Structured Sparse Regression Methods for Learning from High-Dimensional Genomic Data

Micol Marchetti-Bowick
Machine Learning Department
Carnegie Mellon University
micolmb@cs.cmu.edu

Thesis Proposal Draft
January 11, 2018

Thesis Committee:
Eric P. Xing (CMU)
Jian Ma (CMU)
Seyoung Kim (CMU)
Su-In Lee (UW)

Abstract

The past several decades have witnessed an unprecedented explosion in the size and scope of genomic datasets, paving the way for statistical and computational data analysis techniques to play a critical role in driving scientific discovery in the fields of biology and medicine. However, genomic datasets suffer from a number of problems that weaken their signal to noise ratio, including small sample sizes and widespread data heterogeneity. As a result, the naive application of traditional machine learning approaches to many problems in computational biology can lead to unreliable results and spurious conclusions.

In this thesis, we propose several new techniques for extracting meaningful information from noisy genomic data. To combat the challenges posed by high-dimensional, heterogeneous datasets, we leverage prior knowledge about the underlying structure of a problem to design models with increased statistical power to distinguish signal from noise. Specifically, we rely on structured sparse regularization penalties to encode relevant information into a model without sacrificing interpretability. Our models take advantage of knowledge about the structure shared among related samples, features, or tasks, which we derive from biological insights, to boost their power to identify true patterns in the data.

Finally, we apply these methods to several widely studied problems in computational biology, including identifying genetic loci that are associated with a phenotype of interest, learning gene regulatory networks, and predicting the survival rates of cancer patients. We demonstrate that leveraging prior knowledge about the structure of a problem leads to increased statistical power to detect associations between different components of a biological system (e.g., SNPs and genes), providing greater insight into complex biological processes, and to more accurate predictions of disease phenotypes, leading to improved diagnosis or treatment of human diseases.

Contents

Chapter 1

Introduction

1.1 A Wealth of Information

When the first complete human genome sequence was published in the early 2000s, it was the culmination of a project that spanned nearly 15 years and cost approximately $3 billion. Over the past two decades since that milestone was reached, the development of next-generation sequencing technologies has led to an unprecedented explosion in the sheer quantity of genomic data that is generated and stored on a daily basis. The amount of data has expanded so rapidly that its rate of growth has been projected to outstrip both every other scientific domain and every source of user-generated content on the web to become the ultimate new source for truly "big" data over the next 10 years [1].

The information contained within these datasets has the potential to answer age-old questions about how the human body works that continue to perplex scientists and physicians today. However, due to their vast size and complexity, gleaning meaningful information from the raw data is virtually impossible without the aid of computational techniques. By far the most promising among these are statistical methods that can uncover subtle but salient patterns buried deep in the data that a human might never detect.

Recent years have seen a confluence of factors that are critical for knowledge discovery in this sphere: larger and more comprehensive datasets, cheaper and faster computer processors, and new models and algorithms that take advantage of both. These changes have paved the way for statistical and computational data analysis techniques to play a critical role in the fields of biology and medicine over the coming decades.

1.2 The Promise of Omic Data

The work in this thesis focuses on two major areas of research within the field of computational biology. The first is the study of the cellular processes that form the basis for the relationship between an organism's genotype and phenotype. The second is the study of how computational techniques can be used to improve the diagnosis and treatment of human diseases.

In the first domain, the principal goal is to understand how changes in an organism's DNA sequence lead to changes in one or more observable characteristics. Many different types of so-called "omic" data are commonly used to study this question. They include *genomic* data that captures information about the raw DNA sequence, *transcriptomic* data that captures information about the genes that are transcribed to RNA, *proteomic* data that captures information about proteins, and *metabolomic* data that captures information about small-molecule chemicals found within the cell. These datasets capture interactions among genes, proteins, and other molecular

structures, including feedback loops and regulatory networks. As a result, they can be used to gain insight into the complex mechanisms that govern the influence of molecular changes on external traits ranging from physical appearance to disease status. Ultimately, these insights help scientists formulate a more complete picture of how organisms function.

In the second domain, the aim is to develop new methods for better diagnosis, prognosis, and treatment of an array of human diseases. A wide range of datasets can be combined to address these problems, including patient medical histories, clinical test results, molecular profiles of diseased tissue, treatment information, patient outcomes, and more. These datasets capture the complex interactions between each patient's unique disease, the treatments they receive, and their response to those treatments. As a result, they can be used to improve the accuracy and specificity of disease diagnosis, provide more realistic prognostic information, and aid physicians in making personalized decisions about the best course of treatment for each individual patient.

Both of these areas have seen an explosion of data in recent years due in part to the spread of high-throughput data acquisition techniques. In addition, policy changes in some countries, such as the HITECH Act in the United States, have led to the expanded use of electronic health records to store patients' medical data. Finally, the rise of online databases such as GenBank, GEO, TCGA, and many others have made both genomic and clinical datasets broadly accessible to researchers all over the world. These changes, most of which have occurred only over the past 15-20 years, have contributed to the collection and dissemination of petabytes of biological and medical data.

1.3 Obstacles to Knowledge Discovery

Despite the sheer quantity of information captured by the wide range of biological datasets available today, they have one major drawback: they are inherently extremely noisy. The low signal-to-noise ratio (SNR) is an enormous roadblock to answering all of the questions that scientists would like to address within this domain. Although all datasets contain some noise, there are certain characteristics specific to biological datasets that pose a unique set of challenges.

The first challenge is the fact that nearly all biological datasets, and particularly omic datasets, contain a small number of samples but a large number of measurements for each sample. In other words, the datasets contain many more features than samples. This situation is prevalent among biological datasets for two important reasons. First, collecting each sample is very expensive, because one sample frequently corresponds to an individual organism (e.g., a human patient) or a specific wet lab condition. Second, due to high-throughput techniques (such as whole-genome sequencing), extracting a large number of features from each sample is becoming increasingly cost effective. This results a situation known as the high-dimensional data setting, in which the sample size is often many orders of magnitude smaller than the dimensionality of each sample.

In statistics, this problem is often called the curse of dimensionality. As the dimensionality d of a dataset increases, the total volume of the feature space increases exponentially in d. This means that a fixed number of data points will become increasingly separated in space as their dimensionality grows, making it extremely challenging for statistical methods to identify reliable patterns in high-dimensional settings. Because we want to extract patterns from the data that generalize to unseen samples, high-dimensional datasets inherently have a very low signal-to-noise ratio, and therefore provide an especially challenging setting for machine learning.

