted.

We formulate the Lagrange function for optimization

$$\begin{aligned} & L(\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, ..., \mathbf{H}^{(d)}) \\ & = \sum_{i=1}^{d} ||\mathbf{A}^{(i)} - \mathbf{H}^{(i)}(\mathbf{H}^{(i)})^{\mathrm{T}}||_F^2 \\ & + \sum_{(i,j)\in\mathcal{I}} \lambda^{(i,j)} ||\mathbf{S}^{(i,j)}\mathbf{H}^{(i)}(\mathbf{S}^{(i,j)}\mathbf{H}^{(i)})^{\mathrm{T}} - \mathbf{H}^{(j)}(\mathbf{H}^{(j)})^{\mathrm{T}}||_F^2 \end{aligned} \tag{3.13}$$

Algorithm 1: Co-regularized Graph Clustering (CGC)

Input: graphs from d domains, each of which is represented by an affinity matrix $\mathbf{A}^{(\pi)}$, k_π(number of clusters in domain $\mathcal{D}_\pi$), a set of pairwise relationships $\mathcal{I}$ and the corresponding matrices $\{\mathbf{S}^{(i,j)}\}$, parameters $\{\lambda^{(i,j)}\}$

Output: clustering results for each domain (inferred from $\mathbf{H}^{(\pi)}$)

```
begin
    Normalize all graph affinity matrices by Frobenius norm;
    foreach (i, j) ∈ I do
        Normalize non-zero rows of S^(i,j);
    end
    for π ← 1 to d do
        Initialize H^(π) with random values between (0,1];
    end
    repeat
        for π ← 1 to d do
            Update H^(π) by Eq. (3.7) or (3.10);
        end
    until convergence;
end
```

Without loss of generality, we only show the derivation of the updating rule for one domain π ($\pi \in [1, d]$). The partial derivative of Lagrange function with respect to $\mathbf{H}^{(\pi)}$ is:

$$\begin{aligned}
&\nabla_{\mathbf{H}^{(\pi)}} L = \\
&- \mathbf{A}^{(\pi)}\mathbf{H}^{(\pi)} + \mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} \\
&+ \sum_{(\pi,j)\in\mathcal{I}} \lambda^{(\pi,j)}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)} \\
&- \sum_{(\pi,j)\in\mathcal{I}} \lambda^{(\pi,j)}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{H}^{(j)}(\mathbf{H}^{(j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)} \\
&- \sum_{(i,\pi)\in\mathcal{I}} \lambda^{(i,\pi)}\mathbf{S}^{(i,\pi)}\mathbf{H}^{(i)}(\mathbf{H}^{(i)})^{\mathrm{T}}(\mathbf{S}^{(i,\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} \\
&+ \sum_{(i,\pi)\in\mathcal{I}} \lambda^{(i,\pi)}\mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)}
\end{aligned} \tag{3.14}$$

Using the Karush-Kuhn-Tucker (KKT) complementarity condition (Boyd and Vandenberghe, 2004) for the non-negative constraint on $\mathbf{H}^{(\pi)}$, we have

$$\nabla_{\mathbf{H}^{(\pi)}} L \circ \mathbf{H}^{(\pi)} = \mathbf{0} \tag{3.15}$$

The above formula leads to the updating rule for $\mathbf{H}^{(\pi)}$ in Eq. (3.10).

3.3.3.2 Convergence

We use the auxiliary function approach (Lee and Seung, 2000) to prove the convergence of Eq. (3.10) in Theorem 3. We first introduce the definition of auxiliary function as follows:

Definition 3.3.1. $Z(h, \tilde{h})$ is an auxiliary function for $L(h)$ if the conditions

$$Z(h, \tilde{h}) \geq L(h) \qquad and \qquad Z(h, h) = L(h), \tag{3.16}$$

are satisfied for any given $h, \tilde{h}$ (Lee and Seung, 2000).

Lemma 2. If Z is an auxiliary function for L, then L is non-increasing under the update (Lee and Seung, 2000).

$$h^{(t+1)} = \arg\min_{h} Z(h, h^{(t)}) \tag{3.17}$$

Theorem 4. Let $L(\mathbf{H}^{(\pi)})$ denote the sum of all terms in L containing $\mathbf{H}^{(\pi)}$. The following function

$$\begin{aligned}
Z(\mathbf{H}^{(\pi)}, \tilde{\mathbf{H}}^{(\pi)}) &= -2 \sum_{klm} \mathbf{A}^{(\pi)}_{ml} P(k,l,m) \\
&+ (1 + \sum_{(i,\pi)\in\mathcal{I}} \lambda^{(i,\pi)}) \sum_{kl} \left(\tilde{\mathbf{H}}^{(\pi)} (\tilde{\mathbf{H}}^{(\pi)})^{\mathrm{T}} \tilde{\mathbf{H}}^{(\pi)} \right)_{kl} \cdot \frac{(\mathbf{H}^{(\pi)}_{kl})^4}{(\tilde{\mathbf{H}}^{(\pi)}_{kl})^3} \\
&- 2 \sum_{(i,\pi)\in\mathcal{I}} \lambda^{(i,\pi)} \sum_{klm} \left(\mathbf{S}^{(i,\pi)} \mathbf{H}^{(i)} (\mathbf{H}^{(i)})^{\mathrm{T}} (\mathbf{S}^{(i,\pi)})^{\mathrm{T}} \right)_{lm} P(k,l,m) \\
&+ \sum_{(\pi,j)\in\mathcal{I}} \lambda^{(\pi,j)} \sum_{kl} (\mathbf{Q}(j))_{kl} \cdot \frac{(\mathbf{H}^{(\pi)}_{lk})^4}{(\tilde{\mathbf{H}}^{(\pi)}_{lk})^3} \\
&- 2 \sum_{(\pi,j)\in\mathcal{I}} \lambda^{(\pi,j)} \sum_{klm} \left((\mathbf{S}^{(\pi,j)})^{\mathrm{T}} \mathbf{H}^{(j)} (\mathbf{H}^{(j)})^{\mathrm{T}} \mathbf{S}^{(\pi,j)} \right)_{lm} P(k,l,m)
\end{aligned} \tag{3.18}$$

is an auxiliary function for $L(\mathbf{H}^{(\pi)})$, where $P(k,l,m) = \tilde{\mathbf{H}}^{(\pi)}_{lk} \tilde{\mathbf{H}}^{(\pi)}_{mk} \left(1 + \log \frac{\mathbf{H}^{(\pi)}_{lk} \mathbf{H}^{(\pi)}_{mk}}{\tilde{\mathbf{H}}^{(\pi)}_{lk} \tilde{\mathbf{H}}^{(\pi)}_{mk}} \right)$ and $\mathbf{Q}(j) = (\tilde{\mathbf{H}}^{(\pi)})^{\mathrm{T}} (\mathbf{S}^{(\pi,j)})^{\mathrm{T}} \mathbf{S}^{(\pi,j)} \tilde{\mathbf{H}}^{(\pi)} (\tilde{\mathbf{H}}^{(\pi)})^{\mathrm{T}} (\mathbf{S}^{(\pi,j)})^{\mathrm{T}} \mathbf{S}^{(\pi,j)}$. Furthermore, it is a convex function in $\mathbf{H}^{(\pi)}$ and has a global minimum.

Theorem 4.3 can be proved using a similar idea to that in (Ding et al., 2006) by validating $Z(\mathbf{H}^{(\pi)}, \tilde{\mathbf{H}}^{(\pi)}) \geq L(\mathbf{H}^{(\pi)})$, $Z(\mathbf{H}^{(\pi)}, \mathbf{H}^{(\pi)}) = L(\mathbf{H}^{(\pi)})$, and the Hessian matrix $\nabla\nabla_{\mathbf{H}^{(\pi)}} Z(\mathbf{H}^{(\pi)}, \tilde{\mathbf{H}}^{(\pi)}) \succeq \mathbf{0}$. Due to space limitation, we omit the details.

Based on Theorem 4.3, we can minimize $Z(\mathbf{H}^{(\pi)}, \tilde{\mathbf{H}}^{(\pi)})$ with respect to $\mathbf{H}^{(\pi)}$ with $\tilde{\mathbf{H}}^{(\pi)}$ fixed. We set $\nabla_{\mathbf{H}^{(\pi)}} Z(\mathbf{H}^{(\pi)}, \tilde{\mathbf{H}}^{(\pi)}) = \mathbf{0}$, and get the following updating formula

$$\mathbf{H}^{(\pi)} \leftarrow \tilde{\mathbf{H}}^{(\pi)} \circ \left(\frac{\mathbf{\Psi}(\tilde{\mathbf{H}}^{(\pi)})}{\mathbf{\Xi}(\tilde{\mathbf{H}}^{(\pi)})} \right)^{\frac{1}{4}},$$

which is consistent with the updating formula derived from the aforementioned KKT condition.

From Lemma 4.2 and Theorem 4.3, for each subsequent iteration of updating $\mathbf{H}^{(\pi)}$, we have $L((\mathbf{H}^{(\pi)})^0) = Z((\mathbf{H}^{(\pi)})^0, (\mathbf{H}^{(\pi)})^0) \geq Z((\mathbf{H}^{(\pi)})^1, (\mathbf{H}^{(\pi)})^0) \geq Z((\mathbf{H}^{(\pi)})^1, (\mathbf{H}^{(\pi)})^1) = L((\mathbf{H}^{(\pi)})^1) \geq ... \geq L((\mathbf{H}^{(\pi)})^{Iter})$. Thus $L(\mathbf{H}^{(\pi)})$ monotonically decreases. This is also true for the other variables. Since the objective function Eq. (3.6) is lower bounded by 0, the correctness of Theorem 3 is proved. Theorem 2 can be proven with a similar strategy.

3.3.3.3 Complexity Analysis

The time complexity of Algorithm 1 (for both loss functions) is $\mathcal{O}(Iter \cdot d|\mathcal{I}|(\tilde{n}^3 + \tilde{n}^2\tilde{k}))$, where $\tilde{n}$ is the largest n_π ($1 \leq \pi \leq d$), $\tilde{k}$ is the largest k_π and $Iter$ is the number of iterations needed before convergence. In practice, $|\mathcal{I}|$ and d are usually small constants. Moreover, from Eq. (3.10) and Eq. (3.7), we observe that the $\tilde{n}^3$ term is from the matrix multiplication $(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}$. Since $\mathbf{S}^{(\pi,j)}$ is the input matrix and often very sparse, we can compute $(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}$ in advance in sparse form. In this way, the complexity of Algorithm 1 is reduced to $\mathcal{O}(Iter \cdot \tilde{n}^2\tilde{k})$.

3.3.4 Finding Global Optimum

The objective function Eq. (3.6) is a fourth-order non-convex function with respect to $\mathbf{H}^{(\pi)}$. The achieved stationary points (satisfying KKT condition in Eq. (3.15)) may not be the global optimum. Many methods have been proposed in the literature to avoid local optima, such as Tabu search (Glover and McMillan, 1986), particle swarm optimization (PSO) (Dorigo et al., 2008), and estimation of distribution algorithm (EDA) (Larraanaga and Lozano, 2001). Since our objective function is continuously differentiable over the entire parameter space, we develop a learning algorithm for global optimization by population-based Tabu Search. We further develop a parallelized version of the learning algorithm.

3.3.4.1 Tabu Search Based Algorithm for Finding Global Optimum

In Algorithm 1, we find a local optima for $\mathbf{H}^{(\pi)}(0 \leq \pi \leq d)$ from the starting point initialized in lines 6 to 8. Here, we treat all $\mathbf{H}^{(\pi)}$'s as one point $\mathbf{H}$ (for example, converting them into one vector). Then, the iterations for finding global optimum are summarized below.

1. Given the probability ϕ that a random point converges to the global minimum and a confidence level α, set termination threshold c_ϕ according to equation (3.23). Initialize counter $c := 0$, and randomly chose one initial point; then use Algorithm 1 to find the corresponding local optima.
2. Mark this local optima point as a *Tabu* point T_c, and keep track of the "global optimum" found so far in H^*, set counter $c := c + 1$.
3. If $c \geq c_\phi$, return;
4. Randomly choose another point far from the *Tabu* points, and use Algorithm 1 to find the corresponding local optima, go to Step 2.

In the above steps, we try to avoid converging to any known local minimums by applying the dropping and re-selecting scheme. The nearer a point lies to a *Tabu* point, the less likely it is to get selected as a new initial state. As more iterations are taken, the risk that all iterations converge to local optima drops substantially. Our method not only keeps track of local information (KKT points), but also does global search so that the probability of finding the optimal minima significantly increases. Such Markov chain process ensures that the algorithm converges to the global minimum with probability 1 when c_ϕ is large enough.

3.3.4.2 Lower Bound of Termination Threshold c_ϕ

To find the global optimum with confidence at least α, the probability of all searched c_ϕ points in local minimum should be less than $1 - \alpha$, i.e.,

$$\prod_{i=1}^{c_\phi} p(\textit{point } i \textit{ converge to local minima}) \leq 1 - \alpha. \tag{3.19}$$

ϕ	0.5	0.1	0.01	0.001	0.5	0.1	0.01	0.001	0.0001
α	0.99	0.99	0.99	0.99	0.999	0.999	0.999	0.999	0.999
c_ϕ	4	9	30	96	4	11	37	118	372

Table 3.2: Population size and termination threshold for the Tabu search algorithm

Given ϕ, the probability of a random point that converges to global minimum, we know that the first point has probability $1-\phi$ to converge to a *local* one. If the system lacks memory and never keeps records of existing points, all points would have the same converging probability to the global minimum. However, we mark each local optima point as a *Tabu* point, and try to locate further chosen ones far from existing local minima. Such operation decreases the probability of getting into the same local minimum. It results in an increasing of the global converging probability by a factor of $1-\phi$ in each step, i.e.,
$p(point\ i\ converges\ to\ local\ minima) = (1-\phi)p(point\ i-1\ converges\ to\ local\ minima)$.
Substituting this and $p(first\ point\ converges\ to\ local\ minima) = 1-\phi$ into equation (3.19), we have

$$\prod_{i=1}^{c_\phi}(1-\phi)^i \leq 1-\alpha. \tag{3.20}$$

Thus we have

$$c_\phi \geq \sqrt{2\log_{1-\phi}(1-\alpha)+\frac{1}{4}}-\frac{1}{2}. \tag{3.21}$$

Table 3.2 shows the value of c_ϕ for some typical choices of ϕ and α. We can see that the proposed CGC algorithm converges to the global optimum with a small number of steps.

3.3.4.3 Parallelizing the Global Optimum Search Process

Assume that we have N processors (which may not be identical) that can run in parallel. A simple version is to randomly choose N ($N>1$) points in each step (that are all far from *Tabu* points), and to find N local optima in parallel using Algorithm 1 (i.e., population size = N). The termination threshold can be derived in a similar way. Initially, the probability of all N nodes converging to local minima is $(1-\phi)^N$, and such probability is decreasing by a factor of $(1-\phi)^N$ for each step. Thus, the termination threshold c_ϕ should agree with the following equation:

$$\prod_{i=1}^{c_\phi}(1-\phi)^{iN} \leq 1-\alpha. \tag{3.22}$$

This results in the following expression of the threshold:

$$c_\phi \geq \sqrt{\frac{2}{N}\log_{1-\phi}(1-\alpha)+\frac{1}{4}}-\frac{1}{2}. \tag{3.23}$$

This algorithm can speed up by a factor of $\sqrt{N}$ (with N being the number of processors).

3.3.5 Re-Evaluating Cross-Domain Relationship

In real applications, the cross-domain instance relationship based on prior knowledge may contain noise. Thus, it is crucial to allow users to evaluate whether the provided relationships violate any single-domain clustering structures. In this section, we develop a principled way to achieve this goal. In fact, we only need to slightly modify the co-regularization loss functions in Section 3.3.1.2 by multiplying a confidence matrix $\mathbf{W}^{(i,j)}$ to each $\mathbf{S}^{(i,j)}$. Each element in the confidence matrix $\mathbf{W}^{(i,j)}$ is initialized to 1. For RSS loss, we give the modified loss function below (the case for CD loss is similar).

$$\mathcal{J}_W^{(i,j)} = ||(\mathbf{W}^{(i,j)} \circ \mathbf{S}^{(i,j)})\mathbf{H}^{(i)} - \mathbf{H}^{(j)}||_F^2 \tag{3.24}$$

Here, $\circ$ is element-wise product. By optimizing the following objective function, we can learn the optimal confidence matrix

$$\min_{\mathbf{W}\geq 0, \mathbf{H}^{(\pi)}\geq 0 (1\leq\pi\leq d)} \mathcal{O} = \sum_{i=1}^{d}\mathcal{L}^{(i)} + \sum_{(i,j)\in\mathcal{I}}\lambda^{(i,j)}\mathcal{J}_W^{(i,j)} \tag{3.25}$$

Eq. (3.25) can be optimized by iteratively implementing following two steps until convergence: 1) replace $\mathbf{S}^{(\pi,j)}$ and $\mathbf{S}^{(i,\pi)}$ in Eq. (3.7) with $(\mathbf{W}^{(\pi,j)} \circ \mathbf{S}^{(\pi,j)})$ and $(\mathbf{W}^{(i,\pi)} \circ \mathbf{S}^{(i,\pi)})$ respectively, and use the replaced formula to update each $\mathbf{H}^{(\pi)}$; 2) use the following formula to update each $\mathbf{W}^{(i,j)}$

$$\mathbf{W}^{(i,j)} \leftarrow \mathbf{W}^{(i,j)} \circ \sqrt{\frac{(\mathbf{H}^{(j)}(\mathbf{H}^{(i)})^{\mathrm{T}}) \circ \mathbf{S}^{(i,j)}}{((\mathbf{W}^{(i,j)} \circ \mathbf{S}^{(i,j)})\mathbf{H}^{(i)}(\mathbf{H}^{(i)})^{\mathrm{T}}) \circ \mathbf{S}^{(i,j)}}} \tag{3.26}$$

Here, $\sqrt{\cdot}$ is element-wise square root. Note that many elements in $\mathbf{S}^{(i,j)}$ are 0. We only update the elements in $\mathbf{W}^{(i,j)}$ whose corresponding elements in $\mathbf{S}^{(i,j)}$ are positive. In the following, we only focus on such elements. The learned confidence matrix minimizes the

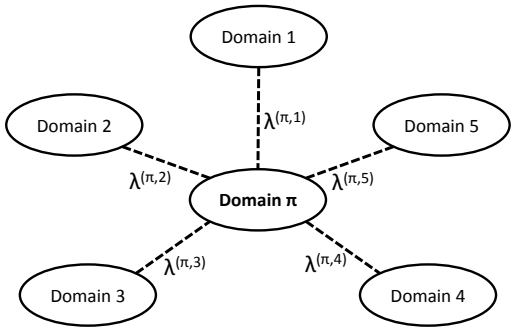


Figure 3.2: Focused domain π and 5 domains related to it

inconsistency between the original single-domain clustering structure and the prior cross-domain relationship. Thus for any element $\mathbf{W}_{a,b}^{(i,j)}$, the smaller the value, the stronger the inconsistency between $\mathbf{S}_{a,b}^{(i,j)}$ and single-domain clustering structures in $\mathcal{D}_i$ and $\mathcal{D}_j$. Therefore, we can sort the values of $\mathbf{W}^{(i,j)}$ and report to users the smallest elements and their corresponding cross-domain relationships. Accurate relationship can help to improve the overall results. On the other hand, an inaccurate relationship may provide wrong guidance of the clustering process. Our method allows the users to examine these critical relationships and improve the accuracy of the results.

3.3.6 Assigning Optimal Weights Associated with Focused Domain

In Section 3.3.1.3, we use parameter $\lambda^{(i,j)} \geq 0$ to balance between in-domain clustering objective and cross-domain regularizer. Typically, the parameter is given based on the prior knowledge of the cross-domain relationship. Therefore, the more reliable the prior cross-domain relationship, the larger the value of $\lambda^{(i,j)}$. In real applications, such prior knowledge may not be available. In this case, we need an effective approach to automatically balance different cross-domain regularizers. This problem, however, is hard to solve due to the arbitrary topologies of relationships among domains. To make it feasible, we simplify the problem to the case where the user focuses on the clustering accuracy of only one domain at a time.

As illustrated in Figure 3.2, domain π is the focused domain. There are 5 other domains related to it. These related domains serve as side information. As such, we can do a single domain clustering for all related domains to obtain each $\mathbf{H^{(i)}}, (1 \leq i \leq 5)$, then use these auxiliary domains to improve the accuracy of graph partition for domain π. We make a reasonable

assumption that the associated weights add up to 1, i.e., $\sum_{j=1}^{5} \lambda^{(\pi,j)} = 1$. Formally, if domain π is the focused domain, then the following objective function can be used to automatically assign optimal weights

$$\begin{aligned} \min_{\mathbf{H}^{(\pi)},\boldsymbol{\lambda}} \mathcal{O} = \mathcal{L}^{(\pi)} + & \sum_{\substack{(\pi,k_j)\in\mathcal{I} \\ 1\le j\le R}} \lambda^{(\pi,t_j)}\mathcal{J}^{(\pi,t_j)} + \gamma||\boldsymbol{\lambda}||_2^2 \\ & \text{s.t.} \quad \mathbf{H}^{(\pi)} \ge 0, \boldsymbol{\lambda} \ge 0, \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{1} = 1 \end{aligned} \tag{3.27}$$

where $\boldsymbol{\lambda} = [\lambda^{(\pi,t_1)}, \lambda^{(\pi,t_2)}, ..., \lambda^{(\pi,t_R)}]^{\mathrm{T}}$ are the weights on the R regularizers for related domains, $\mathbf{1} \in \mathbb{R}^{R\times 1}$ is a vector of all ones, $\gamma > 0$ is used to control the complexity of $\boldsymbol{\lambda}$. By adding the ℓ_2-norm, Eq. 3.27 avoids the trivial solution. Eq. 3.27 can selectively integrate auxiliary domains and assign smaller weights to noisy domains. This will be beneficial to the graph partition performance of the focused domain π.

Eq. 3.27 can be solved using an alternating scheme similar as Algorithm 1, in which $\mathbf{H}^{(\pi)}$ and $\boldsymbol{\lambda}$ are iteratively considered as constants. Specifically, in the first step, we fix $\boldsymbol{\lambda}$ and update $\mathbf{H}^{(\pi)}$ using similar strategy as in Algorithm 1, then we fix $\mathbf{H}^{(\pi)}$ and optimize $\boldsymbol{\lambda}$. For simplicity, we denote $\boldsymbol{\mu} = [\mu_{t_1}, \mu_{t_2}, ..., \mu_{t_R}]^{\mathrm{T}}$, where $\mu_r = \mathcal{J}^{(\pi,t_r)}$. Since we fix $\mathbf{H}^{(\pi)}$ at this step, the first term in Eq. 3.27 is a constant and can be ignored, then we can rewrite 3.27 as follows:

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \tilde{\mathcal{O}} &= \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\mu} + \gamma\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\lambda} \\ \text{s.t.} \quad & \boldsymbol{\lambda} \ge 0, \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{1} = 1. \end{aligned} \tag{3.28}$$

Eq. 3.28 is a quadratic optimization problem with respect to $\boldsymbol{\lambda}$, and can be formulated as a minimization problem

$$\hat{\mathcal{O}}(\boldsymbol{\lambda}, \boldsymbol{\beta}, \theta) = \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\mu} + \gamma\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\lambda} - \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\beta} - \theta(\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{1} - 1) \tag{3.29}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_R]^{\mathrm{T}} \ge 0$ and $\theta \ge 0$ are the Karush-Kuhn-Tucker (KKT) multipliers (Boyd and Vandenberghe, 2004). The optimal $\boldsymbol{\lambda}^*$ should satisfy the following four conditions:

1. *Stationary condition*: $\nabla_{\boldsymbol{\lambda}^*}\hat{\mathcal{O}}(\boldsymbol{\lambda}^*, \boldsymbol{\beta}, \theta) = \boldsymbol{\mu} + 2\gamma\boldsymbol{\lambda}^* - \boldsymbol{\beta} - \theta\mathbf{1} = \mathbf{0}$
2. *Feasible condition*: $\lambda_r^* \ge \mathbf{0}, \sum_{r=1}^{R} \lambda_r^* - 1 = 0$
3. *Dual feasibility*: $\beta_r \ge 0, 1 \le r \le R$
4. *Complementary slackness*: $\beta_r\lambda_r^* = 0, 1 \le r \le R$

From the stationary condition, λ_r can be computed as

$$\lambda_r = \frac{\beta_r + \theta - \mu_r}{2\gamma} \tag{3.30}$$

We observed that λ_r depends on the specification of β_r and γ, similar as in (Yu et al., 2013), we can divide the problem into three cases:

1. When $\theta - \mu_r > 0$, since $\beta_r \geq 0$, we get $\lambda_r > 0$. From the complementary slackness, we know that $\beta_r \lambda_r = 0$, then we have $\beta_r = 0$, and therefore, $\lambda_r = \frac{\theta - \lambda_r}{2\gamma}$.
2. When $\theta - \mu_r < 0$, since $\lambda_r \geq 0$, then we have $\beta_r > 0$. Since $\beta_r \lambda_r = 0$, we have $\lambda_r = 0$.
3. When $\theta - \mu_r = 0$, since $\beta_r \lambda_r = 0$ and $\lambda_r = \frac{\beta_r}{2\gamma}$, then we have $\beta_r = 0$ and $\lambda_r = 0$.

Therefore, if we sort μ_r by ascending order, $\mu_1 \leq \mu_2 \leq ... \leq \mu_R$, then there exists $\tilde{\theta} > 0$ such that $\tilde{\theta} - \mu_p > 0$ and $\tilde{\theta} - \mu_{p+1} \leq 0$. Then λ_r can be calculated with following formula:

$$\lambda_r = \begin{cases} \frac{\theta - \mu_r}{2\gamma}, & if \quad r \leq p \\ 0. & else \end{cases} \tag{3.31}$$

Eq. 3.31 implies the intuition of the optimal weights assignment. That is when μ_r is large, which means the clustering inconsistency is high between domain π and t_r. The inconsistency may come from either the noisy data in domain k_r or noise in cross-domain relationship matrix $\mathbf{S}^{(\pi, t_r)}$. At this time, Eq. 3.31 will assign a small weight λ_r so that the model is less likely to suffer from those noisy domains and instead get the most applicable clustering result.

Considering that $\sum_{r=1}^{p} = 1$, we can calculate θ as follows

$$\theta = \frac{2\gamma + \sum_{r=1}^{p} \mu_r}{p} \tag{3.32}$$

Thus, we can search the value of p from R to 1 decreasingly (Yu et al., 2013). Once $\theta - \mu_p > 0$, then we find the value of p. After we obtain the value of p, we can assign values for each $\lambda_r (1 \leq r \leq R)$ according to Eq. 3.31. We observe that when γ is very large, θ will be large, and

Algorithm 2: Assigning Optimal Weights Associated with Focused Domain π

Input: graphs from R domains that are associated with the focused domain π, each of which is represented by an affinity matrix $\mathbf{A}^{(t_r)}, (1 \leq r \leq R)$, k_r(number of clusters in domain $\mathcal{D}_r$), a set of pairwise relationships $\mathcal{I}$ and the corresponding matrices $\{\mathbf{S}^{(\pi,k_r)}\}$, γ.

Output: clustering result for domain π(inferred from $\mathbf{H}^{(\pi)}$), optimal weights λ_r, $(1 \leq r \leq R)$.

begin
  Do single domain clustering for all associated domains t_r to get $\mathbf{H}^{(t_r)}$, $(1 \leq r \leq R)$;
  for $r \leftarrow 1$ **to** R **do**
    $\lambda_r \leftarrow 1/R$;
  end
  repeat
    Use Algorithm 1 to infer $\mathbf{H}^{(\pi)}$;
    for $r \leftarrow 1$ **to** R **do**
      $\mu_r \leftarrow \mathcal{J}^{(\pi,t_r)}$;
    end
    Sort $\mu_r (1 \leq r \leq R)$ in increasing order;
    $p \leftarrow R + 1$;
    do
      $p \leftarrow p - 1$;
      $\theta \leftarrow \frac{2\gamma + \sum_{r=1}^{p} \mu_r}{p}$;
    while $\theta - \mu_p \leq 0$;
    for $r \leftarrow 1$ **to** p **do**
      $\lambda_r \leftarrow \frac{\theta - \mu_r}{2\gamma}$;
    end
    for $r \leftarrow p + 1$ **to** R **do**
      $\lambda_r \leftarrow 0$;
    end
  until *convergence*;
end

all domains will be selected, i.e., each λ_r will be a small but non-zero value. In contrast, when γ is very small, at least one domain (domain t_1) will be selected, and other λ_r's ($r \neq 1$) will be 0. Hence, we can use γ to control how many auxiliary domains will be integrated for graph partition for domain π. Specifically, the detailed algorithm for assigning optimal weights associated with focused domain π is shown in Algorithm 2.

Identifier	#Instances	#Attributes
Iris	100	4
Wine	119	13
Ionosphere	351	34
WDBC	569	30

Table 3.3: The UCI benchmarks

Algorithm 2 alternatively optimizes $\mathbf{H}^{\pi}$ (line 7) and $\boldsymbol{\lambda}$ (line 8–22). Since both steps decrease the value of the objective function (3.27) and the objective function is lower bounded by 0, the convergence of the algorithm is guaranteed.

3.4 Experimental Results

In this section, we present extensive experimental results on evaluating the performance of our method.

3.4.1 Effectiveness Evaluation

We evaluate the proposed method by clustering benchmark data sets from the UCI Archive (Asuncion and Newman., 2007). We use four data sets with class label information, namely Iris, Wine, Ionosphere and Breast Cancer Wisconsin (Diagnostic) data sets. They are from four different domains. To make each data set contain the same number of clusters, we follow the preprocessing step in (Wang and Davidson, 2010) to remove the SETOSA class from the Iris data set and Class 1 from the Wine data set. The statistics of the resulting data sets are shown in Table 3.3.

For each data set, we compute the affinity matrix using the RBF kernel (Boyd and Vandenberghe, 2004). Our goal is to examine whether cross-domain relationships can help to enhance the accuracy of the clustering results. We construct two cross-domain relationships: Wine-Iris and Ionosphere-WDBC. The relationships are generated based on the class labels, i.e., positive (negative) instances in one domain can only be mapped to positive (negative) instances in another domain. We employ the widely used Clustering Accuracy (Xu et al., 2003) to measure the quality of the clustering results. Parameter λ is set to 1 throughout the experiments. Since no existing method can handle the multi-domain

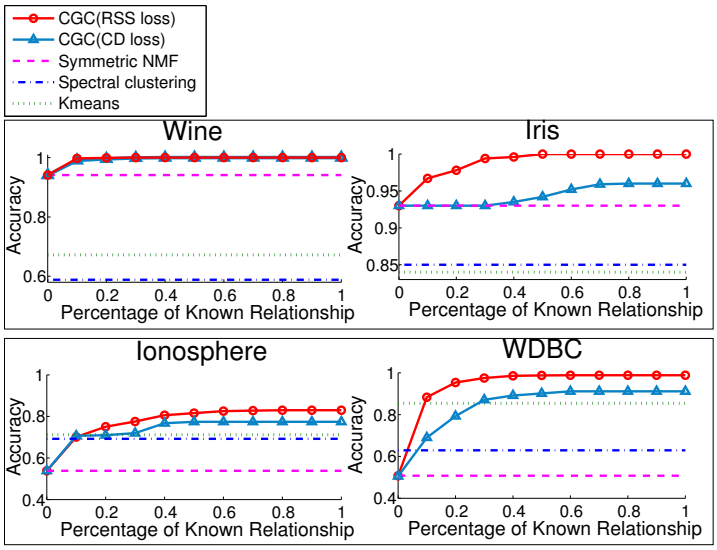


Figure 3.3: Clustering results on UCI datasets(Wine v.s. Iris, Ionosphere v.s. WDBC)

co-regularized graph clustering problem, we compare our CGC method with three representative single-domain methods: symmetric NMF (Kuang et al., 2012), K-means (Späth, 1985) and spectral clustering (Ng et al., 2001). We report the accuracy when varying the available cross-domain instance relationships (from 0 to 1 with 10% increment). The accuracy shown in Figure 3.3 is averaged over 100 sets of randomly generated relationships.

We have several key observations from Figure 3.3. First, CGC significantly outperforms all single-domain graph clustering methods, even though single-domain methods may perform differently on different data sets. For example, symmetric NMF works better on Wine and Iris data sets, while K-means works better on Ionosphere and WDBC data sets. Note that when the percentage of available relationships is 0, CGC degrades to symmetric NMF. CGC outperforms all alternative methods when cross-domain relationships are available. This demonstrates the effectiveness of the cross-domain relationship co-regularized method. We also notice that the performance of CGC dramatically improves when the available relationships increase from 0 to 30%, suggesting that our method can effectively improve the clustering result even with limited information on cross-domain relationship. This is crucial for many real-life applications. Finally, we can see that RSS loss is more effective than CD loss. This is because RSS loss directly measures the weights of clustering assignment, while the CD loss does this indirectly by using

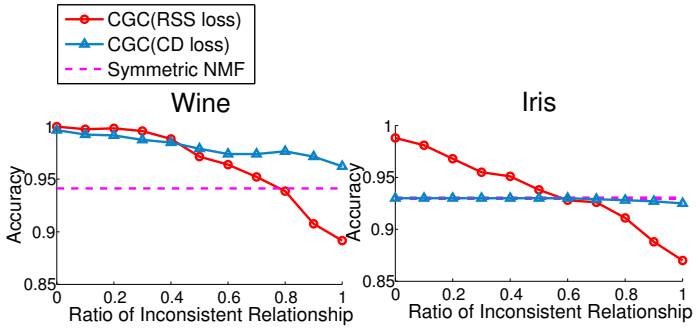


Figure 3.4: Clustering with inconsistent cross-domain relationship

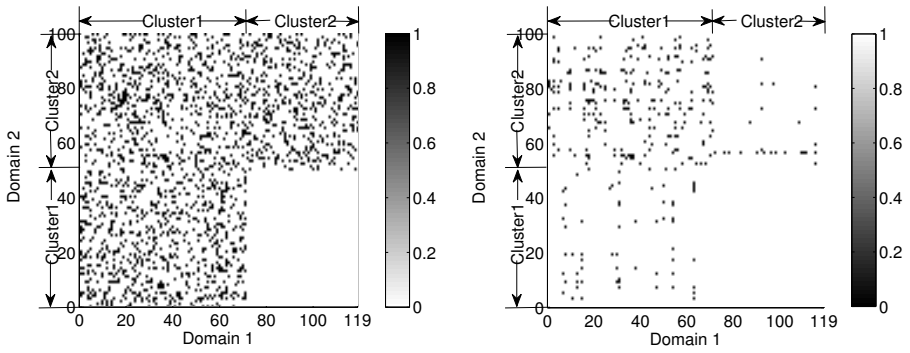
linear kernel similarity first (see Section 3.3.1). Thus, for a given percentage of cross-domain relationships, the method using RSS loss gains more improvements over the single-domain clustering than that using CD loss.

3.4.2 Robustness Evaluation

In real-life applications, both graph data and cross-domain instance relationship may contain noise. In this section, we 1) evaluate whether CGC is sensitive to the inconsistent relationships, and 2) study the effectiveness of the relationship re-evaluation strategy proposed in Section 3.3.5. Due to space limitations, we only report the results on Wine-Iris data set used in the previous section. Similar results can be observed in other data sets.

We add inconsistency into matrix $\mathbf{S}$ with ratio r. The results are shown in Figure 3.4. The percentage of available cross-domain relationships is fixed at 20%. Single-domain symmetric NMF is used as a reference method. We observe that, even when the inconsistency ratio r is close to 50%, CGC still outperforms the single-domain symmetric NMF method. This indicates that our method is robust to noisy relationships. We also observe that, when r is very large, CD loss works better than RSS loss, although when r is small, RSS loss outperforms the CD loss (as discussed in Section 3.4.1). When r reaches 1, the relationship is full of noise. From the figure, we can see that CD loss is immune to noise.

In Section 3.3.5, we provide a method to report the cross-domain relationships that violate the single-domain clustering structure. We still use the Wine-Iris data set to evaluate its



(a) S (b) W

Figure 3.5: Relationship matrix $\mathbf{S}$ and confidence matrix $\mathbf{W}$ on Wine-Iris data set

Group Id	Label
3	comp.os.ms-windows.misc
4	comp.sys.ibm.pc.hardware
5	comp.sys.mac.hardware
9	rec.motorcycles
10	rec.sport.baseball
11	rec.sport.hockey

Table 3.4: The newsgroup data

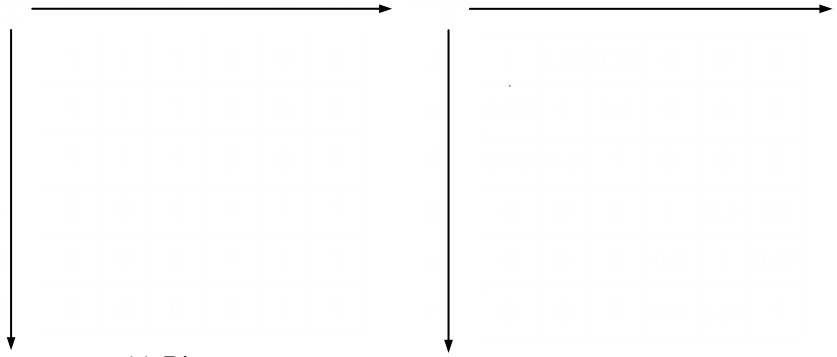
effectiveness. As shown in Figure 3.5, in the relationship matrix $\mathbf{S}$, each black point represents a cross-domain relationship (all with value 1) mapping classes between the two domains. We leave the bottom right part of the matrix blank intentionally so that the inconsistent relationships only appear between instances in cluster 1 of domain 1 and cluster 2 of domain 2. The learned confidence matrix $\mathbf{W}$ is shown in the figure (entries normalized to [0,1]). The smaller the value is, the stronger the evidence that the cross-domain relationship violates the original single-domain clustering structure. Reporting these suspicious relationships to users will allow them to examine the cross-domain relationships that are likely resulting from inaccurate prior knowledge.

3.4.3 Binary v.s. Weighted Relationship

In this section, we demonstrate that CGC can effectively incorporate weighted cross-domain relationship, which may carry richer information than binary relationship. The 20 Newsgroup data set contains documents organized by a hierarchy of topic classes. We choose 6 groups as shown in Table 3.4. For example, at the top level, the 6 groups belong to two topics,

computer (groups {3,4,5}) or recreation (groups {9,10,11}). The computer related data sets can be further partitioned into two subcategories, os (group 3) and sys (groups {4, 5}). Similarly, the recreation related data sets consist of subcategories motocycles (group 9) and sport (groups 10 and 11).

We generate two domains; each contains 300 documents randomly sampled from the 6 groups (50 documents from each group). To generate binary relationships, two articles are related if they are from the same high-level topic, i.e., computer or recreation, as shown in Figure 3.6(a). Weighted relationships are generated based on the topic hierarchy. Given two group labels, we compute the longest common prefix. The weight is assigned to be the ratio of the length of the common prefix over the length of the shorter of the two labels. The weighted relationship matrix is shown in Figure 3.6(b). For example, if two documents come from the same group, we set the corresponding entry to 1; if one document is from rec.sport.baseball and the other from rec.sport.hockey, we set the corresponding entry to 0.67; if they do not share any label term at all, we set the entry to 0.



(a) Binary (b) Weighted

Figure 3.6: Binary and weighted relationship matrices

We perform experiments using binary and weighted relationships respectively. The affinity matrix of documents is computed based on cosine similarity. We cluster the data set into either 2 or 6 clusters and results are shown in Figure 3.7. We observe that when each domain is partitioned into 2 clusters, the binary relationship outperforms the weighted one. This is because the binary relationship better represents the top-level topics, computer and recreation. On the

other hand, for the domain partitioned into 6 clusters, the weighted relationship performs significantly better than the binary one. This is because weights provide more detailed information on cross-domain relationships than the binary relationships.

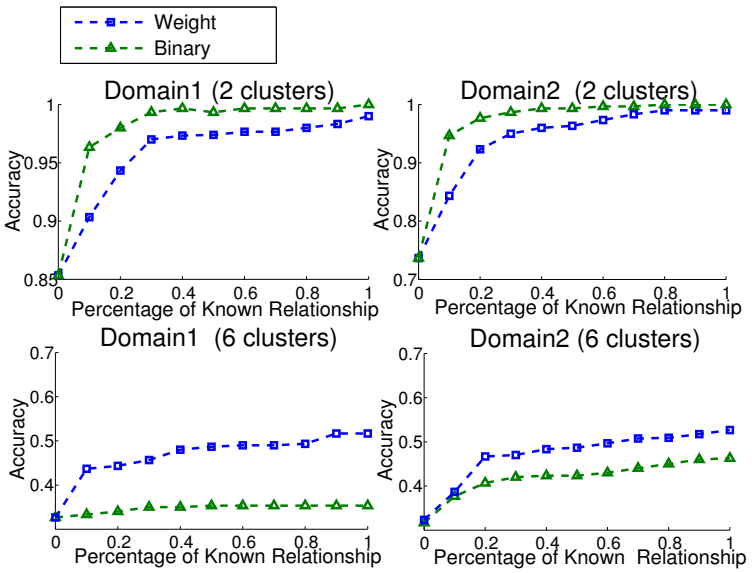


Figure 3.7: Clustering results on the newsgroup data set with binary or weighted relationships

3.4.4 Evaluation of Assigning Optimal λ's Associated with Focused Domain

In this section, we evaluate the effectiveness of the algorithm proposed in Section 3.3.6 to automatically balance different cross-domain regularizers. We perform evaluation using the same setting as in Figure 3.2. We have 6 different domains; each contains 300 documents randomly sampled from the 6 groups (50 documents from each group). Domain π is the one on which the user focuses. There are 5 other domains related to it. Each has randomly selected 20% available cross-domain instance relationships.