The second problem is the extreme heterogeneity of biological data. The heterogeneity comes from a number of sources. First, there are many different types of data. This makes it difficult to combine datasets in order to obtain a larger sample size. Second, datasets of the same type still often originate from many different sources, where they may be collected using different tools

or procedures and pre-processed in different ways, which introduces completely different types of measurement error and other noise. Because of this, each dataset generated by a different experimental study may be biased in a different way. Even when two different studies measure the same set of features across a different set of samples, the samples should not be naively combined. Third, even samples within the same dataset collected from a single source may be heterogeneous and intercorrelated. This is particularly the case when dealing with human data (compared with data collected from organisms grown in vitro) because we are forced to collect data only from the samples that are available in the real world.

From a statistical perspective, this means that nearly all biological datasets contain samples that are not I.I.D., i.e. are not drawn independently from the same underlying distribution with the same sources of noise. Because statistical models frequently rely upon I.I.D. assumptions, this inherent heterogeneity again makes it difficult for traditional machine learning techniques to extract signal from the data.

As a result of the low SNR, when applying statistical analysis techniques to biological datasets, it's easy to identify spurious patterns that are not real, but merely artifacts of the data. Although I.I.D. assumptions do not hold in many real-world datasets, the problem of non-I.I.D. data is exacerbated in the high-dimensional data setting, making biological data one of the most challenging types of data to work with from a machine learning perspective.

1.4 How Machine Learning Can Help

Despite these challenges, specialized machine learning techniques can be designed to boost the signal-to-noise ratio and reach meaningful conclusions from complex, noisy datasets. There are several widely studied approaches for achieving this goal.

One common approach is to constrain the learning task by encoding domain-specific prior knowledge into the model. This can be viewed as restricting the space of hypotheses that we are allowed to choose from when fitting a statistical model. Given that we restrict ourselves to only considering the hypotheses that fit with our prior knowledge, we are more likely to select a "correct" hypothesis that identifies true patterns in the data rather than spurious patterns arising from noisy observations or correlated samples. This approach relies on making assumptions about the structure of the problem, which are typically derived from pre-existing biological knowledge.

Another approach is to share information between tasks that are related but not identical. Called transfer learning, this approach can help combat small sample sizes by leveraging samples across multiple tasks even when the same features are not available for all tasks. Rather than naively combining samples across related tasks, transfer learning approaches provide a framework for leveraging the patterns identified in one dataset to restrict the set of hypothesis we consider when analyzing another dataset, and vice versa.

Yet another approach is to learn a compact but informative representation of the data from the observed features. Known as representation learning, this approach can help reduce noise by identifying a representation of the input space that clearly captures the factors that explain variation in the output. This can also help combat the high-dimensional data setting by reducing the dimensionality of the input space. There are many different approaches to representation learning, including feature selection methods that choose the most informative subset of features, regularization methods that shrink the coefficients of the least informative features to zero, and feature combination methods that learn a set of latent features as a function of the raw features.

1.5 Thesis Statement

The goal of this thesis is to develop new machine learning techniques for extracting signal from inherently noisy biological datasets. In particular, we will leverage ideas from the three categories of approaches described in the previous section, and will introduce new methods that apply these ideas to several problems in computational genomics and health informatics, including:

1. Genome wide association studies (GWAS), whose goal is to identify genomic loci (e.g., SNPs) whose genotype affects a particular downstream trait.
2. Gene network estimation, whose goal is to understand the regulatory relationships among a set of genes using mRNA expression data or other transcriptomic information.
3. Survival analysis, which entails leveraging genomic and clinical data to predict how long a patient with a given disease (e.g., a given type of cancer) will survive.

To address these problems, we leverage regularization penalties to incorporate structure into statistical models. Specifically, we describing a uniform modeling framework that can be used as a guideline for designing methods that can learn from low-SNR data. Consider a model parameterized by θ that we want to estimate from a dataset x given additional information α. We construct the following optimization problem to estimate θ.

$$\min_{\theta} \ \text{loss}(\theta; x) + \lambda \ \text{sparsity penalty}(\theta) + \gamma \ \text{structure penalty}(\theta; \alpha)$$

Here λ and γ are hyperparameters that are not known ahead of time but require additional tuning, whereas α represents external knowledge about the structure of θ that is known *a priori* and is used to specify the penalty term. The structure penalty serves to effectively restrict the space of possible values of θ to those that satisfy the structure captured in α. We purposefully define this term in a very flexible way. In practice, the structure penalty may encode prior biological knowledge, information shared across multiple related task, information shared across multiple related feature sets, or anything else.

This framework can be used to design models for a wide range of tasks across both supervised and unsupervised learning. In the next section, we demonstrate that many existing models can be cast into this framework. Furthermore, all of the new methods proposed in this thesis are captured by the above formulation, although in some cases the sparsity penalty and structure penalty are combined into a single penalty term.

In the remainder of this thesis proposal, we introduce three novel approaches for learning from high-dimensional, heterogeneous, noisy data. We first introduce a time-varying group sparse additive model for GWAS that is capable of detecting a sparse set of genomic loci that are associated with phenotypes that vary over time. This method leverages assumptions about the smoothly varying nature of SNP effects on a phenotype to boost the statistical power of GWAS. Next, we develop a structured multi-task regression model for jointly performing eQTL mapping and gene network estimation. This approach shares information between these two tasks via a structured sparsity penalty that is designed based on external knowledge about the relationship between SNP-gene and gene-gene associations. Finally, we propose a representation learning method that is tailored toward high-dimensional, noisy data, and uses structured sparsity to simultaneously perform feature selection and feature combination. We apply this method to learning compact representations of cancer genomic data in order to better predict the survival rates of cancer patients.

For each of the methods described above, we present rigorous empirical evaluations on both simulated and real data, and demonstrate that our approaches achieve greater statistical power to distinguish signal from noise compared with baseline methods.

Chapter 2

Time-Varying Group SpAM

2.1 Introduction

The goal of genome-wide association studies (GWAS) is to analyze a large set of genetic markers that span the entire genome in order to identify loci that are associated with a phenotype of interest. Over the past decade, GWAS has been used to successfully identify genetic variants that are associated with numerous diseases and complex traits, ranging from breast cancer to blood pressure [2]. However, a significant challenge in performing GWAS is that the studies are often vastly under-powered due to the high dimensionality of the feature set relative to the small number of human samples available.