Figure 3.8 shows the clustering accuracy of the 5 auxiliary domains and the focused domain π using different methods ($\gamma = 0.05$). We observed that for the focused domain π, the CGC algorithm with equal weights (λ_r=1/5) for regularizers outperforms the single domain clustering (NMF). The CGC algorithm with optimal weights inferred by the algorithm in Section 3.3.6 outperforms the equal weights setting. This demonstrates the effectiveness of the proposed

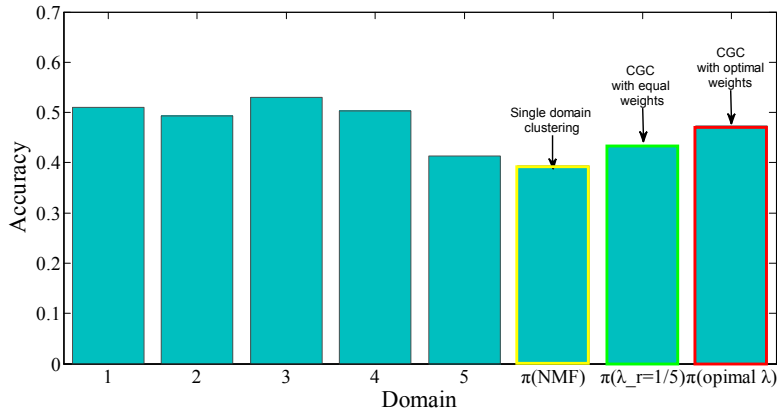


Figure 3.8: Clustering accuracy of the auxiliary(1–5) and the focused domains ($\gamma = 0.05$)

algorithm. In Figure 3.10, we show the clustering accuracy of the case that $\gamma = 0.1$. Similar observation can be made.

Figure 3.9 reports the optimal weights (λ_r) and the corresponding clustering inconsistency μ_r of each auxiliary domain when $\gamma = 0.05$. Clearly, the higher clustering inconsistency between domains r and π, the smaller weight will be assigned to r. These auxiliary domains with large μ_r are treated as noisy domains. In Figure 3.9, only domain 1 and 4 are left when γ is 0.05.

We can further use γ to control how many auxiliary domains will be integrated for graph partition for domain π. Figure 3.11 shows the optimal weights assignments when $\gamma = 0.1$ and $\gamma = 0.15$ respectively. We observed that when γ is large, all domains will be selected, i.e., each λ_r will be a small but non-zero value. In contrast, when γ is small, fewer domains will be selected such as shown in Figure 3.9. This is consistent with what has been discussed in Section 3.3.6.

3.4.5 Protein Module Detection by Integrating Multi-Domain Heterogenous Data

In this section, we apply the proposed method to detect protein functional modules (Hub and de Groot, 2009). The goal is to identify clusters of proteins that have strong interconnection with each other. A common approach is to cluster the protein-protein interaction (PPI) networks (Asur et al., 2007). We show that, by integrating multi-domain heterogeneous information, such as gene co-expression network (Horvath and Dong, 2008) and genetic

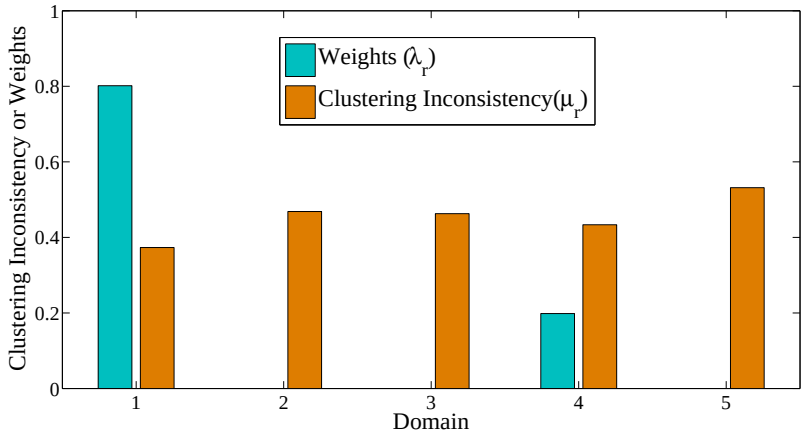


Figure 3.9: Optimal weights (λ_r) and the corresponding μ_r ($\gamma = 0.05$)

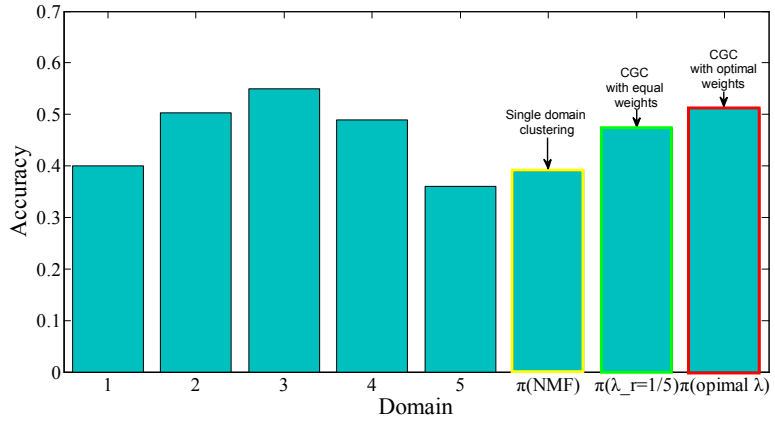


Figure 3.10: Clustering accuracy of auxiliary domains 1–5 and the focused domain ($\gamma = 0.1$)

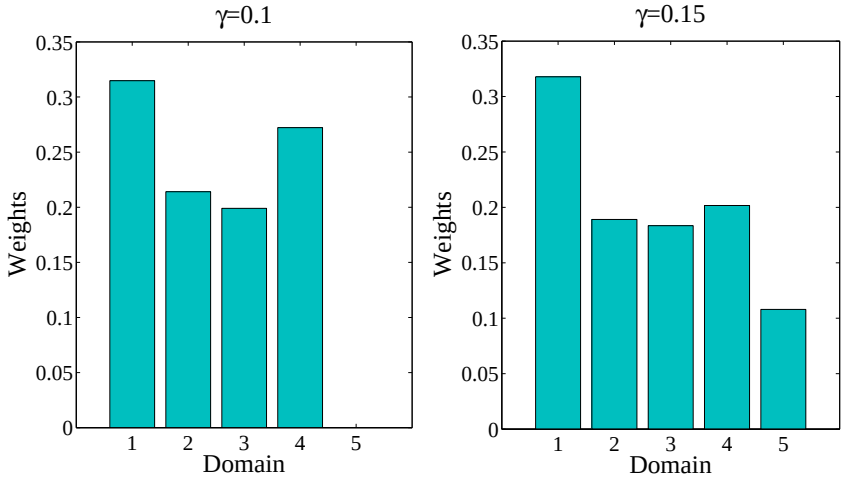


Figure 3.11: Optimal weights (λ_r) of auxiliary domains 1–5 with different γ

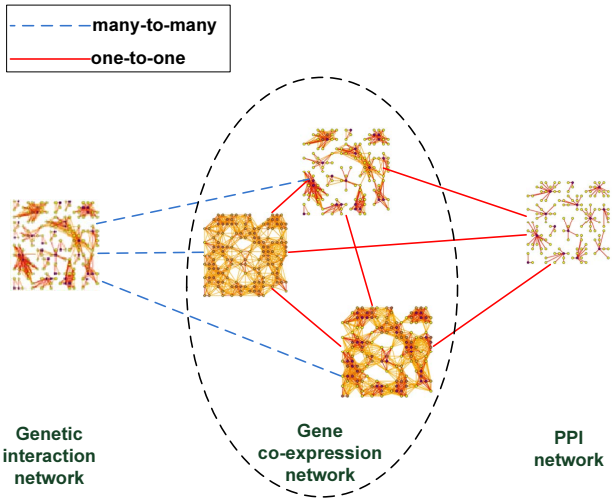
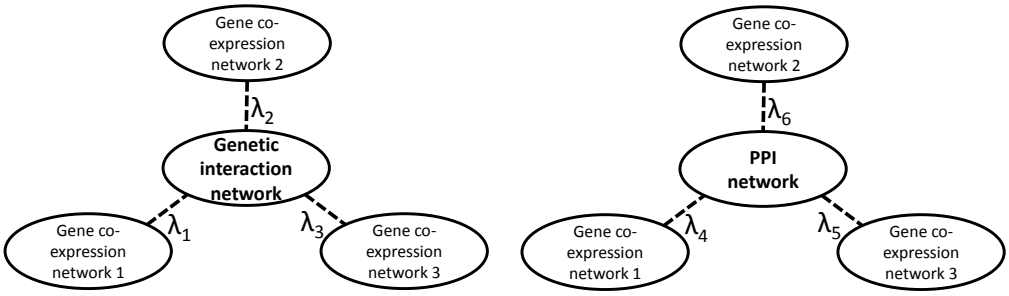


Figure 3.12: PPI network, gene co-expression network, genetic interaction network.

interaction network (Cordell, 2009), the performance of the detection algorithm can be dramatically improved.

We download the widely used human PPI network from BioGrid (http://thebiogrid.org/download.php). Three Hypertension related gene expression data sets are downloaded from Gene Expression Ominbus (http://www.ncbi.nlm.nih.gov/gds) with ids GSE2559, GSE703, and GSE4737. In total, 5412 genes are included in all three data sets used to construct gene co-expression network. Pearson correlation coefficients(normalized between [0 1]) are used as the weights on edges between genes. The genetic interaction network is constructed using a large-scale Hypertension genetic data (Feng and Zhu, 2010), which contains 490032 genetic markers across 4890 (1952 disease and 2938 healthy) samples. We use 1 million top-ranked genetic marker-pairs to construct the network, and the test statistics are used as the weights on the edges between markers (Zhang et al., 2010). The constructed heterogenous networks are shown in Figure 3.12. The relationship between genes and genetic markers is many-to-many, since multiple genetic markers may be covered by a gene, and each marker may be covered by multiple genes due to the overlapping between genes. The relationship between proteins and genes is one-to-one.



(a) star network 1 (b) star network 2

Figure 3.13: Two star networks for inferring optimal weights

We apply CGC (with RSS loss) to cluster the generated multi-domain graphs with two different settings: (1) equal weights for each cross-domain regularizer; (2) optimal weights for each cross-domain relationship. For the first setting, we simply set weights for each cross-domain regularizer to 1. For the second setting, we consider Figure 3.12 as the combination of the two star networks. They have been shown in Figure 3.13. In the first star network, genetic interaction network is the focused domain. In the second star network, PPI network is the focused domain. Then, we execute the algorithm proposed in Section 3.3.6 on the two star networks respectively to assign optimal λ's. Finally, we use these optimal λ's for clustering.

We use the standard Gene Set Enrichment Analysis (GSEA) (Mootha et al., 2003) to evaluate the significance of the inferred clusters. In particular, for each inferred cluster (protein/gene set) T, we identify the most significantly enriched Gene Ontology categories (The Gene Ontology Consortium, 2000; Cheng et al., 2012). The significance, or p-value, is determined by the Fisher's exact test. The raw p-values are further calibrated to correct for the multiple testing problem (Westfall and Young, 1993). To compute calibrated p-values for each T, we perform a randomization test, wherein we apply the same test to 1000 randomly created gene sets that have the same number of genes as T.

The calibrated p-values of the gene sets learned by CGC and single-domain graph clustering methods, symmetric NMF (Kuang et al., 2012), Markov clustering (van Dongen, 2000) and spectral clustering, when applied on PPI network, are shown in Figure 3.14. The clusters are

Method	Number of significant modules
Markov Clustering	21
Spectral Clustering	44
Symmetric NMF	77
CGC(equal weights)	**84**
CGC(optimal weights)	**87**

Table 3.5: GO enrichment analysis of the gene sets identified by different methods

Method	Number of significant modules
LMF (Tang et al., 2009)	13
CSC (Kumar et al., 2011)	15
MO-Pareto (Davidson et al., 2013)	19

Table 3.6: Number of identified protein modules by different methods.

arranged in ascending order of their p-values. As can be seen from the figure, by integrating three types of heterogeneous networks, CGC achieves better performance than the single-domain methods. Table 3.5 shows the number of significant (calibrated p-value ≤ 0.05) modules identified by different methods. We find that CGC reports more significant functional modules than the single-domain methods. The CGC model using optimal weights reports more significant functional modules than those using equal weights. We also apply existing state-of-the-art multi-view graph clustering methods (Kumar et al., 2011; Tang et al., 2009; Davidson et al., 2013) on the gene co-expression networks and PPI network. Since these four networks are of the same size, multi-view method can be applied. LMF (Tang et al., 2009) used a linked matrix factorization model to do multi-view graph clustering. CSC (Kumar et al., 2011) used a centroid-based co-regularized model to do multi-view spectral clustering. MO-Pareto (Davidson et al., 2013) designed a multi-objective optimization model to do multi-view spectral clustering and solve it using Pareto optimization. In total, fewer than 20 significant modules are identified by multi-view graph clustering algorithms on gene co-expression networks and PPI network. This is because the gene expression data are very noisy on this data set. Multi-view graph clustering methods are forced to find one common clustering assignment over different data sets and thus are more sensitive to noise.

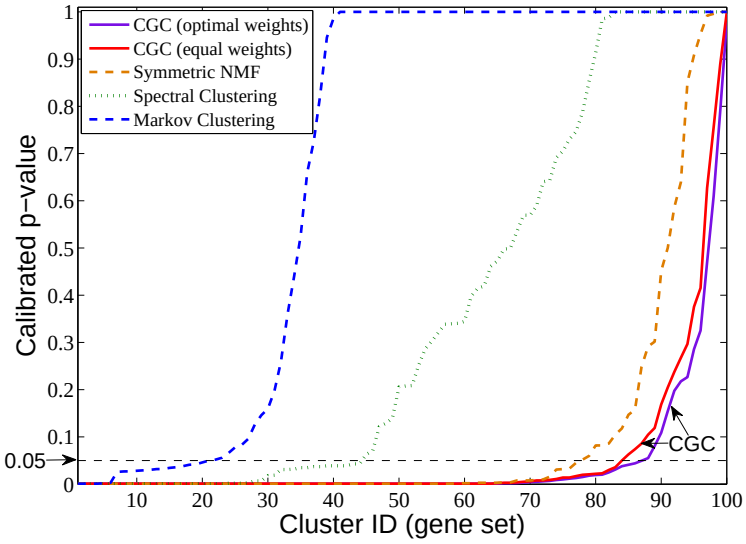


Figure 3.14: Comparison of CGC and single-domain graph clustering ($k = 100$)

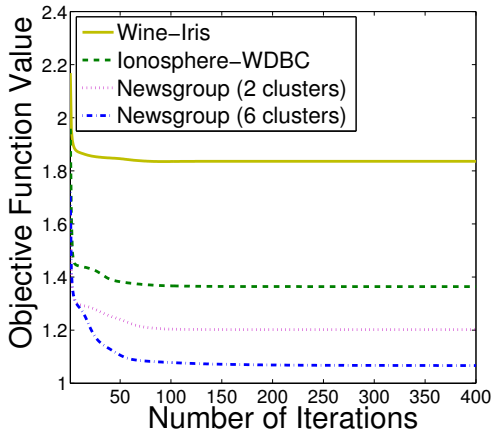


Figure 3.15: Number of iterations to converge (CGC)

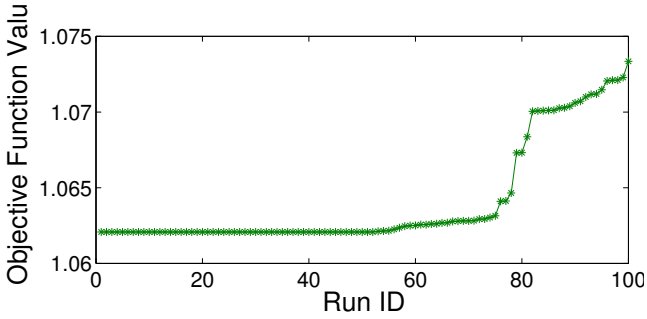


Figure 3.16: Objective function values of 100 runs with random initializations (newsgroup data)

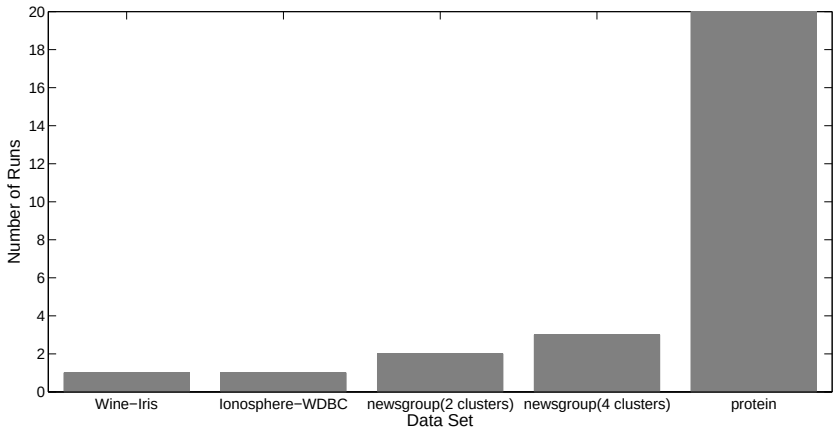


Figure 3.17: Number of runs used for finding global optima

3.4.6 Performance Evaluation

In this section, we study the performance of the proposed methods: the number of iterations before converging to a local optima and the number of runs needed to find the global optima.

Figure 3.15 shows the value of the objective function with respect to the number of iterations on different data sets. We observe that the objective function value decreases steadily with more iterations. Usually, fewer than 100 iterations are needed before convergence. Next, we study the proposed population-based Tabu search algorithm for finding global optima using the newsgroup data sets. Figure 3.16 shows the objective function values (arranged in ascending order) of 100 runs with randomly selected starting points. It can be seen that most runs converge to a global minimum. This observation is consistent with Table 3.2. For example, according to Table 3.2, only 4 runs are needed to find the global optima with confidence 0.999. Thus, the possibility ϕ that a random point converge to a global minimum is very high. Figure 3.17 shows the number of runs used for finding global optima on various data sets. We find that only a few runs are needed to find the global optima.

To further validate the scalability and efficiency of the proposed approach, we report the running time of CGC on each data set in Table 3.7. All experiments are performed (with matlab) on a PC with 2.80 GHz AMD Opteron(tm) 16-core CPU and 32 GB memory. We can observe

Data set	#networks	Largest #nodes	#processors	Time cost
Wine-Iris	2	119	1	0.1 ms
Ionosphere-WDBC	2	569	1	2.1 ms
Newsgroup (4 clusters)	2	300	1	1.3 ms
Protein	5	490032	1	10 hours
Protein(Parallel)	5	490032	4	6 hours
Protein(Parallel)	5	490032	16	3 hours

Table 3.7: Running time on different data sets

that even the largest number of nodes in the graph reaches 490032, the time cost of the algorithm is still reasonable.

3.5 Conclusion

Integrating multiple data sources for graph clustering is an important problem in data mining research. Robust and flexible approaches that can incorporate multiple sources to enhance graph clustering performance are highly desirable. We develop CGC, which utilizes cross-domain relationship as co-regularizing penalty to guide the search of consensus clustering structure. By using a population-based Tabu Search, CGC can be optimized efficiently with guarantee of finding the global optimum with given confidence requirement. CGC is robust even when the cross-domain relationships based on prior knowledge are noisy. Moreover, it is able to automatically identify noisy domains. By assigning smaller weights to noisy domains, the CGC algorithm is able to obtain optimal graph partition performance for the focused domain. Using various benchmark and real-life data sets, we show that the proposed CGC method can dramatically improve the graph clustering performance compared with single-domain methods.

CHAPTER 4: INCORPORATING PRIOR GROUPING KNOWLEDGE

4.1 Introduction

eQTL mapping aims to identify SNPs that influence the expression level of genes. It has been widely applied to dissect the genetic basis of complex traits (Bochner, 2003; Michaelson et al., 2009a). Several important issues need to be considered in eQTL mapping. First, the number of SNPs is usually much larger than the number of samples (Tibshirani, 1996). Second, the existence of confounding factors, such as expression heterogeneity, may result in spurious associations (Listgarten et al., 2010). Third, SNPs (and genes) usually work together to cause variation in complex traits (Michaelson et al., 2009a). The interplay among SNPs and the interplay among genes can be represented as networks and used as prior knowledge (Musani et al., 2007b; Pujana et al., 2007). However, such prior knowledge is far from being complete and may contain a lot of noise. Developing effective models to address these issues in eQTL studies has recently attracted increasing research interests (Biganzoli et al., 2006; Kim and Xing, 2012; Lee et al., 2010; Lee and Xing, 2012).