Traditional GWAS methodologies test each variant independently for association with the phenotype, and use a stringent significance threshold to adjust for multiple hypothesis testing [3]. While this approach works well for traits that depend on strong effects from a few loci, it is less suitable for complex, polygenic traits that are influenced by weak effects from many different genetic variants. More recently, a significant body of work has emerged on penalized regression approaches for GWAS that capture the joint effects of all markers [4; 5]. The majority of these methods model the phenotype as a weighted sum of the genotype values at each locus, and use a regularization penalty such as the ℓ_1 norm to identify a sparse set of SNPs that are predictive of the trait. Although this technique helps to reduce overfitting and detect fewer spurious SNP-trait associations, the lack of statistical power to identify true associations persists.

Here we aim to further boost the statistical power of GWAS by proposing a new model that leverages dynamic trait data, in which a particular trait is measured in each individual repeatedly over time, as depicted in Figure 2.1(a). Such datasets are often generated by longitudinal studies that follow participants over the course of months, years, or even decades. Though broadly available, dynamic trait datasets are frequently underutilized by practitioners who ignore the temporal information. We believe that leveraging time-sequential trait measurements in GWAS can lead to greater statistical power for association mapping.

To illustrate this concept, consider the hypothetical patterns of SNP influence on the phenotype shown in Fig 2.1(b). As in traditional GWAS, an association between a SNP and the phenotype exists if the three SNP genotypes (which we denote AA, Aa, and aa) have differential effects on the trait. In the first example, the effects of the three SNP genotypes only differ in the $t \in [0.5, 1]$ time interval. A static method that uses data from an arbitrarily chosen time point or simply treats the time series as i.i.d. samples could easily miss this association, whereas a dynamic method that considers the entire dataset would detect it. The second example shows a SNP in which the difference between the effects of the three genotypes is small but consistent over time. Although this signal could be too weak to be interpreted as a significant association in the static case, it gets

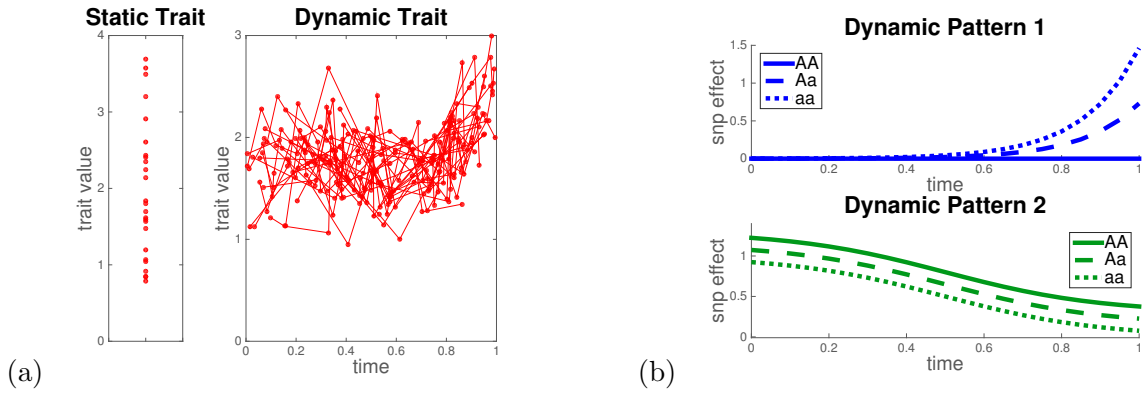


Figure 2.1: GWAS has greater statistical power with dynamic traits. (a) A toy dataset illustrating the difference between static and dynamic traits. (b) Two synthetic examples of time-dependent patterns of SNP influence on the trait that would be difficult to detect with a static model.

much stronger once evidence from the entire time series is considered.

The longitudinal data setting is challenging because traits are measured at irregularly spaced time points over subject-specific intervals. One approach that has been proposed for performing GWAS of dynamic traits, called functional GWAS, or fGWAS [6], constructs a separate model to estimate the smooth, time-varying influence of each SNP on the phenotype. Once the mean effects have been estimated for each genotype at each time point, a hypothesis test is performed to determine whether the SNP has any additive or dominant effect on the trait. Although the use of dynamic trait data gives fGWAS more statistical power than a standard hypothesis test on static data, the principal drawback of this method is that it is inappropriate for modeling complex traits that arise from interactions between genetic effects at different loci. A related approach extends the fGWAS framework to model multiple SNPs at once using a Bayesian group lasso framework [7]. Although this approach seems promising, it is severely limited by its very slow MCMC inference procedure. There are a number of other methods that have been developed for dynamic trait GWAS, including [8], [9], [10], and [11]. However, the majority of them either perform single-locus analysis (as in fGWAS) or fail to learn an explicit, interpretable representation of the dynamic effects of the genetic variants at each locus.[1]

In this work, we introduce a new penalized multivariate regression approach for GWAS of dynamic quantitative traits, in which the phenotype is modeled as a sum of nonparametric, time-varying SNP effects. We call this Time-Varying Group Sparse Additive Models, or TV-GroupSpAM. Our method is based on GroupSpAM [12], a nonparametric regression model with a group-structured penalty over the input features, which we extend to capture the dynamic effects of SNPs. This model has three major advantages over existing approaches: (1) we leverage dynamic trait data; (2) we model the contribution of each SNP to the phenotype as a smooth function of time, and explicitly learn these influence patterns; (3) we model the combined effects of multiple SNPs on the phenotype and select a sparse subset that participate in the model, thereby identifying meaningful SNP-trait associations. We show that TV-GroupSpAM exhibits desirable empirical advantages over baseline methods on both simulated and real datasets.

[1]The notable exception to this is fGWAS with Bayesian group lasso, which we directly compare to our approach.

2.2 Method

In this section, we first introduce a time-varying additive model for dynamic complex traits that captures the underlying patterns of genetic effects. We then apply a group sparse regularization scheme to this model in order to impose bias useful for discovering a sparse set of markers that influence the phenotype in a longitudinal setting. Finally, we provide an efficient optimization algorithm for parameter estimation, and thereby association mapping, under our model.

Notation. Let $X_{ij} \in \{0, 1, 2\} : i = 1, \ldots, n \ ; \ j = 1, \ldots, p$ denote the genotype of individual i at SNP locus j, where n and p denote the number of individuals and SNPs, respectively. Let $Y_{i\tau} \in \mathbb{R} : i = 1, \ldots, n \ ; \ \tau = 1, \ldots, m$ denote the phenotype value of individual i at the τ-th time point. Note that the exact time readings for different individuals at their τ-th time point may be different, i.e. the measurements are not necessarily time-aligned. We therefore introduce an explicit time variable $T_{i\tau} \in \mathbb{R}^+$ to capture the time reading for individual i at the τ-th time point, and define $Y_{i\tau} \equiv Y(T_{i\tau})$ as a stochastic process that captures the trait values at each time point. In what follows, we will use uppercase letters X, Y, T to denote random variables and lowercase letters x, y, t to denote their instantiated values.

2.2.1 Time-Varying Additive Model

We consider the following time-varying additive model with scalar input variables $X_1, \ldots, X_p$ and functional response variable $Y(T)$:

$$Y(T) = f_0(T) + \sum_{j=1}^{p} f_j(T, X_j) + \omega(T) \tag{2.1}$$

Here $Y(T)$, which represents the trait value at time T, is decomposed into three terms: $f_0(T)$ is an intercept term that represents the non-genetic influence on the phenotype at time T (e.g. from unknown environmental factors); $f_j(T, X_j)$ represents the genetic effect of marker j with genotype X_j at time T; $\omega(T)$ is the noise term that models the random fluctuation of the underlying process.

Since X_j is a categorical variable, each bivariate component function f_j can be represented more simply as a set of three univariate functions of time, given by $f_j = \{f_j^0, f_j^1, f_j^2\}$. We can then define $f_j(T, X_j) = \sum_g f_j^g(T)\, \mathbb{I}\{X_j = g\}$ where $f_j^g(\cdot) = f_j(\cdot, X_j = g)$. Next we simplify our notation by expanding each X_j into a set of three binary indicator variables such that $X_j^g = 1 \Leftrightarrow X_j = g$. This allows us to rewrite the model in the following form.

$$Y(T) = f_0(T) + \sum_{j=1}^{p} \sum_{g=0}^{2} f_j^g(T)\, X_j^g + \omega(T) \tag{2.2}$$

Note that in the above formulation, the indicator variable X_j^g selects a single function among the set $\{f_j^0, f_j^1, f_j^2\}$ for each SNP.

In the data setting, since each observation is subject to measurement error, we assume $Y_{i\tau} = Y_i(T_{i\tau}) + \epsilon_{i\tau}$ where $\epsilon_{i\tau} \sim \mathcal{N}(0, \sigma^2)$. It follows from the model defined in (2.2) that the observed phenotypic values satisfy

$$y_{i\tau} = f_0(t_{i\tau}) + \sum_{j=1}^{p} \sum_{g=0}^{2} f_j^g(t_{i\tau})\, x_{ij}^g + \omega(t_{i\tau}) + \epsilon_{i\tau} \tag{2.3}$$

for subjects $i = 1, \ldots, n$ and measurements $\tau = 1, \ldots, m$. In the remainder of this article, we assume that the residual errors $e_{i\tau} = \omega(t_{i\tau}) + \epsilon_{i\tau}$ are i.i.d. across both subjects and measurements, though an alternative approach would be to impose an autocorrelation structure on $\omega(T)$ to capture the temporal pattern of the underlying longitudinal process [6; 11].

In the model specified above, our only assumption about the dynamic genetic effects $\{f_j^0, f_j^1, f_j^2 : j = 1, \ldots, p\}$ is that they are smooth functions of time. A well-established approach to estimate nonparametric functions in additive models [13] is to minimize the expected squared error loss:

$$h(f) = \mathbb{E}\left[Y(T) - f_0(T) - \sum_{j=1}^{p}\sum_{g=0}^{2} f_j^g(T)\, X_j^g\right]^2 \tag{2.4}$$

where the expectation is calculated with respect to the distributions over SNP genotypes $(X_1, \ldots, X_p)$, time T, and phenotypic value Y. In the sample setting, this translates to minimizing

$$\hat{h}(f) = \sum_{i=1}^{n}\sum_{\tau=1}^{m}\left(y_{i\tau} - f_0(t_{i\tau}) - \sum_{j=1}^{p}\sum_{g=0}^{2} f_j^g(t_{i\tau})\, x_{ij}^g\right)^2 \tag{2.5}$$

subject to a set of smoothness constraints over each function. We go into detail about how to estimate the parameters of this model in Section 2.2.3.

2.2.2 Group Sparse Regularization

In a typical genome-wide association study, though a large number of markers are assayed, it is believed that only a small subset of them have a real effect on the trait of interest. This assumption motivates us to impose sparsity at the level of the SNPs $X_1, \ldots, X_p$ in the time-varying additive model of (2.2), such that the effects of many of these variables are zero. To achieve this, we apply a group-sparsity-inducing penalty that leads to shrinkage on the estimated effect of each locus as a whole, including the component functions for all genotypes and their values at all time points. Specifically, we employ a group norm penalty over the component functions in which each group consists of the three functions $\{f_j^0, f_j^1, f_j^2\}$ that correspond to a particular marker X_j.

To construct this group penalty, we use the $\ell_{1,2}$ norm first introduced in the context of the group lasso [14]. The empirical objective function for our model with group sparsity is given by

$$\hat{h}(f) + \lambda \sum_{j=1}^{p} \sqrt{\sum_{g=0}^{2} \|f_j^g\|_2^2} \tag{2.6}$$

and is again subject to a set of smoothness constraints. Here $\lambda > 0$ is a tunable regularization parameter that controls the amount of sparsity in the model, and the squared ℓ_2 norm over f_j^g is defined as

$$\|f_j^g\|_2^2 = \sum_{i=1}^{n}\sum_{\tau=1}^{m} f_j^g(t_{i\tau})^2 x_{ij}^g \tag{2.7}$$

The penalty term in (2.6) induces sparsity at the level of groups by encouraging each set of functions $\{f_j^0, f_j^1, f_j^2\}$ to be set exactly to zero, which implies that the corresponding marker X_j has no effect whatsoever on the phenotype at any time point.

In what follows, we will refer to the model defined by the objective function in (2.6) as a Time-Varying Group Sparse Additive Model (TV-GroupSpAM). This model is based on both the Group Sparse Additive Model of [12], in which a group sparse regularization penalty is applied to a standard additive model, and the Time-Varying Additive Model of [15], in which an unpenalized additive model is used to regress a functional response on scalar covariates.

2.2.3 Optimization Algorithm

To estimate the TV-GroupSpAM model, we use a block coordinate descent algorithm in which we optimize the objective with respect to a particular group of functions at once while all remaining functions are kept fixed.

Before presenting a complete algorithm for the regularized model, we first describe how to estimate the simpler, unpenalized model introduced in Section 2.2.1. Given the loss function of (2.4), some algebra shows that the optimal solution for f_j^g satisfies the following conditional expectation for each genetic marker $j = 1, \ldots, p$ and each genotype value $g \in \{0, 1, 2\}$.

$$f_j^g(T) = \mathbb{E}\left[Y(T) - f_0(T) - \sum_{k \neq j} \sum_{\ell} f_k^\ell(T)\, X_k^\ell \,\middle|\, T, X_j = g \right] \tag{2.8}$$

A similar formula holds for the intercept term $f_0(T)$.