In eQTL studies, two types of networks can be utilized. One is the genetic interaction network (Charles Boone and Andrews, 2007). Modeling genetic interaction (e.g., epistatic effect between SNPs) is essential to understanding the genetic basis of common diseases, since many diseases are complex traits (Lander, 2011). Another type of network is the network among traits, such as the PPI network or the gene co-expression network. Interacting proteins or genes in a PPI network are likely to be functionally related, i.e., part of a protein complex or in the same biological pathway (von Mering et al., 2002). Effectively utilizing such prior network information can significantly improve the performance of eQTL mapping (Lee and Xing, 2012; Lee et al., 2010).

Figure 4.1 shows an example of eQTL mapping with prior network knowledge. The interactions among SNPs and genes are represented by matrices $\mathbf{S}$ and $\mathbf{G}$ respectively. The goal of eQTL mapping is to infer associations between SNPs and genes represented by the coefficient matrix $\mathbf{W}$. Suppose that SNP ② is strongly associated with gene Ⓒ. Using the network prior, the moderate association between SNP ① and gene Ⓐ may be identified since ① and ②, Ⓐ and Ⓒ have interactions.

To leverage the network prior knowledge, several methods based on Lasso have been proposed (Biganzoli et al., 2006; Kim and Xing, 2012; Lee and Xing, 2012; Lee et al., 2010). The group-lasso penalty is applied to model the genetic interaction network (Biganzoli et al., 2006). Xing et al. consider groupings of genes and apply a multi-task lasso penalty (Kim and Xing, 2012; Lee et al., 2010). They further extend the model to consider grouping information of both SNPs and genes (Lee and Xing, 2012). These methods apply a "hard" clustering of SNPs (genes) so that a SNP (gene) cannot belong to multiple groups. However, a SNP may affect multiple genes and a gene may function in multiple pathways. To address this limitation, Jenatton et al. develop a model allowing overlap between different groups (Jenatton et al., 2011).

Despite their success, there are three common limitations of these group penalty based approaches. First, a clustering step is usually needed to obtain the grouping information. To address this limitation, Xing et al. introduce a network-based fusion penalty on the genes (Kim and Xing, 2009; Li and Li, 2008). However, this method does not consider the genetic interaction network. A two-graph-guided multi-task Lasso approach is developed by Chen et al. (Chen et al., 2012) to make use of gene co-expression network and SNP correlation network. However, this method does not consider the network prior knowledge. The second limitation of the existing methods is that they do not take into consideration the incompleteness of the networks and the noise in them (von Mering et al., 2002). For example, PPI networks may contain false interactions and miss true interactions (von Mering et al., 2002). Directly using the grouping penalty inferred from the noisy and partial prior networks may introduce new bias and thus impair the performance. Third, in addition to the network information, other prior

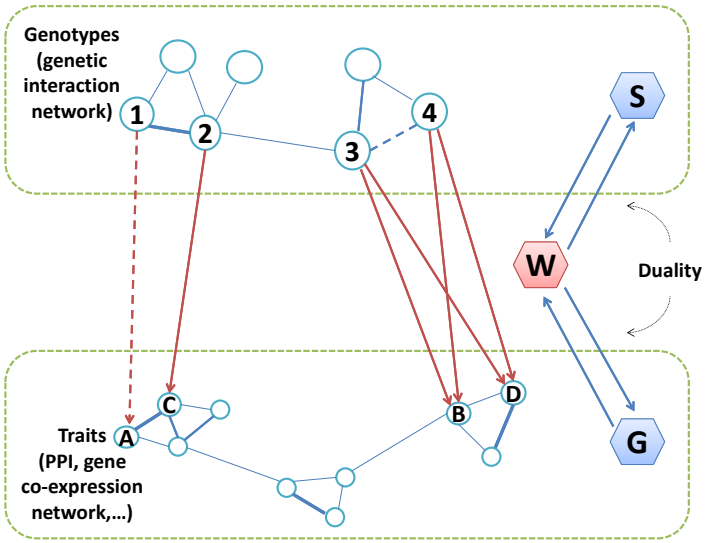


Figure 4.1: Examples of prior knowledge on $\mathbf{S}$ and $\mathbf{G}$.

knowledge, such as location of genetic markers and gene pathway information, are also available. The existing methods cannot incorporate such information.

To address the limitations of the existing methods, this chapter proposes a novel approach, Graph-regularized Dual Lasso (GDL), which simultaneously learns the association between SNPs and genes and refines the prior networks. To support "soft" clustering (allowing genes and SNPs to be members of multiple clusters), we adopt the graph regularizer to encode structured penalties from the prior networks. The penalties encourage the connected nodes (SNPs/genes) to have similar coefficients. This enables us to find multiple-correlated genetic markers with pleiotropic effects that affect multiple-correlated genes jointly. To tackle the problem of noisy and incomplete prior networks, we exploit the *duality* between learning the associations and refining the prior networks to achieve smoother regularization. That is, learning regression coefficients can help to refine the prior networks, and vice versa. For example, in Figure 4.1, if SNPs ③ and ④ have strong associations with the same group of genes, they are likely to have interaction, which is not captured in the prior network. An ideal model should allow an update to the prior network according to the learned regression coefficients. GDL can also incorporate other available prior knowledge such as the physical location of SNPs and biology pathways to which the genes belong. The resultant optimization problem is convex and can be efficiently solved by

Symbols	Description
K	Number of SNPs
N	Number of genes
D	Number of samples
$\mathbf{X} \in \mathbb{R}^{K \times D}$	The SNP matrix data
$\mathbf{Z} \in \mathbb{R}^{N \times D}$	The gene matrix data
$\mathbf{L} \in \mathbb{R}^{N \times D}$	A low-rank matrix
$\mathbf{S}_0 \in \mathbb{R}^{K \times K}$	The input affinity matrices of the genetic interaction network
$\mathbf{G}_0 \in \mathbb{R}^{N \times N}$	The input affinity matrices of the network of traits
$\mathbf{S} \in \mathbb{R}^{K \times K}$	The refined affinity matrices of the genetic interaction network
$\mathbf{G} \in \mathbb{R}^{N \times N}$	The refined affinity matrices of the network of traits
$\mathbf{W} \in \mathbb{R}^{N \times K}$	The coefficient matrix to be inferred
$\mathcal{R}^{(S)}$	The graph regularizer from the genetic interaction network
$\mathcal{R}^{(G)}$	The graph regularizer from the PPI network
$\mathcal{D}(\cdot, \cdot)$	A nonnegative distance measure

Table 4.1: Summary of Notations

using an alternating minimization procedure. We perform extensive empirical evaluation of the proposed method using both simulated and real eQTL datasets. The results demonstrate that GDL is robust to the incomplete and noisy prior knowledge and can significantly improve the accuracy of eQTL mapping compared to the state-of-the-art methods.

4.2 Background: Linear Regression with Graph Regularizer

Throughout the chapter, we assume that, for each sample, the SNPs and genes are represented by column vectors. Important notations are listed in Table 4.1. Let $\mathbf{x} = [x_1, x_2, \ldots, x_K]^{\mathrm{T}}$ represent the K SNPs in the study, where $x_i \in \{0, 1, 2\}$ is a random variable corresponding to the i-th SNP. For example, 0, 1, 2 may encode the homozygous major allele, heterozygous allele, and homozygous minor allele, respectively. Let $\mathbf{z} = [z_1, z_2, \ldots, z_N]^{\mathrm{T}}$ represent expression levels of the N genes in the study, where z_j is a continuous random variable corresponding to the j-th gene. The traditional linear regression model for association mapping between $\mathbf{x}$ and $\mathbf{z}$ is

$$\mathbf{z} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \epsilon, \tag{4.1}$$

where $\mathbf{z}$ is a linear function of $\mathbf{x}$ with coefficient matrix $\mathbf{W}$. $\boldsymbol{\mu}$ is an $N \times 1$ translation factor vector. ϵ is the additive noise of Gaussian distribution with zero-mean and variance $\gamma\mathbf{I}$, where γ is a scalar. That is, $\epsilon \sim \mathcal{N}(\mathbf{0}, \gamma\mathbf{I})$.

The question now is how to define an appropriate objective function over $\mathbf{W}$ that 1) can effectively incorporate the prior network knowledge, and 2) is robust to the noise and incompleteness in the prior knowledge. Next, we first briefly review Lasso and its variations and then introduce the proposed GD-Lasso method.

4.2.1 Lasso and LORS

Lasso (Tibshirani, 1996) is a method for estimating the regression coefficients $\mathbf{W}$ using ℓ_1 penalty for sparsity. It has been widely used for association mapping problems. Let $\mathbf{X} = \{\mathbf{x}_d | 1 \leq d \leq D\} \in \mathbb{R}^{K \times D}$ be the SNP matrix and $\mathbf{Z} = \{\mathbf{z}_d | 1 \leq d \leq D\} \in \mathbb{R}^{N \times D}$ be the gene expression matrix. Each column of $\mathbf{X}$ and $\mathbf{Z}$ stands for one sample. The objective function of Lasso is

$$\min_{\mathbf{W}} \frac{1}{2} ||\mathbf{Z} - \mathbf{W}\mathbf{X} - \boldsymbol{\mu}\mathbf{1}||_F^2 + \eta ||\mathbf{W}||_1 \tag{4.2}$$

where $||\cdot||_F$ denotes the Frobenius norm, $||\cdot||_1$ is the ℓ_1-norm. $\mathbf{1}$ is an $1 \times D$ vector of all 1's. η is the empirical parameter for the ℓ_1 penalty. $\mathbf{W}$ is the parameter (also called weight) matrix parameterizing the space of linear functions mapping from $\mathbf{X}$ to $\mathbf{Z}$.

Confounding factors, such as unobserved covariates, experimental artifacts, and unknown environmental perturbations, may mask real signals and lead to spurious findings. LORS (Yang et al., 2013) uses a low-rank matrix $\mathbf{L} \in \mathbb{R}^{N \times D}$ to account for the variations caused by hidden factors. The objective function of LORS is

$$\min_{\mathbf{W}, \boldsymbol{\mu}, \mathbf{L}} \frac{1}{2} ||\mathbf{Z} - \mathbf{W}\mathbf{X} - \boldsymbol{\mu}\mathbf{1} - \mathbf{L}||_F^2 + \eta ||\mathbf{W}||_1 + \lambda ||\mathbf{L}||_* \tag{4.3}$$

where $||\cdot||_*$ is the nuclear norm. η is the empirical parameter for the ℓ_1 penalty to control the sparsity of $\mathbf{W}$, and λ is the regularization parameter to control the rank of $\mathbf{L}$. $\mathbf{L}$ is a low-rank matrix assuming that there are only a small number of hidden factors influencing the gene expression levels.

4.2.2 Graph-regularized Lasso

To incorporate the network prior knowledge, group sparse Lasso (Biganzoli et al., 2006), multi-task Lasso (Obozinski and Taskar, 2006) and SIOL (Lee and Xing, 2012) have been

proposed. Group sparse Lasso makes use of grouping information of SNPs; multi-task Lasso makes use of grouping information of genes, while SIOL uses information from both networks. A common drawback of these methods is that the number of groups (SNP and gene clusters) has to be predetermined. To overcome this drawback, we propose to use two graph regularizers to encode the prior network information. Compared with the previous group penalty based methods, our method does not need to pre-cluster the networks and thus may obtain smoother regularization. Moreover, these methods do not consider confounding factors that may mask real signals and lead to spurious findings. In this chapter, we further incorporate the idea in LORS (Yang et al., 2013) to tackle the confounding factors simultaneously.

Let $\mathbf{S}_0 \in \mathbb{R}^{K\times K}$ and $\mathbf{G}_0 \in \mathbb{R}^{N\times N}$ be the affinity matrices of the genetic interaction network (e.g., epistatic effect between SNPs) and network of traits (e.g., PPI network or gene co-expression network), and $\mathbf{D}_{S_0}$ and $\mathbf{D}_{G_0}$ be their degree matrices. Given the two networks, we can employ a pairwise comparison between $\mathbf{w}_{*i}$ and $\mathbf{w}_{*j}$ $(1 \leq i < j \leq K)$: if SNPs i and j are closely related, $||\mathbf{w}_{*i} - \mathbf{w}_{*j}||_2^2$ is small. The pairwise comparison can be naturally encoded in the *weighted fusion penalty* $\sum_{ij} ||\mathbf{w}_{*i} - \mathbf{w}_{*j}||_2^2 (\mathbf{S}_0)_{i,j}$. This penalty will enforce $||\mathbf{w}_{*i} - \mathbf{w}_{*j}||_2^2 = 0$ for closely related SNP pairs (with large $(\mathbf{S}_0)_{i,j}$ value). Then, the graph regularizer from the genetic interaction network takes the following form

$$\begin{aligned} \mathcal{R}^{(S)} &= \frac{1}{2}\sum_{ij} ||\mathbf{w}_{*i} - \mathbf{w}_{*j}||_2^2 (\mathbf{S}_0)_{i,j} \\ &= \mathrm{tr}(\mathbf{W}(\mathbf{D}_{S_0} - \mathbf{S}_0)\mathbf{W}^{\mathrm{T}}) \end{aligned} \tag{4.4}$$

Similarly, the graph regularizer for the network of traits is

$$\mathcal{R}^{(G)} = \mathrm{tr}(\mathbf{W}^{\mathrm{T}}(\mathbf{D}_{G_0} - \mathbf{G}_0)\mathbf{W}) \tag{4.5}$$

These two regularizers encourage the connected nodes in a graph to have similar coefficients. A heavy penalty occurs if the learned regression coefficients for neighboring SNPs (genes) are disparate. $(\mathbf{D}_{S_0} - \mathbf{S}_0)$ and $(\mathbf{D}_{G_0} - \mathbf{G}_0)$ are known as the combinatorial graph Laplacian, which are positive semi-definite (Chung, 1997). Graph-regularized Lasso (G-Lasso) solves the following

optimization problem

$$\min_{\mathbf{W},\boldsymbol{\mu},\mathbf{L}} \frac{1}{2}||\mathbf{Z}-\mathbf{W}\mathbf{X}-\mu\mathbf{1}-\mathbf{L}||_F^2 + \eta||\mathbf{W}||_1 + \lambda||\mathbf{L}||_* + \alpha\mathcal{R}^{(S)} + \beta\mathcal{R}^{(G)} \tag{4.6}$$

where $\alpha, \beta > 0$ are regularization parameters.

4.3 Graph-regularized Dual Lasso

In eQTL studies, the prior knowledge is usually incomplete and contains noise. It is desirable to refine the prior networks according to the learned regression coefficients. There is a *duality* between the prior networks and the regression coefficients: learning coefficients can help to refine the prior networks, and vice versa. This leads to mutual reinforcement when learning the two parts simultaneously.

Next, we introduce the Graph-regularized Dual Lasso (GD-Lasso). We further relax the constraints from the prior networks (two graph regularizers) introduced in Section 4.2.2, and integrate the graph-regularized Lasso and the dual refinement of graphs into a unified objective function

$$\begin{aligned}\min_{\mathbf{W},\boldsymbol{\mu},\mathbf{L},\mathbf{S}\geq 0,\mathbf{G}\geq 0} & \frac{1}{2}||\mathbf{Z}-\mathbf{W}\mathbf{X}-\mu\mathbf{1}-\mathbf{L}||_F^2 + \eta||\mathbf{W}||_1 + \lambda||\mathbf{L}||_* \\ & + \alpha\mathrm{tr}(\mathbf{W}(\mathbf{D}_S-\mathbf{S})\mathbf{W}^{\mathrm{T}}) + \beta\mathrm{tr}(\mathbf{W}^{\mathrm{T}}(\mathbf{D}_G-\mathbf{G})\mathbf{W}) \\ & + \gamma||\mathbf{S}-\mathbf{S}_0||_F^2 + \rho||\mathbf{G}-\mathbf{G}_0||_F^2\end{aligned} \tag{4.7}$$

where $\gamma, \rho > 0$ are positive parameters controlling the extent to which the refined networks should be consistent with the original prior networks. $\mathbf{D}_S$ and $\mathbf{D}_G$ are the degree matrices of $\mathbf{S}$ and $\mathbf{G}$. Note that the objective function considers the non-negativity of $\mathbf{S}$ and $\mathbf{G}$. As an extension, the model can be extended easily to incorporate prior knowledge from multiple sources. We only need to revise the last two terms in Eq. 4.7 to $\gamma\sum_{i=1}^{f}||\mathbf{S}-\mathbf{S}_i||_F^2 + \rho\sum_{i=1}^{e}||\mathbf{G}-\mathbf{G}_i||_F^2$, where f and e are the number of sources for genetic interaction networks and gene trait networks respectively.

4.3.1 Optimization: An Alternating Minimization Approach

In this section, we present an alternating scheme to optimize the objective function in Eq. (4.7) based on block coordinate techniques. We divide the variables into three sets: $\{\mathbf{L}\}$,$\{\mathbf{S},\mathbf{G}\}$,

and $\{\mathbf{W},\boldsymbol{\mu}\}$. We iteratively update one set of variables while fixing the other two sets. This procedure continues until convergence. Since the objective function is convex, the algorithm will converge to a global optima. The optimization process is as follows. The detailed algorithm is included in Algorithm 3.

Algorithm 3: Graph-regularized Dual Lasso (GD-Lasso)

Input: $\mathbf{X} = \{\mathbf{x}_d\} \in \mathbb{R}^{K\times D}$, $\mathbf{Z} = \{\mathbf{z}_d\} \in \mathbb{R}^{N\times D}$, $\mathbf{S}_0 \in \mathbb{R}^{K\times K}$, $\mathbf{G}_0 \in \mathbb{R}^{N\times N}$, η,α,β,γ,ρ
Output: $\mathbf{W}$,$\boldsymbol{\mu}$,$\mathbf{S}$,$\mathbf{G}$,$\mathbf{L}$

```
begin
    Initialize W using Eq. (4.2), μ ← 0, S ← rand(K, K), G ← rand(N, N);
    repeat
        Update L by Eq. (4.9);
        repeat
            Update S by Eq. (4.10);
            Update G by Eq. (4.11);
        until convergence;
        Update W by the coordinate descent algorithm (4.15);
        Update μ by Eq. (4.17);
    until convergence;
end
```

(1). While fixing $\{\mathbf{W}, \boldsymbol{\mu}\}$, $\{\mathbf{S}, \mathbf{G}\}$, optimize $\{\mathbf{L}\}$ using singular value decomposition (SVD).

Lemma 4.1. *(Mazumder et al., 2010) Suppose that matrix* $\mathbf{A}$ *has rank* r*. The solution to the optimization problem*

$$\min_{\mathbf{B}} \frac{1}{2}||\mathbf{A}-\mathbf{B}||_F^2 + \lambda||\mathbf{B}||_* \tag{4.8}$$

is given by $\widehat{\mathbf{B}} = \mathbf{H}_\lambda(\mathbf{A})$*, where* $\mathbf{H}_\lambda(\mathbf{A}) = \mathbf{U}\mathbf{D}_\lambda\mathbf{V}^{\mathrm{T}}$ *with* $\mathbf{D}_\lambda = \mathrm{diag}[(d_1-\lambda)_+, ..., (d_r-\lambda)_+]$*,* $\mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$ *is the Singular Value Decomposition (SVD) of* $\mathbf{A}$*,* $\mathbf{D} = \mathrm{diag}[d_1, ..., d_r]$*, and* $(d_i-\lambda)_+ = \max((d_i-\lambda), 0), (1 \le i \le r)$*.*

Thus, for fixed $\mathbf{W}, \boldsymbol{\mu}, \mathbf{S}, \mathbf{G}$, the formula for updating $\mathbf{L}$ is

$$\mathbf{L} \leftarrow \mathbf{H}_\lambda(\mathbf{Z} - \mathbf{W}\mathbf{X} - \mu\mathbf{1}) \tag{4.9}$$

(2). While fixing $\{\mathbf{W}, \boldsymbol{\mu}\}$, $\{\mathbf{L}\}$, optimize $\{\mathbf{S}, \mathbf{G}\}$ using semi-nonnegative matrix factorization (semi-NMF) multiplicative updating on $\mathbf{S}$ and $\mathbf{G}$ iteratively (Ding et al., 2010). For

the optimization with non-negative constraints, our updating rule is based on the following two theorems. The proofs of the theorems are given in Section 4.3.2.

Theorem 4.1. *For fixed* $\mathbf{L}, \boldsymbol{\mu}, \mathbf{W}$, *and* $\mathbf{G}$, *updating* $\mathbf{S}$ *according to Eq.* (4.10) *monotonically decreases the value of the objective function in Eq.* (4.7) *until convergence.*

$$\mathbf{S} \leftarrow \mathbf{S} \circ \frac{\alpha(\mathbf{W}^{\mathrm{T}}\mathbf{W})^{+} + 2\gamma\mathbf{S}_0}{2\gamma\mathbf{S} + \alpha(\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-} + \alpha\,\mathrm{diag}(\mathbf{W}^{\mathrm{T}}\mathbf{W})\mathbf{J}_K} \tag{4.10}$$

where $\mathbf{J}_K$ *is a* $K \times K$ *matrix of all 1's.* $\circ$, $\frac{[\cdot]}{[\cdot]}$ *are element-wise operators. Since* $\mathbf{W}^{\mathrm{T}}\mathbf{W}$ *may take mixed signs, we denote* $\mathbf{W}^{\mathrm{T}}\mathbf{W} = (\mathbf{W}^{\mathrm{T}}\mathbf{W})^{+} - (\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-}$, *where* $(\mathbf{W}^{\mathrm{T}}\mathbf{W})^{+}_{i,j} = (|(\mathbf{W}^{\mathrm{T}}\mathbf{W})_{i,j}| + (\mathbf{W}^{\mathrm{T}}\mathbf{W})_{i,j})/2$ *and* $(\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-}_{i,j} = (|(\mathbf{W}^{\mathrm{T}}\mathbf{W})_{i,j}| - (\mathbf{W}^{\mathrm{T}}\mathbf{W})_{i,j})/2$.