It has been well established in the statistics literature that a scatterplot smoother matrix can be viewed as a natural estimate of the conditional expected value [13]. To evaluate (2.8) in the sample setting, we therefore replace the conditional expectation operator $\mathbb{E}[\cdot \,|\, T, X_j = g]$ by left multiplication with an n-by-n smoother matrix $\mathbf{S}_j^g = \{S_j^g[a, b]\}$, which is defined as

$$\begin{aligned} &S_j^g[a,b] \propto K_h(|t^{(a)} - t^{(b)}|) && \text{if } x_j^{(a)} = g \text{ and } x_j^{(b)} = g \\ &S_j^g[a,b] = 0 && \text{otherwise} \end{aligned}$$

where (a, b) is a pair of data points, each corresponding to a particular individual i and time point τ, and K_h is a smoothing kernel function with bandwidth h. An alternative way to think about $\mathbf{S}_j^g$ is as the element-wise product of a smoother matrix for T, in which entry (a, b) is proportional to $K_h(|t^{(a)} - t^{(b)}|)$, and an indicator matrix for $X_j = g$, in which entry (a, b) is given by $\mathbb{I}\{x_j^{(a)} = x_j^{(b)} = g\}$. This makes intuitive sense because we want to estimate a smooth function over time for each genotype value of each SNP. Thus, to learn each function f_j^g for a particular SNP j and a particular genotype g, we only want to consider data points for which the genotype at SNP j is g and we want to smooth over time.

The empirical estimate of f_j^g will be a vector $\hat{\mathbf{f}}_j^g \in \mathbb{R}^{nm}$ whose entries correspond to smoothed estimates of the effect of marker j with genotype g on the phenotype at each of the observed time points. Note that the entries of $\hat{\mathbf{f}}_j^g$ corresponding to samples with genotype $\neq g$ for SNP j will be set to zero because the function is not applicable to those samples. In practice, we drop these dummy entries at the very end to obtain our final function estimates. We calculate $\hat{\mathbf{f}}_j^g$ using the empirical formula for (2.8), given by

$$\hat{\mathbf{f}}_j^g = \mathbf{S}_j^g \left(\mathbf{y} - \hat{\mathbf{f}}_0 - \sum_{k \neq j} \sum_{\ell} \hat{\mathbf{f}}_k^\ell \, \mathbb{I}\{\mathbf{x}_k = \ell\} \right) \tag{2.9}$$

where $\mathbf{y}$ is the vector of concatenated trait values for each sample, and $\mathbf{x}_k$ is the corresponding vector of genotypes at SNP k for each sample. Here a sample is a measurement for a specific individual i at a specific time point τ. Cycling though SNPs and genotypes one at a time and applying the update rule of (2.9) leads to a variant of the well-known backfitting algorithm. We refer the readers to [13] for more details about smoothing and backfitting.

Finally, in order to optimize the penalized objective given in (2.6), we adapt the block coordinate descent and thresholding algorithms from [12] to our setting. The complete optimization routine is shown in Algorithm 2.1. After smoothing the partial residual at each iteration, we perform a thresholding step by estimating the group norm $\hat{w}_j$ and using it to determine whether the group

of functions $\hat{\mathbf{f}}_j$ should be set to zero. If not, we re-estimate the function values by iteratively solving a fixed point equation. We note that Step 9 of our algorithm runs more efficiently than the corresponding step of the thresholding algorithm presented in [12] because we do not need to perform a matrix inversion on each iteration. This property results from the fact that within a particular group of function estimates $\{\hat{\mathbf{f}}_j^0, \hat{\mathbf{f}}_j^1, \hat{\mathbf{f}}_j^2\}$, each one covers a disjoint set of observations, which ultimately simplifies the update equation.

2.3 Experiments

Next we conduct experiments on both simulated and real data to compare the performance of our approach against several baselines and evaluate its ability to detect genomic loci that are associated with a dynamic trait of interest.

2.3.1 Simulation Study

In order to illustrate the utility of our method, we perform several experiments on synthetic data. We generate data according to the following procedure. First we construct a set of realistic genotypes X_{ij} by randomly subsampling individuals and SNPs from the real asthma dataset that we analyze in the next section. Next we independently sample time points $T_{i\tau} \sim \text{Unif}(0, 1)$ and measurement errors $\epsilon_{i\tau} \sim \mathcal{N}(0, 1)$. We select a subset of SNPs that will have nonzero contribution to the phenotype by placing their functions in an active set $\mathcal{A} \subseteq \{f_1, \ldots, f_p\}$. We then construct

Algorithm 2.1 Block Coordinate Descent for TV-GroupSpAM

1: **inputs:** genotypes $\mathbf{x}_1, \ldots, \mathbf{x}_p$, time points $\mathbf{t}$, trait values $\mathbf{y}$
2: initialize $\hat{\mathbf{f}}_0 = \mathbf{0}$ and $\hat{\mathbf{f}}_j^g = \mathbf{0}$ for $j = 1, \ldots, p$ and $g \in \{0, 1, 2\}$
3: **repeat**
4: update intercept term: $\hat{\mathbf{f}}_0 = \mathbf{S}_0\big(\mathbf{y} - \sum_k \sum_\ell \hat{\mathbf{f}}_k^\ell \, \mathbb{I}\{\mathbf{x}_k = \ell\}\big)$
5: **for** $j = 1, \ldots, p$ **do**
6: compute partial residual: $\hat{\mathbf{R}}_j = \mathbf{y} - \hat{\mathbf{f}}_0 - \sum_{k \neq j} \sum_\ell \hat{\mathbf{f}}_k^\ell \, \mathbb{I}\{\mathbf{x}_k = \ell\}$
7: estimate projected residuals by smoothing:

$$\hat{\mathbf{P}}_j^g = \mathbf{S}_j^g \, \hat{\mathbf{R}}_j \quad \forall g$$

8: compute group norm:

$$\hat{w}_j = \sqrt{\textstyle\sum_{g=0}^{2} \|\hat{\mathbf{P}}_j^g\|_2^2}$$

9: **if** $\hat{w}_j \leq \lambda$ **then** set $\hat{\mathbf{f}}_j^g = \mathbf{0} \quad \forall g$
10: **else** update $\hat{\mathbf{f}}_j^g \; \forall g$ by iterating until convergence

$$\hat{\mathbf{f}}_j^{g+} := \Big(1 + \lambda / \|\hat{\mathbf{f}}_j\|_2\Big)^{-1} \hat{\mathbf{P}}_j^g$$

11: **end if**
12: center each $\hat{\mathbf{f}}_j$ by subtracting its mean
13: **end for**
14: **until** convergence
15: **outputs:** estimates $\hat{\mathbf{f}}_0$ and $\hat{\mathbf{f}}_j = \{\hat{\mathbf{f}}_j^0, \hat{\mathbf{f}}_j^1, \hat{\mathbf{f}}_j^2\}$ for $j = 1, \ldots, p$

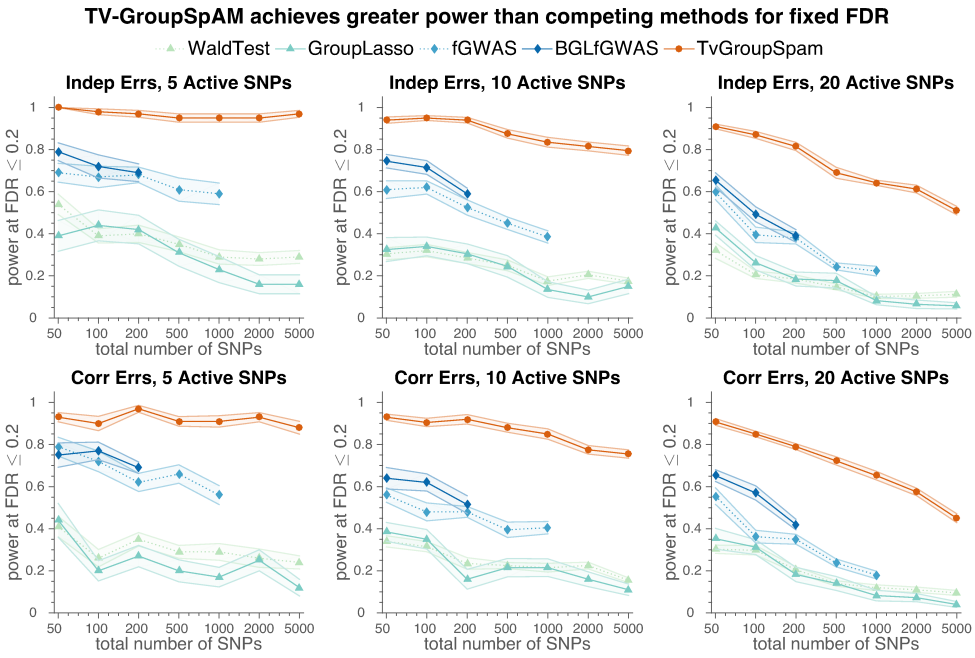


Figure 2.2: Comparison of TV-GroupSpAM to baseline methods shows that our approach achieves greater power for a fixed false discovery rate (FDR ≤ 0.2). The results are averaged over 20 random synthetic datasets for each setting, and the shaded region denotes the standard error.

the active functions by sampling their values from a diverse set of predefined influence patterns that exhibit a variety of trait penetrance models (including additive, multiplicative, dominant, and recessive) and interact differently with time (including some static patterns for balance). All functions not in the active set, including the intercept term, are defined such that $f(t) = 0 \; \forall \, t$. Finally, we generate phenotype values $y_{i\tau}$ according to the model defined in (2.3).

To test the robustness of our model, we generate data according to two slightly different variants of (2.3). In the first setting, we uphold our original assumption that the residual errors are completely uncorrelated by independently generating $\omega_{i\tau} \sim \mathcal{N}(0, \sigma^2)$. In the second setting, we invalidate this assumption and introduce strong correlation among the errors across time by jointly generating $(\omega_{i1}, \ldots, \omega_{im}) \sim \mathcal{N}(0, \Sigma)$. In all of our experiments, we fix the number of samples at $n = 100$ and the number of time points at $m = 10$. Then, to evaluate our approach in a broad range of settings, we vary the total number of SNPs over $p \in \{50, 100, 200, 500, 1000, 2000, 5000\}$, which covers both the $p \leq n$ and $p > n$ cases, and vary the size of the active set over $|\mathcal{A}| \in \{5, 10, 20\}$.

We compare our method against several baselines, including single-marker hypothesis testing (using the Wald test), group lasso (where each group consists of the 3 genotype indicators for one SNP), fGWAS, and BGL-fGWAS. We used several software packages to run these methods: the PLINK toolkit [16] for the Wald test, the SLEP Matlab package [17] for lasso and group lasso, and the fGWAS2 R package [18] for fGWAS and BGL-fGWAS. To run the static data methods (hypothesis test and group lasso), we summarize the phenotype values by averaging across time.

To evaluate performance, we calculate the maximum power attained by each method at a fixed false discovery rate. In order to calculate this metric, we first generate a ranked list of the top $|\mathcal{A}|$ SNPs identified by each method. For the Wald test and fGWAS, this is given by the SNPs with the smallest p-values. For the penalized regression methods, we test a series of values of the

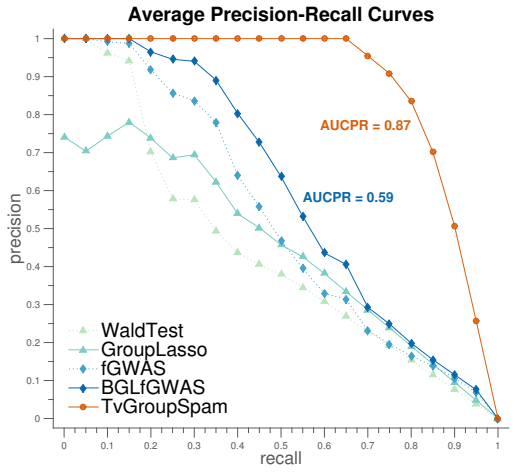


Figure 2.3: Comparison of precision-recall curves of TV-GroupSpAM to baseline methods shows that our approach has an average AUCPR of 0.87 ± 0.01, which is much higher than the most competitive baseline, BGL-fGWAS, which has average AUCPR 0.59 ± 0.02.