Theorem 4.2. *For fixed* $\mathbf{L}, \boldsymbol{\mu}, \mathbf{W}$, *and* $\mathbf{S}$, *updating* $\mathbf{G}$ *according to Eq.* (4.11) *monotonically decreases the value of the objective function in Eq.* (4.7) *until convergence.*

$$\mathbf{G} \leftarrow \mathbf{G} \circ \frac{\beta(\mathbf{W}\mathbf{W}^{\mathrm{T}})^{+} + 2\rho\mathbf{G}_0}{2\rho\mathbf{G} + \beta(\mathbf{W}\mathbf{W}^{\mathrm{T}})^{-} + \beta\,\mathrm{diag}(\mathbf{W}\mathbf{W}^{\mathrm{T}})\mathbf{J}_N} \tag{4.11}$$

where $\mathbf{J}_N$ *is an* $N \times N$ *matrix of all 1's.*

The above two theorems are derived from the KKT complementarity condition (Boyd and Vandenberghe, 2004). We show the updating rule for $\mathbf{S}$ below. The analysis for $\mathbf{G}$ is similar and omitted. We first formulate the Lagrange function of $\mathbf{S}$ for optimization

$$L(\mathbf{S}) = \alpha\mathrm{tr}(\mathbf{W}(\mathbf{D}_S - \mathbf{S})\mathbf{W}^{\mathrm{T}}) + \gamma||\mathbf{S} - \mathbf{S}_0||_F^2 \tag{4.12}$$

The partial derivative of the Lagrange function with respect to $\mathbf{S}$ is

$$\nabla_{\mathbf{S}} L = -\alpha\mathbf{W}^{\mathrm{T}}\mathbf{W} - 2\gamma\mathbf{S}_0 + 2\gamma\mathbf{S} + \alpha\,\mathrm{diag}(\mathbf{W}^{\mathrm{T}}\mathbf{W})\mathbf{J}_K \tag{4.13}$$

Using the KKT complementarity condition for the non-negative constraint on $\mathbf{S}$, we have

$$\nabla_{\mathbf{S}} L \circ \mathbf{S} = \mathbf{0} \tag{4.14}$$

The above formula leads to the updating rule for $\mathbf{S}$ in Eq. (4.10). It has been shown that the multiplicative updating algorithm has first order convergence rate (Ding et al., 2010).

(3). While fixing $\{\mathbf{L}\}$, $\{\mathbf{S}, \mathbf{G}\}$, optimize $\{\mathbf{W}, \boldsymbol{\mu}\}$ using the coordinate descent algorithm.

Because we use the ℓ_1 penalty on $\mathbf{W}$, we can use the coordinate descent algorithm for the optimization of $\mathbf{W}$, which gives the following updating formula:

$$\mathbf{W}_{i,j} = \frac{F(m(i,j),\eta)}{(\mathbf{X}\mathbf{X}^{\mathrm{T}})_{j,j} + 2\alpha(\mathbf{D}_{\mathbf{S}} - \mathbf{S})_{j,j} + 2\beta(\mathbf{D}_{\mathbf{G}} - \mathbf{G})_{i,i}} \tag{4.15}$$

where $F(m(i,j),\eta) = sign(m(i,j))\max(|m(i,j)| - \eta, 0)$, and

$$\begin{aligned} m(i,j) = (\mathbf{Z}\mathbf{X}^{\mathrm{T}})_{i,j} - \sum_{\substack{k=1 \\ k\neq j}}^{K} \mathbf{W}_{i,k}(\mathbf{X}\mathbf{X}^{\mathrm{T}})_{k,j} \\ - 2\alpha \sum_{\substack{k=1 \\ k\neq j}}^{K} \mathbf{W}_{i,k}(\mathbf{D}_{\mathbf{S}} - \mathbf{S})_{k,j} - 2\beta \sum_{\substack{k=1 \\ k\neq i}}^{N} (\mathbf{D}_{\mathbf{G}} - \mathbf{G})_{i,k}\mathbf{W}_{k,j} \end{aligned} \tag{4.16}$$

The solution of updating $\boldsymbol{\mu}$ can be derived by setting $\nabla_{\boldsymbol{\mu}} L(\boldsymbol{\mu}) = 0$, which gives

$$\boldsymbol{\mu} = \frac{(\mathbf{Z} - \mathbf{W}\mathbf{X})\mathbf{1}^{\mathrm{T}}}{D} \tag{4.17}$$

4.3.2 Convergence Analysis

In the following, we investigate the convergence of the algorithm. First, we study the convergence for the second step. We use the auxiliary function approach (Lee and Seung, 2000) to analyze the convergence of the multiplicative updating formulas. Here we first introduce the definition of auxiliary function.

Definition 4.1. Given a function $L(h)$ of any parameter h, a function $Z(h, \tilde{h})$ is an auxiliary function for $L(h)$ if the conditions

$$Z(h,\tilde{h}) \geq L(h) \qquad and \qquad Z(h,h) = L(h), \tag{4.18}$$

are satisfied for any given $h, \tilde{h}$ (Lee and Seung, 2000).

Lemma 4.2. *If Z is an auxiliary function for function $L(h)$, then $L(h)$ is non-increasing under the update (Lee and Seung, 2000).*

$$h^{(t+1)} = \arg\min_{h} Z(h, h^{(t)}) \tag{4.19}$$

Theorem 4.3. *Let $L(\mathbf{S})$ denote the Lagrange function of $\mathbf{S}$ for optimization. The following function*

$$\begin{aligned} Z(\mathbf{S},\widetilde{\mathbf{S}}) = \alpha \sum_{ijk} \mathbf{W}_{i,j}^2 \frac{\mathbf{S}_{j,k}^2 + \widetilde{\mathbf{S}}_{j,k}^2}{2\widetilde{\mathbf{S}}_{j,k}} + \alpha \sum_{ijk} (\mathbf{W}_{i,j}\mathbf{W}_{i,k})^{-} \frac{\mathbf{S}_{j,k}^2 + \widetilde{\mathbf{S}}_{j,k}^2}{2\widetilde{\mathbf{S}}_{j,k}} \\ - \alpha \sum_{ijk} (\mathbf{W}_{i,j}\mathbf{W}_{i,k})^{+} \widetilde{\mathbf{S}}_{j,k}(1 + \log \frac{\mathbf{S}_{j,k}}{\widetilde{\mathbf{S}}_{j,k}}) + \gamma \sum_{jk} \mathbf{S}_{j,k}^2 \\ - 2\gamma \sum_{jk} (\mathbf{S}_0)_{j,k} \widetilde{\mathbf{S}}_{j,k}(1 + \log \frac{\mathbf{S}_{j,k}}{\widetilde{\mathbf{S}}_{j,k}}) + \gamma \sum_{jk} (\mathbf{S}_0)_{j,k}^2. \end{aligned} \tag{4.20}$$

is an auxiliary function for $L(\mathbf{S})$*. Furthermore, it is a convex function in* $\mathbf{S}$ *and its global minimum is*

$$\mathbf{S} = \widetilde{\mathbf{S}} \circ \frac{\alpha(\mathbf{W}^T\mathbf{W})^+ + 2\gamma\mathbf{S}_0}{2\gamma\widetilde{\mathbf{S}} + \alpha(\mathbf{W}^T\mathbf{W})^- + \alpha\,\mathrm{diag}(\mathbf{W}^T\mathbf{W})\mathbf{J}_K}. \tag{4.21}$$

Theorem 4.3 can be proved using a similar idea to that in (Ding et al., 2006) by validating three **Properties**: 1) $L(\mathbf{S}) \leq Z(\mathbf{S}, \widetilde{\mathbf{S}})$; 2) $L(\mathbf{S}) = Z(\mathbf{S}, \mathbf{S})$; 3) $Z(\mathbf{S}, \widetilde{\mathbf{S}})$ is convex with respect to $\mathbf{S}$. The formal proof is provided below.

Proof: We will prove the three properties respectively. The Lagrange function of $\mathbf{S}$ for optimization is

$$L(\mathbf{S}) = \alpha\mathrm{tr}(\mathbf{W}(\mathbf{D}_\mathbf{S} - \mathbf{S})\mathbf{W}^T) + \gamma||\mathbf{S} - \mathbf{S}_0||_F^2. \tag{4.22}$$

To prove **Properties** 1 and 2, we first deduce the following identities:

$$\mathrm{tr}(\mathbf{W}\mathbf{D}_\mathbf{S}\mathbf{W}^T) \quad = \sum_{ijk} \mathbf{W}_{i,j}^2\mathbf{S}_{j,k}. \tag{4.23}$$

Similarly,

$$\mathrm{tr}(\mathbf{W}\mathbf{S}\mathbf{W}^\mathrm{T}) = \sum_{ijk} \mathbf{W}_{i,j}\mathbf{W}_{i,k}\mathbf{S}_{j,k}. \tag{4.24}$$

And,

$$\begin{aligned} ||\mathbf{S} - \mathbf{S}_0||_F^2 =& \mathrm{tr}(\mathbf{S}\mathbf{S}^\mathrm{T}) - 2\mathrm{tr}(\mathbf{S}_0\mathbf{S}^\mathrm{T}) + \mathrm{tr}(\mathbf{S}_0\mathbf{S}_0^\mathrm{T}) \\ =& \sum_{jk} \mathbf{S}_{j,k}^2 - 2\sum_{jk}(\mathbf{S}_0)_{j,k}\mathbf{S}_{j,k} + \sum_{jk}(\mathbf{S}_0)_{j,k}^2. \end{aligned} \tag{4.25}$$

Using identities (4.23), (4.24), and (4.25), and substituting $\widetilde{\mathbf{S}}$ with $\mathbf{S}$ in function (4.20), we get the identity for **Property 2**.

Further, note that $a \leq \frac{a^2+b^2}{2b}$ and $a \geq b(1 + \log\frac{a}{b})$ for all positive a and b, and we have:

- for (4.23),

$$\sum_{ijk} \mathbf{W}_{i,j}^2\mathbf{S}_{j,k} \leq \sum_{ijk} \mathbf{W}_{i,j}^2 \frac{\mathbf{S}_{j,k}^2 + \widetilde{\mathbf{S}}_{j,k}^2}{2\widetilde{\mathbf{S}}_{j,k}};$$

- for (4.24),

$$\begin{aligned} &\sum_{ijk} \mathbf{W}_{i,j}\mathbf{W}_{i,k}\mathbf{S}_{j,k} \\ =& \sum_{ijk} (\mathbf{W}_{i,j}\mathbf{W}_{i,k})^+\mathbf{S}_{j,k} - \sum_{ijk} (\mathbf{W}_{i,j}\mathbf{W}_{i,k})^-\mathbf{S}_{j,k} \\ \geq& \sum_{ijk} (\mathbf{W}_{i,j}\mathbf{W}_{i,k})^+\widetilde{\mathbf{S}}_{j,k}(1 + \log\frac{\mathbf{S}_{j,k}}{\widetilde{\mathbf{S}}_{j,k}}) \\ &- \sum_{ijk} (\mathbf{W}_{i,j}\mathbf{W}_{i,k})^- \frac{\mathbf{S}_{j,k}^2 + \widetilde{\mathbf{S}}_{j,k}^2}{2\widetilde{\mathbf{S}}_{j,k}}; \end{aligned} \tag{4.26}$$

- for the second term in (4.25),

$$\sum_{jk}(\mathbf{S}_0)_{j,k}\mathbf{S}_{j,k} \geq 2\sum_{jk}(\mathbf{S}_0)_{j,k}\widetilde{\mathbf{S}}_{j,k}(1+\log\frac{\mathbf{S}_{j,k}}{\widetilde{\mathbf{S}}_{j,k}})$$

These inequalities together prove **Property 1**.

For **Property 3**, we instead prove the Hessian matrix $\nabla\nabla_{\mathbf{S}}Z(\mathbf{S},\tilde{\mathbf{S}}) \succeq \mathbf{0}$

$$\begin{aligned}
&\frac{\partial Z(\mathbf{S},\widetilde{\mathbf{S}})}{\partial \mathbf{S}_{m,n}}\\
=&\alpha\sum_i \mathbf{W}_{i,m}^2\frac{\mathbf{S}_{m,n}}{\widetilde{\mathbf{S}}_{m,n}} + \alpha\sum_i(\mathbf{W}_{i,m}\mathbf{W}_{i,n})^-\frac{\mathbf{S}_{m,n}}{\widetilde{\mathbf{S}}_{m,n}}\\
&-\alpha\sum_i(\mathbf{W}_{i,m}\mathbf{W}_{i,n})^+\frac{\widetilde{\mathbf{S}}_{m,n}}{\boldsymbol{S}_{m,n}} + 2\gamma\mathbf{S}_{m,n} - 2\gamma(\mathbf{S}_0)_{m,n}\frac{\widetilde{\mathbf{S}}_{m,n}}{\mathbf{S}_{m,n}}.
\end{aligned} \tag{4.27}$$

Hence,

$$\begin{aligned}
&\frac{\partial^2 Z(\mathbf{S},\widetilde{\mathbf{S}})}{\partial \mathbf{S}_{s,t}\partial \mathbf{S}_{m,n}}\\
=&\alpha\sum_i \delta_{ms}\delta_{nt}\mathbf{W}_{i,m}^2\frac{1}{\widetilde{\mathbf{S}}_{m,n}} + \alpha\sum_i \delta_{ms}\delta_{nt}(\mathbf{W}_{i,m}\mathbf{W}_{i,n})^-\frac{1}{\widetilde{\mathbf{S}}_{m,n}}\\
&+\alpha\sum_i \delta_{ms}\delta_{nt}(\mathbf{W}_{i,m}\mathbf{W}_{i,n})^+\frac{\widetilde{\mathbf{S}}_{m,n}}{\mathbf{S}_{m,n}^2}\\
&+2\gamma\delta_{ms}\delta_{nt} + 2\gamma\delta_{ms}\delta_{nt}(\mathbf{S}_0)_{m,n}\frac{\widetilde{\mathbf{S}}_{m,n}}{\mathbf{S}_{m,n}^2}\\
\geq&0.
\end{aligned} \tag{4.28}$$

Therefore, $\nabla_{\mathbf{S}}^2 Z(\mathbf{S},\widetilde{\mathbf{S}})$ is diagonal with positive entries. Thus $\nabla_{\mathbf{S}}^2 Z(\mathbf{S},\widetilde{\mathbf{S}})$ is positively defined, namely, $Z(\mathbf{S},\widetilde{\mathbf{S}})$ is convex, which concludes **Property 3**.

To solve for $\mathbf{S}$, we set $\nabla_{\mathbf{S}}Z(\mathbf{S},\tilde{\mathbf{S}}) = \mathbf{0}$, and get the following formula for all m and n.

$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{S}_{m,n}}Z(\mathbf{S},\widetilde{\mathbf{S}})\\
&= \alpha\sum_i \mathbf{W}_{i,m}^2\frac{\mathbf{S}_{m,n}}{\widetilde{\mathbf{S}}_{m,n}} + \alpha\sum_i(\mathbf{W}_{i,m}\mathbf{W}_{i,n})^-\frac{\mathbf{S}_{m,n}}{\widetilde{\mathbf{S}}_{m,n}}\\
&\quad -\alpha\sum_i(\mathbf{W}_{i,m}\mathbf{W}_{i,n})^+\frac{\widetilde{\mathbf{S}}_{m,n}}{\mathbf{S}_{m,n}} + 2\gamma\mathbf{S}_{m,n} - 2\gamma(\mathbf{S}_0)_{m,n}\frac{\widetilde{\mathbf{S}}_{m,n}}{\mathbf{S}_{m,n}}\\
&= 0.
\end{aligned} \tag{4.29}$$

After sorting the equation, we have

$$\mathbf{S}_{m,n} = \widetilde{\mathbf{S}}_{m,n}\cdot\frac{\alpha\sum_i(\mathbf{W}_{i,m}\mathbf{W}_{i,n})^+ + 2\gamma(\mathbf{S}_0)_{m,n}}{2\gamma\widetilde{\mathbf{S}}_{m,n} + \alpha\sum_i(\mathbf{W}_{i,m}\mathbf{W}_{i,n})^- + \alpha\sum_i \mathbf{W}_{i,m}^2}. \tag{4.30}$$

That is equivalent to the formula (4.21), which is consistent with the updating formula derived from the KKT condition aforementioned. □

Theorem 4.4. *Updating* $\mathbf{S}$ *using Eq.* (4.10) *will monotonically decrease the value of the objective in Eq.* (4.7)*, the objective is invariant if and only if* $\mathbf{S}$ *is at a stationary point.*

Proof: By Lemma 4.2 and Theorem 4.3, for each subsequent iteration of updating $\mathbf{S}$, we have $L((\mathbf{S})^0) = Z((\mathbf{S})^0, (\mathbf{S})^0) \geq Z((\mathbf{S})^1, (\mathbf{S})^0) \geq Z((\mathbf{S})^1, (\mathbf{S})^1) = L((\mathbf{S})^1) \geq ... \geq L((\mathbf{S})^{Iter})$. Thus $L(\mathbf{S})$ monotonically decreases. Since the objective function Eq. (4.7) is obviously bounded below, the correctness of Theorem 4.1 is proved. Theorem 4.2 can be proved similarly. □

In addition to Theorem 4.4, since the computation of $\mathbf{L}$ in the first step decreases the value of the objective in Eq. (4.7), and the coordinate descent algorithm for updating $\mathbf{W}$ in the third step also monotonically decreases the value of the objective, the algorithm is guaranteed to converge.

4.4 Generalized Graph-regularized Dual Lasso

In this section, we extend our model to incorporate additional prior knowledge such as SNP locations and biological pathways. If the physical locations of two SNPs are close or two genes belong to the same pathway, they are likely to have interactions. Such information can be integrated to help refine the prior networks.

Continue with our example in Figure 4.1. If SNPs ③ and ④ affect the same set of genes (Ⓑ and Ⓓ), and at the same time, they are close to each other, then it is likely there exists interaction between ③ and ④.

Formally, we would like to solve the following optimization problem

$$\begin{aligned}\min_{\mathbf{W},\boldsymbol{\mu},\mathbf{L},\mathbf{S}\geq 0,\mathbf{G}\geq 0} & \frac{1}{2}||\mathbf{W}\mathbf{X} - \mathbf{Z} - \mu\mathbf{1} - \mathbf{L}||_F^2 + \eta||\mathbf{W}||_1 + \lambda||\mathbf{L}||_* \\ & + \alpha\sum_{i,j}\mathcal{D}(\mathbf{w}_{*i}, \mathbf{w}_{*j})\mathbf{S}_{i,j} + \beta\sum_{i,j}\mathcal{D}(\mathbf{w}_{i*}, \mathbf{w}_{j*})\mathbf{G}_{i,j}\end{aligned} \tag{4.31}$$

Here $\mathcal{D}(\cdot,\cdot)$ is a non-negative distance measure. Note that the Euclidean distance is used in previous sections. $\mathbf{S}$ and $\mathbf{G}$ are initially given by inputs $\mathbf{S}_0$ and $\mathbf{G}_0$. We refer to this generalized model as the Generalized Graph-regularized Dual Lasso (GGD-Lasso). GGD-Lasso executes the following two steps iteratively until the termination condition is met: 1) update $\mathbf{W}$ while fixing $\mathbf{S}$ and $\mathbf{G}$; 2) update $\mathbf{S}$ and $\mathbf{G}$ according to $\mathbf{W}$, while guarantee that both $\sum_{i,j}\mathcal{D}(\mathbf{w}_{*i}, \mathbf{w}_{*j})\mathbf{S}_{i,j}$ and $\sum_{i,j}\mathcal{D}(\mathbf{w}_{i*}, \mathbf{w}_{j*})\mathbf{G}_{i,j}$ decrease.

Algorithm 4: Generalized Graph-regularized Dual Lasso (GGD-Lasso)

Input: $\mathbf{X} = \{\mathbf{x}_d\} \in \mathbb{R}^{K\times D}$, $\mathbf{Z} = \{\mathbf{z}_d\} \in \mathbb{R}^{N\times D}$, $\mathbf{S}_0 \in \mathbb{R}^{K\times K}$, $\mathbf{G}_0 \in \mathbb{R}^{N\times N}$, Pathway information, SNPs location information, η,α,β,κ_1,κ_2

Output: $\mathbf{W}$,$\boldsymbol{\mu}$,$\mathbf{S}$,$\mathbf{G}$,$\mathbf{L}$

begin
  $\mathbf{S} \leftarrow \mathbf{S}_0, \mathbf{G} \leftarrow \mathbf{G}_0$;
  $updateS \leftarrow 0, updateG \leftarrow 0$;
  repeat
    Update $\mathbf{W}$, $\boldsymbol{\mu}$ and $\mathbf{L}$ that minimize the objective function (4.6) using $\mathbf{S}$ and $\mathbf{G}$;
    Put all pairs (i, j) of columns of $\mathbf{W}$ in order of distance;
    $\mathcal{P}_0 \leftarrow \emptyset$;
    $\mathcal{P}_1 \leftarrow \emptyset$;
    Select κ_1 pairs (i_S, j_S) with smallest $\mathcal{D}(\mathbf{W}_{*i_S}, \mathbf{W}_{*j_S})$ to the set $\mathcal{P}_0$;
    $\mathcal{P}_0 \leftarrow$ pairs in $\mathcal{P}_0$ that satisfy $\mathbf{S}_{i_S,j_S} = 0$ and the distance between the i_S-th SNP and j_S-th SNP is less than 500bp;
    Select κ_1 pairs (i'_S, j'_S) with largest $\mathcal{D}(\mathbf{W}_{*i'_S}, \mathbf{W}_{*j'_S})$ to the set $\mathcal{P}_1$;
    $\mathcal{P}_1 \leftarrow$ pairs in $\mathcal{P}_1$ that satisfy $\mathbf{S}_{i'_S,j'_S} = 1$ and the distance between the i'_S-th SNP and j'_S-th SNP is larger than 500bp;
    $updateS \leftarrow \min(|\mathcal{P}_0|, |\mathcal{P}_1|)$;
    Choose $updateS$ pairs (i_S, j_S) in $\mathcal{P}_0$ and set $\mathbf{S}_{i_S,j_S}$ to 1;
    Choose $updateS$ pairs (i'_S, j'_S) in $\mathcal{P}_1$ and set $\mathbf{S}_{i'_S,j'_S}$ to 0;
    Put all pairs (i, j) of rows of $\mathbf{W}$ in order of distance;
    $\mathcal{Q}_1 \leftarrow \emptyset$;
    $\mathcal{Q}_2 \leftarrow \emptyset$;
    Select κ_2 pairs (i_G, j_G) with smallest $\mathcal{D}(\mathbf{W}_{i_G*}, \mathbf{W}_{j_G*})$ to the set $\mathcal{Q}_0$;
    $\mathcal{Q}_0 \leftarrow$ pairs in $\mathcal{Q}_0$ that satisfy $\mathbf{G}_{i_G,j_G} = 0$ and the i_G-th gene and j_G-th gene belong to the same pathway;
    Select κ_2 pairs (i'_G, j'_G) with largest $\mathcal{D}(\mathbf{W}_{i'_G*}, \mathbf{W}_{j'_G*})$ to the set $\mathcal{Q}_1$;
    $\mathcal{Q}_1 \leftarrow$ pairs in $\mathcal{Q}_1$ that satisfy $\mathbf{G}_{i'_G,j'_G} = 1$ and the i'_G-th gene and j'_G-th gene do not belong to the same pathway;
    $updateG \leftarrow \min(|\mathcal{Q}_0|, |\mathcal{Q}_1|)$;
    Choose $updateG$ pairs (i_G, j_G) in $\mathcal{Q}_0$ and set $\mathbf{G}_{i_G,j_G}$ to 1;
    Choose $updateG$ pairs (i'_G, j'_G) in $\mathcal{Q}_1$ and set $\mathbf{G}_{i'_G,j'_G}$ to 0;
  until $updateS = 0$ **and** $updateG = 0$;
end

These two steps are based on the aforementioned duality between learning $\mathbf{W}$ and refining $\mathbf{S}$ and $\mathbf{G}$. The detailed algorithm is provided in Algorithm 4. Next, we illustrate the updating process assuming that $\mathbf{S}$ and $\mathbf{G}$ are unweighted graphs. It can be easily extended to weighted graphs.

Step 1 can be done by using the coordinate descent algorithm. In Step 2, to guarantee that both $\sum_{i,j} \mathcal{D}(\mathbf{w}_{*i}, \mathbf{w}_{*j})\mathbf{S}_{i,j}$ and $\sum_{i,j} \mathcal{D}(\mathbf{w}_{i*}, \mathbf{w}_{j*})\mathbf{G}_{i,j}$ decrease, we can maintain a fixed number of 1's in $\mathbf{S}$ and $\mathbf{G}$. Taking $\mathbf{G}$ as an example, once $\mathbf{G}_{i,j}$ is selected to change from 0 to 1, another element $\mathbf{G}_{i',j'}$ with $\mathcal{D}(\mathbf{w}_{i*}, \mathbf{w}_{j*}) < \mathcal{D}(\mathbf{w}_{i'*}, \mathbf{w}_{j'*})$ should be changed from 1 to 0.

The selection of (i, j) and (i', j') is based on the ranking of $\mathcal{D}(\mathbf{w}_{i*}, \mathbf{w}_{j*})$ $(1 \leq i < j \leq N)$. Specifically, we examine κ pairs with the smallest distances. Among them, we pick those having no edges in $\mathbf{G}$. Let $\mathcal{P}_0$ be this set of pairs. Accordingly, we examine κ pairs with the largest distances. Among these pairs, we pick up only those having an edge in $\mathbf{G}$. Let $\mathcal{P}_1$ be this set of pairs. The elements of $\mathbf{G}$ corresponding to pairs in $\mathcal{P}_0$ are candidates for updating from 0 to 1, since these pairs of genes are associated with similar SNPs. Similarly, elements of $\mathbf{G}$ corresponding to pairs in $\mathcal{P}_1$ are candidates for updating from 1 to 0.

In this process, the prior knowledge of gene pathways can be easily incorporated to better refine $\mathbf{G}$. For instance, we can further require that only the gene pairs in $\mathcal{P}_0$ belonging to the same pathway are eligible for updating, and only the gene pairs in $\mathcal{P}_1$ belonging to different pathways are eligible for updating. We denote the set of gene pairs eligible for updating by $\mathcal{P}_0'$ and $\mathcal{P}_1'$ respectively. Then, we choose $\min(|\mathcal{P}_0'|, |\mathcal{P}_1'|)$ pairs in set $\mathcal{P}_0'$ with smallest $\mathcal{D}(\mathbf{w}_{i*}, \mathbf{w}_{j*})$ $((i, j) \in \mathcal{P}_0')$ and update $\mathbf{G}_{i,j}$ from 0 to 1. Similarly, we choose $\min(|\mathcal{P}_0'|, |\mathcal{P}_1'|)$ pairs in set $\mathcal{P}_1'$ with largest $\mathcal{D}(\mathbf{w}_{i'*}, \mathbf{w}_{j'*})$ $((i', j') \in \mathcal{P}_1')$ and update $\mathbf{G}_{i',j'}$ from 1 to 0.

Obviously, all $\mathcal{D}(\mathbf{w}_{i*}, \mathbf{w}_{j*})$'s are smaller than $\mathcal{D}(\mathbf{w}_{i'*}, \mathbf{w}_{j'*})$ if $\kappa < \frac{N(N-1)}{4}$. Therefore, $\sum_{i,j} \mathcal{D}(\mathbf{w}_{i*}, \mathbf{w}_{j*})\mathbf{G}_{i,j}$ is guaranteed to decrease. The updating process for $\mathbf{S}$ is similar except that we compare columns rather than rows of $\mathbf{W}$ and use SNP locations rather than pathway information for evaluating the eligibility for updating. The updating process ends when no such pairs can be found so that switching their values will result in a decrease of the objective function.

The convergence of GGD-Lasso can be observed as follows. The decrease of the objective function value in the first step is straightforward since we minimize it using coordinate descent. In the second step, the change of the objective function value is given by

$$-\alpha\mathcal{D}(\mathbf{w}_{*i_S}, \mathbf{w}_{*j_S}) + \alpha\mathcal{D}(\mathbf{w}_{*i'_S}, \mathbf{w}_{*j'_S}) - \beta\mathcal{D}(\mathbf{w}_{i_G*}, \mathbf{w}_{j_G*}) + \beta\mathcal{D}(\mathbf{w}_{i'_G*}, \mathbf{w}_{j'_G*}) \tag{4.32}$$

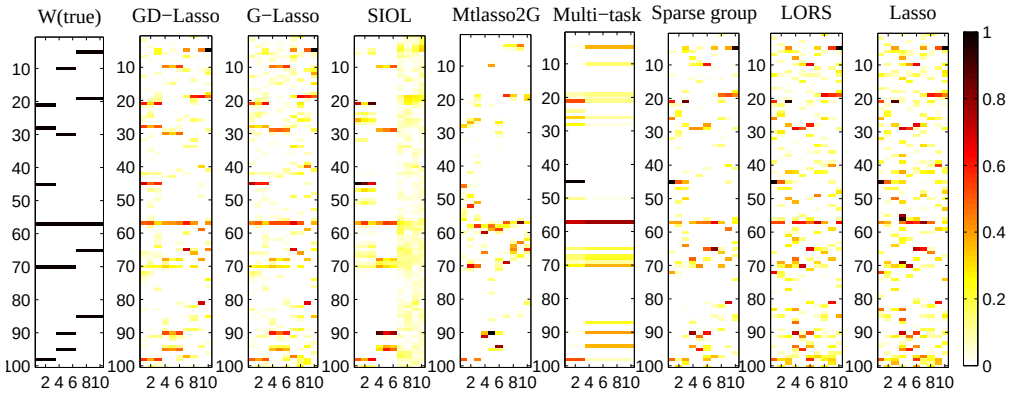


Figure 4.2: Ground truth of $\mathbf{W}$ and that estimated by different methods.

which is always negative. Thus, in each iteration, the objective function value decreases. Since the objective function is non-negative, the process eventually converges.

Theorem 4.5. *GGD-Lasso converges to the global optimum if both* $\sum_{i,j} \mathcal{D}(\mathbf{w}_{i*}, \mathbf{w}_{j*})$ *and* $\sum_{i,j} \mathcal{D}(\mathbf{w}_{*i}, \mathbf{w}_{*j})$ *are convex to* $\mathbf{W}$.

Proof: The last two terms in Eq. (4.31) are linear with respect to $\mathbf{S}$ and $\mathbf{G}$, and convex to $\mathbf{W}$ according to the conditions listed. Thus the objective function is convex over all variables. A convergent result to the global optimum can be guaranteed. □

4.5 Experimental Results

In this section, we perform extensive experiments to evaluate the performance of the proposed methods. We use both simulated datasets and real yeast eQTL dataset (Rachel B. Brem and Kruglyak, 2005). For comparison, we select several state-of-the-art methods, including SIOL (Lee and Xing, 2012), two graph guided multi-task lasso (mtlasso2G) (Chen et al., 2012), sparse group Lasso (Biganzoli et al., 2006), sparse multi-task Lasso (Biganzoli et al., 2006), LORS (Yang et al., 2013) and Lasso (Tibshirani, 1996). For all the methods, the tuning parameters were learned using cross validation.

4.5.1 Simulation Study

We first evaluate the performance of the selected methods using simulation study. Note that GGD-Lasso requires additional prior knowledge and will be evaluated using real dataset.

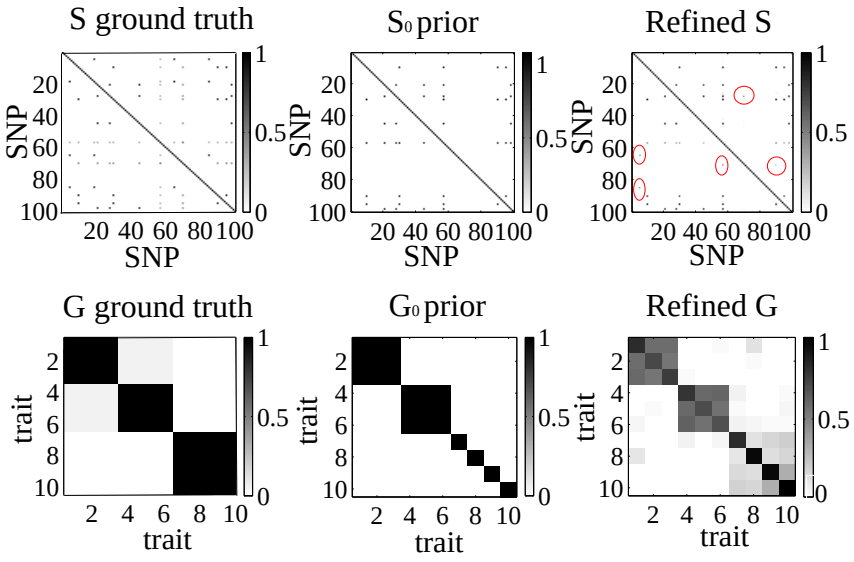


Figure 4.3: The ground truth networks, prior partial networks, and the refined networks

We adopt the same setup for the simulation study as that in (Lee and Xing, 2012; Yang et al., 2013) and generate synthetic datasets as follows. 100 SNPs are randomly selected from the yeast eQTL dataset (112 samples) (Rachel B. Brem and Kruglyak, 2005). 10 gene expression profiles are generated by $\mathbf{Z}_{j*} = \mathbf{W}_{j*}\mathbf{X} + \Xi_{j*} + \mathbf{E}_{j*}$ $(1 \leq j \leq 10)$, where $\mathbf{E}_{j*} \sim \mathcal{N}(0, \sigma^2 I)$ $(\sigma = 1)$ denotes Gaussian noise. Ξ_{j*} is used to model non-genetic effects, which are drawn from $\mathcal{N}(\mathbf{0}, \tau\Sigma)$, where $\tau = 0.1$. Σ is generated by $\mathbf{M}\mathbf{M}^{\mathrm{T}}$, where $\mathbf{M} \in \mathbb{R}^{D\times C}$ and $\mathbf{M}_{ij} \sim \mathcal{N}(0, 1)$. C is the number of hidden factors and is set to 10 by default. The association matrix $\mathbf{W}$ is generated as follows. Three sets of randomly selected four SNPs are associated with three gene clusters (1-3), (4-6), (7-10) respectively. In addition, one SNP is associated with two gene clusters (1-3) and (4-6), and one SNP is associated with all genes. The association strength is set to 1 for all selected SNPs. The clustering structures among SNPs and genes serve as the *ground truth* of the prior network knowledge. Only two of the three SNP (gene) clusters are used in $\mathbf{W}$ to simulate incomplete prior knowledge.

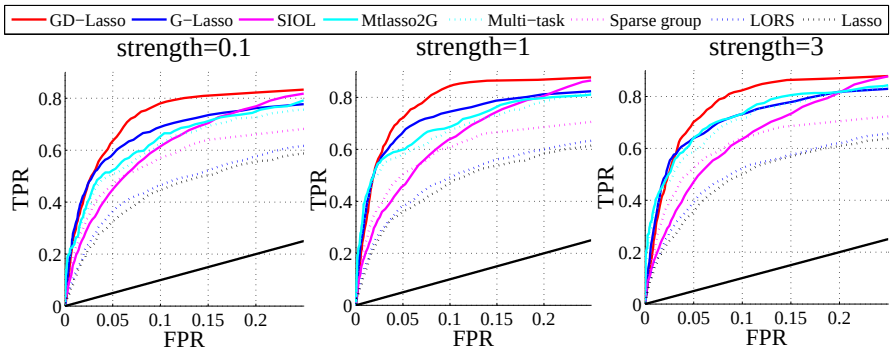
Figure 4.2 shows the estimated $\mathbf{W}$ matrix by various methods. The x-axis represents traits (1-10) and y-axis represents SNPs (1-100). From the figure, we can see that GD-Lasso is more effective than G-Lasso. This is because the dual refinement enables a more robust model.

G-Lasso outperforms SIOL and mtlasso2G, indicating that the graph regularizer provides a smoother regularization than the hard clustering based penalty. In addition, SIOL and mtlasso2G do not consider confounding factors. SIOL and mtlasso2G outperform multi-task Lasso and sparse group Lasso since it uses both SNP and gene grouping information, while multi-task Lasso and sparse group Lasso only use one of them. We also observe that all methods utilizing prior grouping knowledge outperform LORS and Lasso which cannot incorporate prior knowledge. LORS outperforms Lasso since it considers the confounding factors.

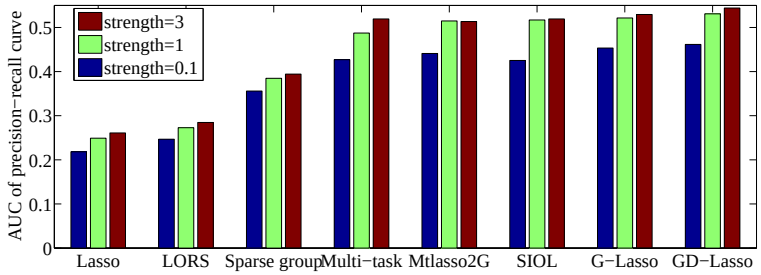
The ground truth networks, prior networks, and GD-Lasso refined networks are shown in Figure 4.3. Note that only a portion of the ground truth networks are used as prior networks. In particular, the information related to gene cluster (7-10) is missing in the prior networks. We observe that the refined matrix $\mathbf{G}$ well captures the missing grouping information of gene cluster (7-10). Similarly, many missing pairwise relationships in $\mathbf{S}$ are recovered in the refined matrix (points in red ellipses).

Using 50 simulated datasets with different gaussian noise ($\sigma^2 = 1$ and $\sigma^2 = 5$), we compare the proposed methods with alternative state-of-the-art approaches. For each setting, we use 30 samples for test and 82 samples for training. We report the average result from 50 realizations. Figure 4.4 shows the ROC curves of TPR-FPR for performance comparison, together with the areas under the precision-recall curve (AUCs) (Chen et al., 2012). The association strengths between SNPs and genes are set to be 0.1, 1 and 3 respectively. It is clear that GD-Lasso outperforms all alternative methods by effectively using and refining the prior network knowledge. We also computed test errors. On average, GD-Lasso achieved the best test error rate of 0.9122, and the order of the other methods in terms of the test errors is: G-Lasso (0.9276), SIOL (0.9485), Mtlasso2G (0.9521), Multi-task Lasso (0.9723), Sparse group Lasso (0.9814), LORS (1.0429) and Lasso (1.2153).

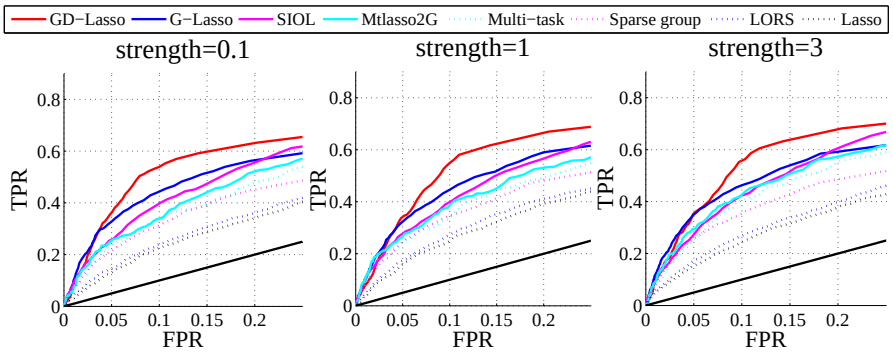
To evaluate the effectiveness of dual refinement, we compare GD-Lasso and G-Lasso since the only difference between these two methods is whether the prior networks are refined during the optimization process. We add noises to the prior networks by randomly shuffling the



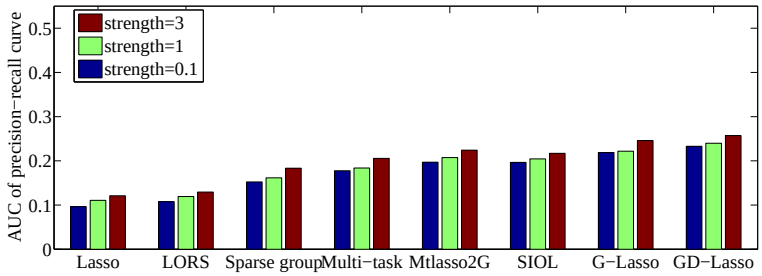
(a) variance of errors ($\sigma^2 = 1$)



(b) AUC of precision-recall curve ($\sigma^2 = 1$)



(c) variance of errors ($\sigma^2 = 5$)



(d) AUC of precision-recall curve ($\sigma^2 = 5$)

Figure 4.4: The ROC curve and AUCs of different methods.

elements in them. Furthermore, we use the signal-to-noise ratio defined as $SNR = \sqrt{\frac{Var(\mathbf{WX})}{Var(\Xi+\mathbf{E})}}$ (Yang et al., 2013) to measure the noise ratio in the eQTL datasets. Here, we fix $C = 10, \tau = 0.1$, and use different σ's to control SNR.

Figure 4.5 shows the results for different SNRs. For a fixed SNR, we vary the percentage of noises in the prior networks and compare the performance of selected methods. From the results, we can see that G-Lasso is more sensitive to noises in the prior networks than GD-Lasso is. Moreover, when the SNR is low, the advantage of GD-Lasso is more prominent. These results indicate using dual refinement can dramatically improve the accuracy of the identified associations.