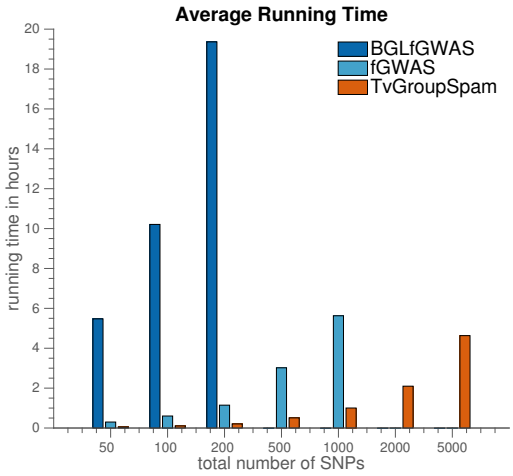


Figure 2.4: Comparison of the running time of TV-GroupSpAM to baseline methods shows that our approach runs much faster than both fGWAS and BGL-fGWAS. We were unable to run fGWAS for $p > 1000$ or BGL-fGWAS for $p > 200$ due to time constraints.

regularization parameter, λ, and select the one that yields approximately the desired number of SNPs. We then rank these SNPs according to their fitted model weights or norms. Given this list, we select a cutoff point that yields the largest set of SNPs such that FDR is below 0.2, and we calculate the power at this threshold.[2] The results of our experiments are shown in Figure 2.2.

Our results indicate that TV-GroupSpAM far outperforms all of the baseline methods in every setting. In many cases, the three dynamic methods are able to detect at least twice as many true associations as the static methods. This underscores the value of leveraging longitudinal data to boost statistical power. The results show that TV-GroupSpAM outperforms fGWAS even when the residual errors are correlated, despite the fact that our model assumes independent errors while fGWAS does not. These results demonstrate that TV-GroupSpAM performs well under many different conditions and is robust to noise.

To obtain a more complete picture of the performance of each method, we plot the precision-recall curves obtained by varying the number of SNPs selected by each method from 0 to p. The average precision-recall curves obtained by averaging results over 20 datasets for the most challenging synthetic data setting ($p = 200$, $|\mathcal{A}| = 20$, correlated errors) are shown in Figure 2.3. We also report the area under the precision recall curve (AUCPR) for BGL-fGWAS and TvGroupSpAM. Our approach outperforms the most competitive baseline by a significant margin. Lastly, we compare the run times of the three dynamic trait methods for different values of p, and show the results in Figure 2.4. For $p = 200$, TvGroupSpAM ran in 12 minutes, fGWAS ran in 69 minutes, and BGL-fGWAS ran in 20 hours. These results show that our method is by far the most computationally efficient.

[2]Note that power is equivalent to recall $= \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$ and FDR is equivalent to $1-\mathrm{precision} = \mathrm{FP}/(\mathrm{TP}+\mathrm{FP})$.

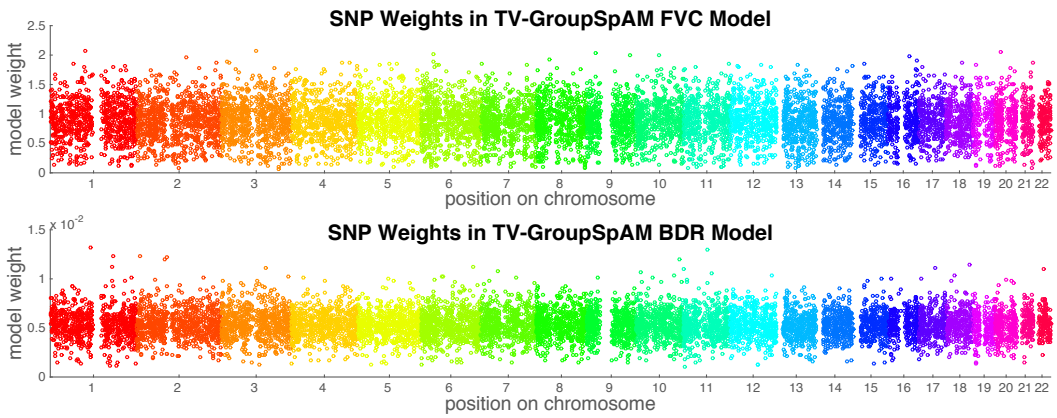


Figure 2.5: Manhattan plots of the model weights for each SNP that was selected in the FVC model (top) and BDR model (bottom) during the filtering stage.

Table 2.1: **Selected SNPs associated with forced vital capacity (FVC)**

SNP	Chrom	Location	Effect Size	Nearby Genes Linked to Asthma
rs6442021	3	46.7 Mb	1.5303	*CCR1, CCR2, CCR3, CCR5* – chemokine receptors in the CC family; CCR2 is a receptor for a protein that plays a role in several inflammatory diseases, and has been directly linked to asthma [19]; CCR3 may play a role in airway inflammation [20] *PRSS42, PRSS46, PRSS45, PRSS50* – trypsin-like serine proteases; tryptases cause bronchoconstriction and have been implicated in asthma [21]
rs2062583	3	56.9 Mb	1.0074	*IL17RD* – interleukin 17 receptor D; IL-17 is a proinflammatory cytokine produced by Th17 cells that plays a role in multiple inflammatory diseases, including asthma [22]
rs1450118	3	190.4 Mb	0.9027	*IL1RAP* – interleukin 1 receptor accessory protein; enables the binding of IL-33 to its receptor encoded by IL1RL1, which has been repeatedly linked to asthma [23]
rs3801148	7	139.3 Mb	0.8538	*TBXAS1* – thromboxane A synthase; this enzyme converts prostaglandin H2 to thromboxane A2, a lipid that constricts respiratory muscle [24]
rs914978	9	132.3 Mb	1.0631	*PTGES* – prostaglandin E synthase; this enzyme converts prostaglandin H2 to prostaglandin E2, a lipid inflammatory mediator that acts in the lung [25]
rs11069178	12	117.9 Mb	0.6869	*NOS1* – nitric oxide synthase 1; nitric oxide affects bronchial tone and its levels are elevated in the air exhaled by asthmatics; NOS1 has been linked to a higher risk of asthma [26]
rs6056242	20	8.8 Mb	1.2298	*PLCB4* – involved in the endothelial cell signaling pathway [27] and plays a role in vascular inflammation [28]

2.3.2 Genome-Wide Association Study of Asthma

Next we use TV-GroupSpAM to perform a genome-wide association analysis of asthma traits. We look for associations between SNPs and two quantitative phenotypes frequently used to assess asthma severity: the forced vital capacity (FVC), a sensitive measure of airway obstruction, and