4.5.2 Yeast eQTL Study

We apply the proposed methods to a yeast (Saccharomyces cerevisiae) eQTL dataset of 112 yeast segregants generated from a cross of two inbred strains (Rachel B. Brem and Kruglyak, 2005). The dataset originally includes expression profiles of 6229 gene expression traits and genotype profiles of 2956 SNPs. After removing SNPs with more than 10% missing values and merging consecutive SNPs high linkage disequilibrium, we get 1017 SNPs with unique genotypes (Huang et al., 2009a). 4474 expression profiles are selected after removing the ones with missing values. The genetic interaction network is generated as in (Lee and Xing, 2012). We use the PPI network downloaded from BioGRID (http://thebiogrid.org/) to represent the prior network among genes. It takes around 1 day for GGD-Lasso, and around 10 hours for GD-Lasso to run into completion.

4.5.2.1 cis and trans Enrichment Analysis

We follow the standard *cis*-enrichment analysis (Listgarten et al., 2010) to compare the performance of two competing models. The intuition behind *cis*-enrichment analysis is that more *cis*-acting SNPs are expected than *trans*-acting SNPs. A two-step procedure is used in the *cis*-enrichment analysis (Listgarten et al., 2010): (1) for each model, we apply a one-tailed Mann-Whitney test on each SNP to test the null hypothesis that the model ranks its *cis* hypotheses

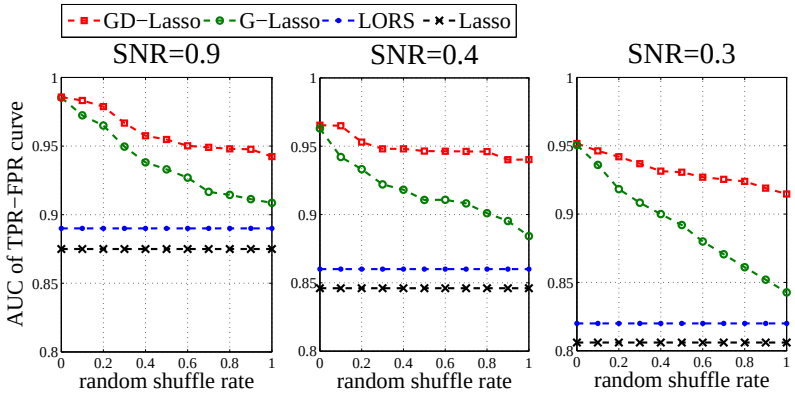


Figure 4.5: The AUCs of the TPR-FPR curve of different methods.

		GD-Lasso	G-Lasso	SIOL	Mtlasso2G	Multi-task	Sparse group	LORS	Lasso
cis-enrichment	GGD-Lasso	0.0003	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
	GD-Lasso	-	0.0009	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
	G-Lasso	-	-	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
	SIOL	-	-	-	0.1213	0.0331	0.0173	< 0.0001	< 0.0001
	Mtlasso2G	-	-	-	-	0.0487	0.0132	< 0.0001	< 0.0001
	Multi-task	-	-	-	-	-	0.4563	0.4132	< 0.0001
	Sparse group	-	-	-	-	-	-	0.4375	< 0.0001
	LORS	-	-	-	-	-	-	-	< 0.0001
trans-enrichment	GGD-Lasso	0.0881	0.0119	0.0102	0.0063	0.0006	0.0003	< 0.0001	< 0.0001
	GD-Lasso	-	0.0481	0.0253	0.0211	0.0176	0.0004	< 0.0001	< 0.0001
	G-Lasso	-	-	0.0312	0.0253	0.0183	0.0007	< 0.0001	< 0.0001
	SIOL	-	-	-	0.1976	0.1053	0.0044	0.0005	< 0.0001
	Mtlasso2G	-	-	-	-	0.1785	0.0061	0.0009	< 0.0001
	Multi-task	-	-	-	-	-	0.0235	0.0042	0.0011
	Sparse group	-	-	-	-	-	-	0.0075	0.0041
	LORS	-	-	-	-	-	-	-	0.2059

Table 4.2: Pairwise comparison of different models using *cis*- and *trans*- enrichment.

no better than its *trans* hypotheses, (2) for each pair of models compared, we perform a two-tailed paired Wilcoxon sign-rank test on the *p*-values obtained from the previous step. The null hypothesis is that the median difference of the *p*-values in the Mann-Whitney test for each SNP is zero. The *trans*-enrichment is implemented using a similar strategy (Yvert et al., 2003), in which genes regulated by transcription factors (obtained from http://www.yeastract.com/download.php) are used as *trans*-acting signals.

In addition to the methods evaluated in the simulation study, GGD-Lasso is also evaluated here (with $\kappa = 100000$,$\eta = 5, \lambda = 8, \alpha = 15, \beta = 1$). For GD-Lasso, $\eta = 5, \lambda = 8, \alpha = 15, \beta = 1, \gamma = 15, \rho = 1$. The Euclidean distance is used as the distance metric. We rank pairs of SNPs and genes according to the learned $\mathbf{W}$. $\mathbf{S}$ is refined if the locations of the two SNPs are less than 500 bp. $\mathbf{G}$ is refined if the two genes are in the same pathway. The

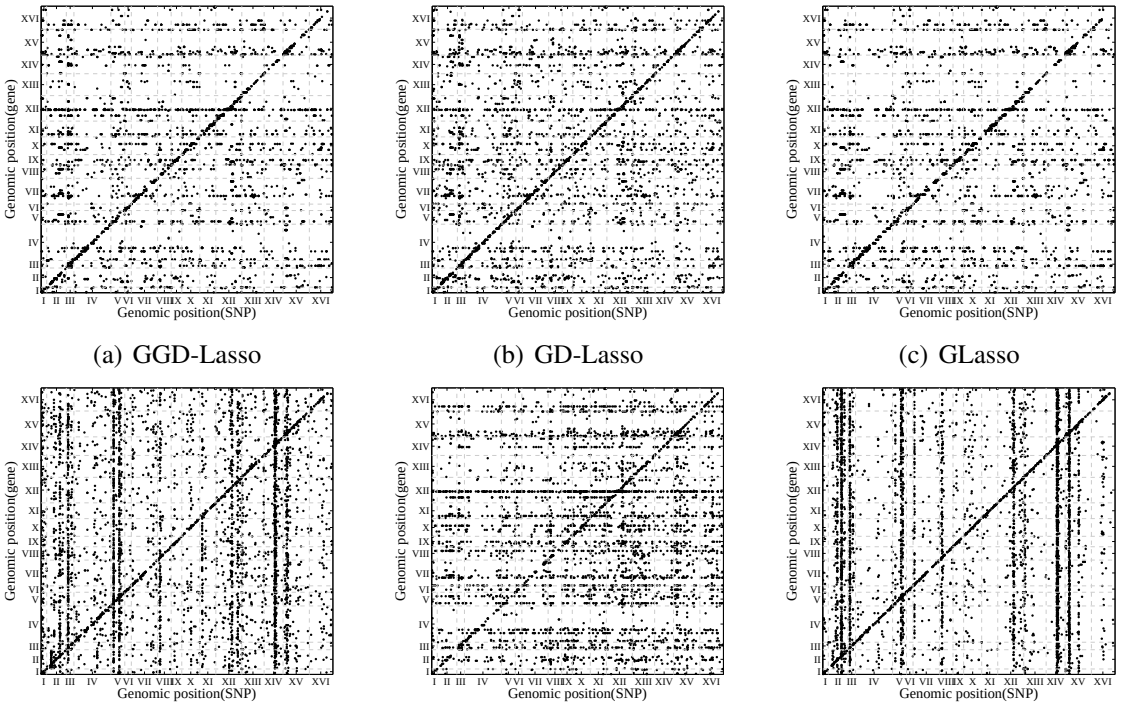
pathway information is downloaded from Saccharomyces Genome Database (SGD (http://www.yeastgenome.org/)).

The results of pairwise comparison of selected models are shown in Table 4.2. In this table, a p-value shows how significant a method on the left column outperforms a method in the top row in terms of *cis* and *trans* enrichments. We observe that the proposed GGD-Lasso and GD-Lasso have significantly better enrichment scores than the other models. By incorporating genomic location and pathway information, GGD-Lasso performs better than GD-Lasso with p-value less than 0.0001. The effectiveness of the dual refinement on prior graphs is demonstrated by GD-Lasso's better performance over G-Lasso. Note that the performance ranking of these models is consistent with that in the simulation study.

The top-1000 significant associations given by GGD-Lasso, GD-Lasso and G-Lasso are shown in Figure 4.7. We can see that GGD-Lasso and GD-Lasso have stronger cis-regulatory signals than G-Lasso does. In total, these methods each detected about 6000 associations according to non-zero $\mathbf{W}$ values. We estimate FDR using 50 permutations as proposed in (Yang et al., 2013). With FDR ≤ 0.01, GGD-Lasso obtains about 4500 significant associations. The plots of all identified significant associations for different methods are given in Figure 4.6.

4.5.2.2 Refinement of the Prior Networks

To investigate to what extent GGD-Lasso is able to refine the prior networks and study the effect of different parameter settings on κ, we intentionally change 75% of the elements in the original prior PPI network and genetic interaction network to random noises. We feed the new networks to GGD-Lasso and evaluate the refined networks. The results are shown in Figure 4.8. We can see that for both PPI and genetic interaction networks, many elements are recovered by GGD-Lasso. This demonstrates the effectiveness of GGD-Lasso. Moreover, when the number of SNP (gene) pairs (κ) examined for updating reaches 100,000, both PPI and genetic iteration networks are well refined.



(a) GGD-Lasso (b) GD-Lasso (c) GLasso

(d) SIOL (e) LORS (f) Lasso

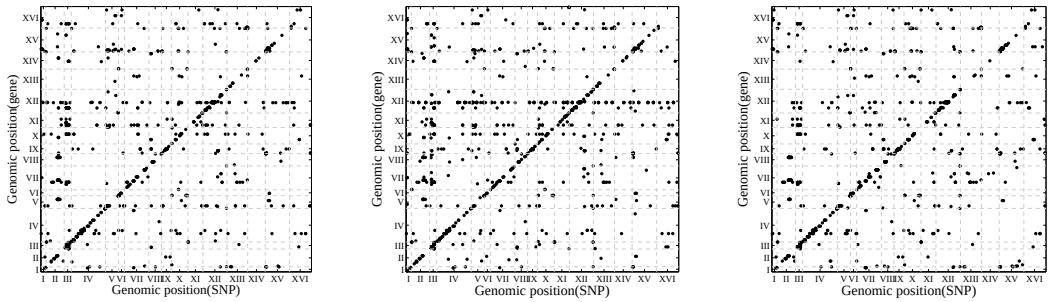
Figure 4.6: The plot of linkage peaks in the study by different methods.

ID	size[a]	Loci[b]	GO[c]	Hits[d]	GD-Lasso (all)[e]	GD-Lasso (hits)[f]	G-Lasso (all)[g]	G-Lasso (hits)[h]	SIOL (all)[i]	SIOL (hits)[j]	LORS (all)[k]	LORS (hits)[l]
1	31	XII:1056097	(1)***	7	31	7	32	7	8	6	31	7
2	28	III:81832..92391	(2)**	5	29	5	28	5	58	5	22	4
3	28	XII:1056103	(1)***	7	29	6	28	6	1	1	2	0
4	27	III:79091	(2)***	6	29	6	28	6	28	7	10	2
5	27	III:175799..177850	(3)*	3	26	3	23	3	9	2	18	4
6	*27*	*XII:1059925..1059930*	*(1)****	*7*	*27*	*7*	*27*	*7*	*0*	*0*	*5*	*1*
7	25	III:105042	(2)***	6	23	6	25	6	5	3	19	4
8	23	III:201166..201167	(3)***	3	23	3	22	3	13	2	23	3
9	22	XII:1054278..1054302	(1)***	7	26	7	24	7	24	5	12	4
10	21	III:100213	(2)**	5	23	5	23	5	5	3	5	1
11	20	III:209932	(3)*	3	21	3	19	3	16	4	15	4
12	**20**	**XII:659357..662627**	**(4)***	**4**	**19**	**4**	**3**	**0**	**37**	**9**	**36**	**6**
13	19	III:210748..210748	(5)*	4	24	4	18	4	2	3	11	4
14	19	VIII:111679..111680	(6)*	3	20	3	19	3	3	3	12	2
15	19	VIII:111682..111690	(7)**	5	21	5	20	5	57	6	22	3
Total hits				75		74		70		59		49

Table 4.3: Summary of the top-15 hotspots detected by GGD-Lasso.

	GGD-Lasso	GD-Lasso	G-Lasso	SIOL	LORS
#hotspots significantly enriched (top 15 hotposts)	**15**	**14**	**13**	**10**	**9**
#total reported hotspots (size $>$ 10)	65	82	96	89	64
#hotspots significantly enriched	45	56	61	53	41
ratio of significantly enriched hotspots	70%	68%	64%	60%	56%

Table 4.4: Hotspots detected by different methods



(a) GGD-Lasso (b) GD-Lasso (c) G-Lasso

Figure 4.7: The top-1000 significant associations identified by different methods.

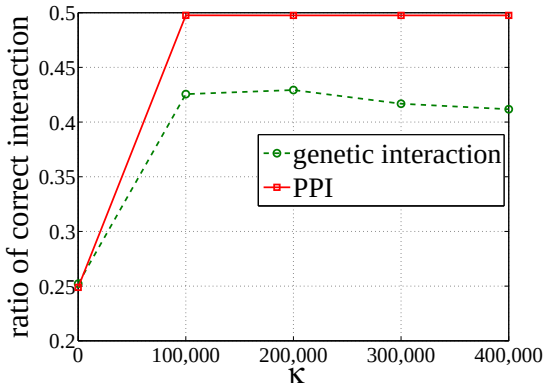


Figure 4.8: Ratio of correct interactions refined when varying κ.

4.5.2.3 Hotspots Analysis

In this subsection, we study whether GGD-Lasso can help detect more biologically relevant associations than the alternatives. Specifically, we examine the hotspots which affect more than 10 gene traits (Lee and Xing, 2012). The top 15 hotspots detected by GGD-Lasso are listed in Table 4.3. The top-15 hotspots detected by other methods are included in Table 4.5, tab:hotspotscompareGL, tab:hotspotscompareSIOL, and tab:hotspotscompareLORS. From Table 4.3, we observe that for all hotspots, the associated genes are enriched with at least one GO category. Note that GGD-Lasso and GD-Lasso detect one hotspot (12), which cannot be detected by G-Lasso. They also detect one hotspot (6), which cannot be detected by SIOL. The number of hotspots that are significant enriched is listed in Table 4.4. From the table, we can see that

chr	start	end	size	GO category	adjusted p-value
XII	1056097	1056097	31	telomere maintenance via recombination	4.72498E-9
III	79091	79091	29	branched chain family amino acid biosynthetic process	1.59139E-8
III	81832	92391	29	branched chain family amino acid biosynthetic process	2.62475E-05
XII	1056103	1056103	29	telomere maintenance via recombination	1.90447E-4
XII	1059925	1059930	27	telomere maintenance via recombination	2.6379E-8
III	175799	177850	26	regulation of mating-type specific transcription, DNA-dependent	2.07885E-03
XII	1054278	1054302	26	telomere maintenance via recombination	2.30417E-9
III	210748	210748	24	regulation of mating-type specific transcription, DNA-dependent	1.61983E-04
III	100213	100213	23	branched chain family amino acid biosynthetic process	7.4936E-3
III	105042	105042	23	branched chain family amino acid biosynthetic process	3.8412E-8
III	201166	201167	23	regulation of mating-type specific transcription, DNA-dependent	0.001998002
III	209932	209932	21	regulation of mating-type specific transcription, DNA-dependent	1.06592E-03
VIII	111682	111690	21	response to pheromone	7.04262E-04
V	**395442**	**395442**	**20**	**SRP-dependent cotranslational protein targeting to membrane, translocation**	**0.100899101**
VIII	111679	111680	20	cytogamy	0.001998002

Table 4.5: Summary of the top 15 detected hotspots by GD-Lasso

chr	start	end	size	GO category	adjusted p-value
XII	1056097	1056097	32	telomere maintenance via recombination	5.52E-08
III	79091	79091	28	branched chain family amino acid biosynthetic process	1.28E-07
III	81832	92391	28	branched chain family amino acid biosynthetic process	2.17E-05
XII	1056103	1056103	28	telomere maintenance via recombination	1.52E-06
XII	1059925	1059930	27	telomere maintenance via recombination	2.64E-08
III	105042	105042	25	branched chain family amino acid biosynthetic process	6.35E-08
XII	1054278	1054302	24	telomere maintenance via recombination	1.78E-08
III	100213	100213	23	branched chain family amino acid biosynthetic process	7.49E-06
III	175799	177850	23	regulation of mating-type specific transcription, DNA-dependent	0.001998002
XII	674651	674651	23	sterol biosynthetic process	3.56E-04
III	201166	201167	22	regulation of mating-type specific transcription, DNA-dependent	1.23E-03
V	**395442**	**395442**	**21**	**SRP-dependent cotranslational protein targeting to membrane, translocation**	**0.086913087**
I	**51324**	**52943**	**20**	**fatty acid metabolic process**	**0.281718282**
VIII	111682	111690	20	response to pheromone	5.39E-04
III	209932	209932	19	regulation of mating-type specific transcription, DNA-dependent	7.77E-03

Table 4.6: Summary of the top 15 detected hotspots by G-Lasso

chr	start	end	size	GO category	adjust p-value
XIV	449639	449639	339	mitochondrial translation	2.92E-07
V	109310	117705	240	translation	2.39E-08
V	350744	350744	183	translation	1.32E-07
XV	**154177**	**154309**	**94**	**replicative cell aging**	**0.264735265**
XII	899898	927421	81	translation	1.45E-06
XIV	486861	486861	81	mitochondrial translation	1.49E-06
II	**548401**	**548401**	**78**	**endonucleolytic cleavage in ITS1 to separate SSU-rRNA from 5.8S rRNA and LSU-rRNA**	**0.030969031**
III	75021	75021	78	cellular amino acid biosynthetic process	1.35E-06
XIV	**502316**	**502496**	**76**	**mitochondrial genome maintenance**	**0.824175824**
XII	674651	674651	73	electron transport chain	8.52E-04
III	81832	92391	58	branched chain family amino acid biosynthetic process	9.78E-05
VIII	111682	111690	57	response to pheromone	5.15E-03
XV	**202370**	**210839**	**49**	**vesicle-mediated transport**	**0.592407592**
XIII	27644	28334	45	dephosphorylation	0.007992008
XV	**170945**	**180961**	**44**	**(1->6)-beta-D-glucan biosynthetic process**	**0.132867133**

Table 4.7: Summary of the top 15 detected hotspots by SIOL

GGD-Lasso slightly outperforms GD-Lasso since it incorporates the location of SNPs and gene pathway information.

4.6 Conclusion

As a promising tool for dissecting the genetic basis of common diseases, eQTL study has attracted increasing research interest. The traditional eQTL methods focus on testing the associations between individual SNPs and gene expression traits. A major drawback of this approach is that it cannot model the joint effect of a set of SNPs on a set of genes, which may correspond to biological pathways.

Recent advancement in high-throughput biology has made a variety of biological interaction networks available. Effectively integrating such prior knowledge is essential for accurate and robust eQTL mapping. However, the prior networks are often noisy and incomplete. In this chapter, we propose novel graph regularized regression models to take into account the prior networks of SNPs and genes simultaneously. Exploiting the duality between the learned coefficients and incomplete prior networks enables more robust model. We also generalize our model to integrate other types of information, such as SNP locations and gene pathways. The experimental results on both simulated and real eQTL datasets demonstrate that our models outperform alternative methods. In particular, the proposed dual refinement regularization can significantly improve the performance of eQTL mapping.

CHAPTER 5: DISCUSSION

Driven by the advancement of cost-effective and high-throughput genotyping technologies, eQTL mapping has revolutionized the field of genetics by providing new ways to identify genetic factors that influence gene expression. Traditional eQTL mapping approaches consider both SNPs and genes individually, such as sparse feature selection using Lasso and single-locus statistical tests using *t*-test or ANOVA test. However, it is commonly believed that many complex traits are caused by the joint effect of multiple genetic factors, and genes in the same biological pathway are often co-regulated and may share a common genetic basis. Thus, it is a crucial challenge to understand *how multiple, modestly-associated SNPs interact to influence the phenotypes*. However, little prior work has studied the grow-wise eQTL mapping problem. Moreover, many prior correlation structures in the form of either physical or inferred molecular networks in the genome and phenome are available in many knowledge bases, such as PPI network, and genetic interaction network. Developing effective models to incorporate prior knowledge on the relationships between SNPs and relationships between genes for more robust eQTL mapping has recently attracted increasing research interests. However, the structures of prior networks are often highly noisy and far from complete. More robust models that are less sensitive to noise and incompleteness of prior knowledge are required to integrate these prior networks for eQTL mapping.

This thesis presents a series of algorithms that take advantage of multiple domain knowledge to help with the eQTL mapping and systematically study the problem of group-wise eQTL mapping. In this chapter, we come to the conclusions of this thesis and discuss the future directions of inferring group-wise associations for eQTL mapping.

5.1 Summary

In this thesis, we presented our solutions for group-wise eQTL mapping. In general, we made the following contributions.

- **Algorithm to Detect Group-wise eQTL Associations with eQTL Data Only**

  To the best of our knowledge, this is the first work to address the group-wise eQTL mapping problem. Three algorithms (Chapter 2) are proposed to address this problem. The three approaches incrementally take into consideration more aspects, such as group-wise association, potential confounding factors and the existence of individual associations. Besides, we illustrate how each aspect could benefit the eQTL mapping. Specifically, in order to accurately capture possible interactions between multiple genetic factors and their joint contribution to a group of phenotypic variations, a sparse linear-Gaussian model (SET-eQTL) is proposed to infer novel associations between multiple SNPs and genes. The proposed method can help unravel true functional components in existing pathways. The results could provide new insights on how genes act and coordinate with each other to achieve certain biological functions. The thesis further extends the approach to consider the confounding factors and also be able to decouple *individual* associations and *group-wise* associations. The results show the superiority over those eQTL mapping algorithms that do not consider the group-wise associations.

- **Algorithm to Integrate Heterogenous Graph Data to Refine Prior Knowledge Bases**

  Based on the intuition of group-wise eQTL mapping, it is natural to integrate multi-domain knowledge about the relationships between SNPs and relationships between genes. Since the prior knowledge is usually heterogeneous, incomplete, and noisy, the thesis proposes the CGC algorithm (Chapter 3) that is robust and flexible to incorporate multiple sources graph data to enhance graph clustering performance. The CGC algorithm is able to automatically identify noisy domains. By assigning smaller

weights to noisy domains, the CGC algorithm is able to obtain optimal graph partition performance for the focused domain.

- **Robust Algorithm to Incorporate Prior Interaction Structures into eQTL Mapping**
  To incorporate the prior SNP-SNP interaction structure and grouping information of genes into eQTL mapping, the proposed algorithm, GDL (Chapter 4), significantly improve the robustness and the interpretability of eQTL mapping. We study how prior graph information would help improve eQTL mapping accuracy and how refinement of prior knowledge would further improve the mapping accuracy. In addition, other different types of prior knowledge, *e.g.*, location information of SNPs and genes, and pathway information, can also be integrated for the graph refinement.

5.2 Future Directions

We envision that the integration of multi-domain knowledge will be the center of interests for eQTL mapping in the future. In the past decade, many efforts have been devoted to developing methods for eQTL mapping. In this thesis, we present approaches that address the group-wise eQTL mapping problem. To further advance the field, there are several important research issues that should be explored.

1. **Disagreement across Diverse Information Sources**
   Although the idea of integrating multiple noisy heterogeneous data sources to establish accurate knowledge bases is straightforward, the development of such models is still very limited. Most existing integrative approaches use multiple data sources evenly without considering possible disagreement across diverse information sources. This might require an in-depth investigation. Effective data mining techniques that can simultaneously do trustworthy analysis are desirably required.
2. **Large Scale Data Sets**
   Scalability is another important issue in eQTL mapping. Especially, for human genetics, the whole genome eQTL mapping includes analysis of millions of SNPs and tens of thousands of genes. Traditional eQTL mapping approaches detect associated SNPs for

each gene separately. Thus, mapping algorithm can be deployed in parallel for each gene expression. For each run, many approaches were proposed to speed up the mapping, such as screening method (Wang et al., 2013). However, these approaches do not work for the group-wise eQTL mapping since the SNPs and genes need to be considered jointly. In our algorithm (Chapter 2), we have developed an effective approach to speed up the computing. However, it is still not able to tackle the whole genome eQTL mapping for human data set. Thus, it is desirable to design new algorithms that are capable of scaling genetic association studies across the whole-genome and support identification of multi-way interactions.

3. **Mining Biological and Medical Data Using Heterogeneous Models**

   Biological and medical research have been facing big data challenges for a long time. With the burst of many new technologies, the data are becoming larger and more complex. Our ability to identify and characterize the effects of genetic factors that contribute to complex traits depends crucially on the development of new computational approaches to integrate, analyze, and interpret these data. It is desirable to develop integrative and scalable methods to study how genetic factors interact with each other to cause common diseases. The developed techniques will dissect the relationships among different components and automatically discover most relevant patterns from the data.

REFERENCES

Andrew, G. and Gao, J. (2007). Scalable training of l1-regularized log-linear models. *International Conference on Machine Learning*.

Asuncion, A. and Newman., D. (2007). Uci machine learning repository.

Asur, S., Ucar, D., and Parthasarathy, S. (2007). An ensemble framework for clustering protein-protein interaction networks. In *Bioinformatics*, pages 29–40.

Balding, D. J. (2006). A tutorial on statistical methods for population association studies. *Nature Reviews Genetics*, 7(10):781–791.

Bickel, S. and Scheffer, T. (2004). Multi-view clustering. In *ICDM*, pages 19–26.

Biganzoli, E. M., Boracchi, P., Ambrogi, F., and Marubini, E. (2006). Artificial neural network for the joint modelling of discrete cause-specific hazards. *Artif Intell Med*, 37(2):119–130.

Bochner, B. R. (2003). New technologies to assess genotype henotype relationships. *Nature Reviews Genetics*, 4:309–314.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Braun, R. and Buetow, K. (2011). Pathways of distinction analysis: a new technique for multi-SNP analysis of GWAS data. *PLoS Genet.*, 7(6):e1002101.

Cantor, R. M., Lange, K., and Sinsheimer, J. S. (2010). Prioritizing gwas results: A review of statistical methods and recommendations for their application. *American journal of human genetics*, 86(1):6–22.

Carlos M. Carvalhoa, Jeffrey Changa, J. E. L. J. R. N. Q. W. and West, M. (2008). High-Dimensional Sparse Factor Modeling: Applications in Gene Expression Genomics. *Journal of the American Statistical Association*, pages 1438–1456.

Charles Boone, H. B. and Andrews, B. J. (2007). Exploring genetic interactions and networks with yeast. *Nature Reviews Genetic*, 8:437449.

Chaudhuri, K., Kakade, S. M., Livescu, K., and Sridharan, K. (2009). Multi-view clustering via canonical correlation analysis. In *ICML*, pages 129–136.

Chen, X., Shi, X., Xu, X., Wang, Z., Mills, R., Lee, C., and Xu, J. (2012). A two-graph guided multi-task lasso approach for eqtl mapping. In *AISTATS'12*, pages 208–217.

Cheng, W., Zhang, X., Wu, Y., Yin, X., Li, J., Heckerman, D., and Wang, W. (2012). Inferring novel associations between snp sets and gene sets in eqtl study using sparse graphical model. In *BCB'12*, pages 466–473.

Cheung, V. G., Spielman, R. S., Ewens, K. G., Weber, T. M., Morley, M., and Burdick, J. T. (2005). Mapping determinants of human gene expression by regional and genome-wide association. *Nature*, pages 1365–1369.

Chung (1997). Spectral graph theory (reprinted with corrections). In *CBMS: Conference Board of the Mathematical Sciences, Regional Conference Series*.

Cordell, H. J. (2009). Detecting gene-gene interactions that underlie human diseases. *Nat. Rev. Genet.*, 10:392–404.

Davidson, I., Qian, B., Wang, X., and Ye, J. (2013). Multi-objective multi-view spectral clustering via pareto optimization. In *SDM*, pages 234–242. SIAM.

Ding, C., Li, T., Peng, W., and Park, H. (2006). Orthogonal nonnegative matrix t-factorizations for clustering. In *KDD*, pages 126–135.

Ding, C. H. Q., Li, T., and Jordan, M. I. (2010). Convex and semi-nonnegative matrix factorizations. *IEEE Trans. Pattern Anal. Mach. Intell*, 32(1):45–55.

Dorigo, M., de Oca, M. A. M., and Engelbrecht, A. P. (2008). Particle swarm optimization. *Scholarpedia*, 3:1486.

Elbers, C. C., Eijk, K. R. v., Franke, L., Mulder, F., Schouw, Y. T. v. d., Wijmenga, C., and Onland-Moret, N. C. (2009). Using genome-wide pathway analysis to unravel the etiology of complex diseases. *Genetic epidemiology*, 33(5):419–31.

Evans, D. M., Marchini, J., Morris, A. P., and Cardon, L. R. (2006). Two-stage two-locus models in genome-wide association. *PLoS Genetics*, 2: e157.

Feng, T. and Zhu, X. (2010). Genome-wide searching of rare genetic variants in WTCCC data. *Hum. Genet.*, 128:269–280.

Fusi, N., Stegle, O., and Lawrence, N. D. (2012). Joint modelling of confounding factors and prominent genetic regulators provides increased accuracy in genetical genomics studies. *PLoS Comput. Biol.*, 8(1):e1002330.

Gao, C., Brown, C. D., and Engelhardt, B. E. (2013). A latent factor model with a mixture of sparse and dense factors to model gene expression data with confounding effects. *ArXiv e-prints*.

Gilad, Y., Rifkin, S. A., and Pritchard, J. K. (2008). Revealing the architecture of gene regulation: the promise of eQTL studies. *Trends Genet.*, 24:408–415.

Glover, F. and McMillan, C. (1986). The general employee scheduling problem. an integration of MS and AI. *Computers & OR*, 13:563–573.

Hirschhorn, J. N. and Daly, M. J. (2005). Genome-wide association studies for common diseases and complex traits. *Nature Reviews Genetics*, 6:95–108.

Hoh, J. and Ott, J. (2003). Mathematical multi-locus approaches to localizing complex human trait genes. *Nature Reviews Genetics*, 4:701–709.

Hoh, J., Wille, A., Zee, R., Cheng, S., Reynolds, R., Lindpaintner, K., and Ott, J. (2000). Selecting snps in two-stage analysis of disease association data: a model-free approach. *Annals of Human Genetics*, 64:413–417.

Holden, M., Deng, S., Wojnowski, L., and Kulle, B. (2008). GSEA-SNP: applying gene set enrichment analysis to SNP data from genome-wide association studies. *Bioinformatics*, 24(23):2784–2785.

Horvath, S. and Dong, J. (2008). Geometric interpretation of gene coexpression network analysis. *PLoS Computational Biology*, 4.

Huang, d. a. W., Sherman, B. T., and Lempicki, R. A. (2009a). Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat Protoc*, 4(1):44–57.

Huang, Y., Wuchty, S., Ferdig, M. T., and Przytycka, T. M. (2009b). Graph theoretical approach to study eqtl: a case study of plasmodium falciparum. *ISMB*, pages i15–i20.

Hub, J. S. and de Groot, B. L. (2009). Detection of functional modes in protein dynamics. *PLoS Computational Biology*.

Ideraabdullah, F., Casa-Esper, E., and et al. (2004). Genetic and haplotype diversity among wild-derived mouse inbred strains. *Genome Research*, 14(10a):1880–1887.

Jeffrey T. Leek, J. D. S. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet*, pages 1724–35.

Jenatton, R., Audibert, J.-Y., and Bach, F. (2011). Structured variable selection with sparsity-inducing norms. *JMLR*, 12:2777–2824.

Joo, J. W., Sul, J. H., Han, B., Ye, C., and Eskin, E. (2014). Effectively identifying regulatory hotspots while capturing expression heterogeneity in gene expression studies. *Genome Biol.*, 15(4):r61.

Kang, H. M., Zaitlen, N. A., Wade, C. M., Kirby, A., Heckerman, D., Daly, M. J., and Eskin, E. (2008). Efficient control of population structure in model organism association mapping. *Genetics*, 178(3):1709–1723.

Kim, S. and Xing, E. P. (2009). Statistical estimation of correlated genome associations to a quantitative trait network. *PLoS Genet.*, 5(8):e1000587.

Kim, S. and Xing, E. P. (2012). Tree-guided group lasso for multi-response regression with structured sparsity, with applications to eqtl mapping. In *ICML*.

Kuang, D., Park, H., and Ding, C. H. Q. (2012). Symmetric nonnegative matrix factorization for graph clustering. In *SDM*, pages 106–117.

Kumar, A. and III, H. D. (2011). A co-training approach for multi-view spectral clustering. In *ICML*, pages 393–400.

Kumar, A., Rai, P., and III, H. D. (2011). Co-regularized multi-view spectral clustering. In *NIPS*, pages 1413–1421.

Lander, E. S. (2011). Initial impact of the sequencing of the human genome. *Nature*, 470(7333):187–197.

Larraanaga, P. and Lozano, J. A. (2001). *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers.

Lee, D. D. and Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562.

Lee, S. and Xing, E. P. (2012). Leveraging input and output structures for joint mapping of epistatic and marginal eQTLs. *Bioinformatics*, 28(12):i137–146.

Lee, S., Zhu, J., and Xing, E. P. (2010). Adaptive multi-task lasso: with application to eqtl detection. In *NIPS*.

Lee, S.-I., Dudley, A. M., Drubin, D., Silver, P. A., Krogan, N. J., Pe'er, D., and Koller, D. (2009). Learning a prior on regulatory potential from eqtl data. *PLoS Genet*, page e1000358.

Leek, J. T. and Storey, J. D. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet.*, 3(9):1724–1735.

Leopold Parts1, Oliver Stegle, J. W. R. D. (2011). *Joint Genetic Analysis of Gene Expression Data with Inferred Cellular Phenotypes*. PLos Genetics.

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

Listgarten, J., Kadie, C., Schadt, E. E., and Heckerman, D. (2010). Correction for hidden confounders in the genetic analysis of gene expression. *Proc. Natl. Acad. Sci. U.S.A.*, 107(38):16465–16470.

Listgarten, J., Lippert, C., Kang, E. Y., Xiang, J., Kadie, C. M., and Heckerman, D. (2013). A powerful and efficient set test for genetic markers that handles confounders. *Bioinformatics*, 29(12):1526–1533.

Long, B., Yu, P. S., and Zhang, Z. M. (2008). A general model for multiple view unsupervised learning. In *SDM*, pages 822–833.

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *JMLR*, 11:2287–2322.

McClurg, P., Janes, J., Wu, C., Delano, D. L., Walker, J. R., Batalov, S., Takahashi, J. S., Shimomura, K., Kohsaka, A., Bass, J., Wiltshire, T., and Su, A. I. (2007). Genomewide association analysis in diverse inbred mice: power and population structure. *Genetics*, 176(1):675–683.

Michaelson, J., Loguercio, S., and Beyer, A. (2009a). Detection and interpretation of expression quantitative trait loci (eQTL). *Methods*, 48(3):265–276.

Michaelson, J. J., Loguercio, S., and Beyer, A. (2009b). Detection and interpretation of expression quantitative trait loci (eQTL). *Methods*, 48:265–276.

Michaelson, J. J., Loguercio, S., and Beyer, A. (2009c). Detection and interpretation of expression quantitative trait loci (eqtl). *Methods*, 48(3):265–276.

Mootha, V. K., Lindgren, C. M., Eriksson, K. F., Subramanian, A., Sihag, S., Lehar, J., Puigserver, P., Carlsson, E., Ridderstrale, M., Laurila, E., Houstis, N., Daly, M. J., Patterson, N., Mesirov, J. P., Golub, T. R., Tamayo, P., Spiegelman, B., Lander, E. S., Hirschhorn, J. N., Altshuler, D., and Groop, L. C. (2003). PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat. Genet.*, 34(3):267–273.

Musani, S., Shriner, D., Liu, N., Feng, R., Coffey, C., Yi, N., Tiwari, H., and Allison, D. (2007a). Detection of gene - gene interactions in genome-wide association studies of human population data. *Human Heredity*, 63(2):67–84.

Musani, S. K., Shriner, D., Liu, N., Feng, R., Coffey, C. S., Yi, N., Tiwari, H. K., and Allison, D. B. (2007b). Detection of gene x gene interactions in genome-wide association studies of human population data. *Human Heredity*, pages 67–84.

Nelson, M. R., Kardia, S. L., Ferrell, R. E., and Sing, C. F. (2001). A combinatorial partitioning method to identify multilocus genotypic partitions that predict quantitative trait variation. *Genome Research*, 11:458–470.

Ng, A. (2004). Feature selection, l1 vs. l2 regularization, and rotational invariance. *International Conference on Machine Learning*.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856.

Nicolo Fusi, O. S. and Lawrence, N. D. (2012). Joint modelling of confounding factors and prominent genetic regulators provides increased accuracy in genetical genomics studies. *PLoS Computational Biology*, page e1002330.

Nocedal, J. and Wright, S. J. (2006). *Numerical optimization*. Springer.

Obozinski, G. and Taskar, B. (2006). Multi-task feature selection. Technical report.

Perry, J. R. B., Mccarthy, M. I., Hattersley, A. T., Zeggini, E., Case, T., Consortium, C., Weedon, M. N., and Frayling, T. M. (2009). Interrogating type 2 diabetes genome-wide association data using a biological pathway-based approach. *Diabetes*, 58(June).

Pujana, M. A., Han, J.-D. J., Starita, L. M., Stevens, K. N., and Muneesh Tewari, e. a. (2007). Network modeling links breast cancer susceptibility and centrosome dysfunction. *Nature Genetics*, pages 1338–1349.

Rachel B. Brem, John D. Storey, J. W. and Kruglyak, L. (2005). Genetic interactions between polymorphisms that affect gene expression in yeast. *Nature*, pages 701–03.

Ritchie, M. D., Hahn, L. W., Roodi, N., Bailey, L. R., Dupont, W. D., Parl, F. F., and Moore, J. H. (2001). Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *American Journal of Human Genetics*, 69:138–147.

Rockman, M. V. and Kruglyak, L. (2006). Genetics of global gene expression. *Nature Reviews Genetics*, 7:862–872.

Smith, E. N. and Kruglyak, L. (2008). Gene-environment interaction in yeast gene expression. *PLoS Biol*, page e83.

Späth, H. (1985). *Cluster Dissection and Analysis. Theory, FORTRAN programs, examples*. Ellis Horwood.

Stegle, O., Kannan, A., Durbin, R., and Winn, J. (2008). Accounting for non-genetic factors improves the power of eqtl studies. In *RECOMB*, pages 411–422.

Stegle, O., Parts, L., Durbin, R., and Winn, J. (2010). A bayesian framework to account for complex non-genetic factors in gene expression levels greatly increases power in eqtl studies. *PLoS Computational Biology*, page e1000770.

Tang, W., Lu, Z., and Dhillon, I. S. (2009). Clustering with multiple graphs. In *ICDM*, pages 1016–1021.

The Gene Ontology Consortium (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 58(1):267–288.

Torkamani, A., Topol, E. J., and Schork, N. J. (2008). Pathway analysis of seven common diseases assessed by genome-wide association. *Genomics*, 92(5):265–72.

van Dongen, S. (2000). A cluster algorithm for graphs. In *Centrum voor Wiskunde en Informatica (CWI)*, page 40.

von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S., and Bork, P. (2002). Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417:399–403.

Wang, J., Zhou, J., Wonka, P., and Ye, J. (2013). Lasso screening rules via dual polytope projection. In *NIPS*, pages 1070–1078.

Wang, K., Li, M., and Hakonarson, H. (2010). Analysing biological pathways in genome-wide association studies. *Nature Reviews Genetics*, 11(12):843–854.

Wang, X. and Davidson, I. (2010). Flexible constrained spectral clustering. In *KDD*, pages 563–572.

Westfall, P. H. and Young, S. S. (1993). *Resampling-based Multiple Testing*. Wiley, New York.

Wu, M. C., Lee, S., Cai, T., Li, Y., Boehnke, M., and Lin, X. (2011). Rare-variant association testing for sequencing data with the sequence kernel association test. *Am. J. Hum. Genet.*, 89(1):82–93.

Xu, W., Liu, X., and Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *SIGIR*, Clustering, pages 267–273.

Yang, C., He, Z., Wan, X., Yang, Q., Xue, H., and Yu, W. (2009). SNPHarvester: a filtering-based approach for detecting epistatic interactions in genomewide association studies. *Bioinformatics*, 25(4):504–511.

Yang, C., Wang, L., Zhang, S., and Zhao, H. (2013). Accounting for non-genetic factors by low-rank representation and sparse regression for eQTL mapping. *Bioinformatics*, pages 1026–1034.

Yu, G.-X., Rangwala, H., Domeniconi, C., Zhang, G., and Zhang, Z. (2013). Protein function prediction by integrating multiple kernels. In *IJCAI*.

Yvert, G., Brem, R. B., Whittle, J., Akey, J. M., Foss, E., Smith, E. N., Mackelprang, R., and Kruglyak, L. (2003). Trans-acting regulatory variation in Saccharomyces cerevisiae and the role of transcription factors. *Nat. Genet.*, 35(1):57–64.

Zhang, X., Huang, S., Zou, F., and Wang, W. (2010). TEAM: Efficient two-locus epistasis tests in human genome-wide association study. *Bioinformatics*, 26(12):i217–227.

Zhu, J., Zhang, B., Smith, E. N., Drees, B., Brem, R. B., Kruglyak, L., Bumgarner, R. E., and Schadt, E. E. (2008). Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nature Genetics*, pages 854–61.