Table 2.2: **Selected SNPs associated with bronchodilator response (BDR)**

SNP	Chrom	Location	Effect Size	Nearby Genes Linked to Asthma
rs7766818	6	46.8 Mb	0.0088	*GPR116* – probable G protein-coupled receptor 116; plays a critical role in lung surfactant homeostasis [29]
				TNFRSF21 – tumor necrosis factor receptor superfamily member 21; plays a central role in regulating immune response and airway inflammation in mice [30]
rs12524603	6	159.8 Mb	0.0075	*SOD2* – superoxide dismutase 2, mitochondrial; plays a role in oxidative stress, and has been linked to bronchial hyperresponsiveness and COPD [31]
rs13239058	7	139.3 Mb	0.0079	*TBXAS1* – see Table 1 above
rs10519096	15	59.1 Mb	0.0086	*ADAM10* – disintegrin and metalloproteinase domain-containing protein 10; plays an important role in immunoglobulin E dependent lung inflammation [32]
rs8111845	19	41.6 Mb	0.0066	*TGFB1* – transforming growth factor β1; has pro-inflammatory as well as anti-inflammatory properties, and has been linked to asthma and airway remodeling [33]
				CYP2A6, CYP2A7, RAB4B, MIA, EGLND – genes located in a known COPD locus [34]
rs6116189	20	4.0 Mb	0.0067	*ADAM33* – disintegrin and metalloproteinase domain-containing protein 33; has been implicated in asthma by several independent studies [35; 36]
rs6077566	20	9.5 Mb	0.0101	*PLCB4* – see Table 1 above
rs1321715	20	58.8 Mb	0.0061	*CDH26* – cadherin-like 26; has been linked to asthma-related traits [37]

bronchodilator response (BDR), which measures lung response to bronchodilator drugs. For this analysis, we use data from the CAMP longitudinal study of childhood asthma [38] with $n = 552$ subjects genotyped at $p = 510{,}540$ SNPs from across all 22 autosomal chromosomes. After preprocessing, in which we removed subjects with missing data and SNPs with minor allele frequency below 0.05, we were left with $n = 465$ and $p = 509{,}299$. In order to control for non-genetic effects, we incorporated several static covariates into our model, including: sex, race, the age of onset of asthma, the clinic where the patient's traits were measured, and the treatment or control group to which the patient was assigned in the clinical trial associated with the CAMP study.

For computational efficiency, we first used our approach to filter out a relatively small set of SNPs to include in the final analysis for each phenotype. To do this, we split the dataset into 100 subsets, each containing approximately $5,000$ SNPs, and ran TV-GroupSpAM separately on each set. We regulated the model sparsity by using a binary search procedure to identify a value of λ that selected between 90 and 110 SNPs from each subset, following the example of [5]. This yielded a filtered set of 10,118 SNPs for the FVC model and 9,621 SNPs for the BDR model. Figure 2.5 shows the model weight (an indicator of significance) of every SNP that was selected in the filtering step for each phenotype. Next we fit a new global model for each trait using only these selected SNPs, and chose a value of λ that yielded approximately 50 SNPs with nonzero effect on the phenotype (yielding 48 for FVC and 51 for BDR). Finally, we refit the model on just these selected SNPs with no regularization penalty, and use the estimated group functional norms to determine the effect size of each SNP. Note that the FVC effect sizes are much higher in magnitude than the BDR effect sizes because the FVC phenotype is measured in different units than the BDR phenotype.

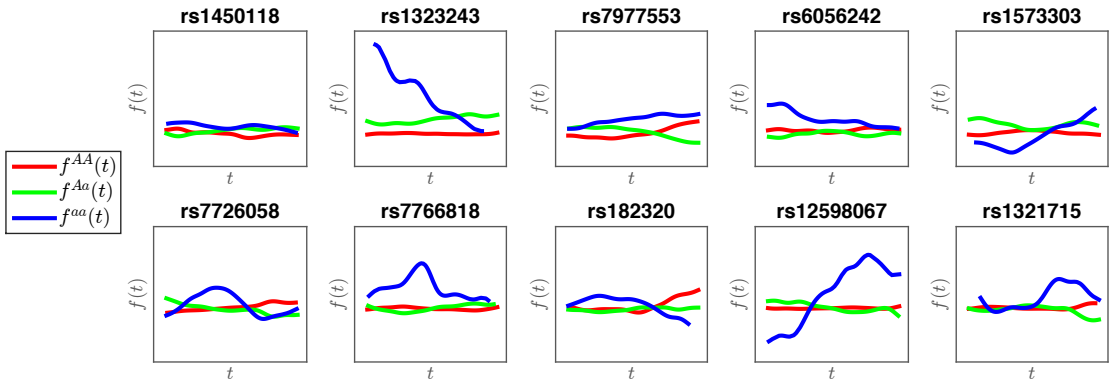


Figure 2.6: Examples of estimated dynamic SNP effects. Top row shows five SNPs selected from FVC model. Bottom row shows five SNPs selected from BDR model.

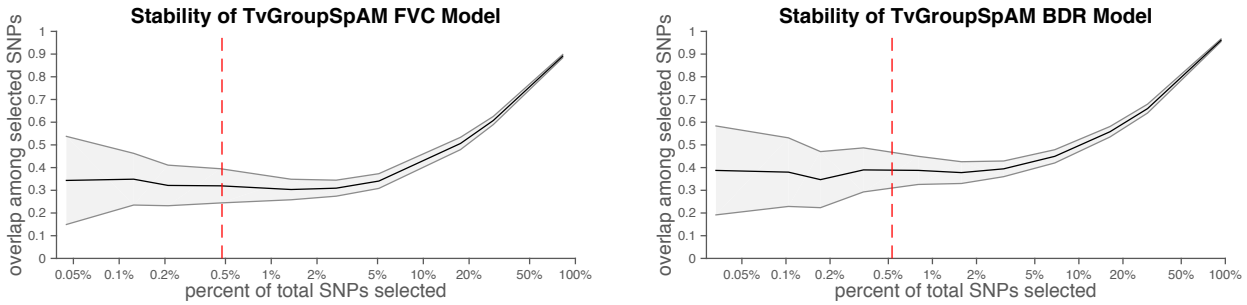


Figure 2.7: Average stability of the FVC model (left) and BDR model (right) for different fractions of selected SNPs. Shaded region shows standard deviation. Red line indicates the fraction of the filtered SNPs that we selected in our final analysis, which yielded 48 SNPs for FVC and 51 SNPs for BDR.

In order to analyze the validity of our results, we identified all genes located within 500 Kb of each SNP in the final selected sets and then determined whether any of the genetic loci or nearby genes are known to be associated with asthma or asthma-related functions in the existing literature. Because asthma is a disease characterized by inflammation and constriction of the airways of the lungs, we specifically searched for genes that have been linked to lung function or inflammatory response. Furthermore, since asthma is partly driven by a series of interactions between vascular endothelial cells and leukocytes [39], we also searched for genes involved in functions of the vascular system or the immune system, particularly those in pathways involving T-helper 2 (Th2) cells, which play a central role in the pathogenesis of asthma [23].

We list a curated subset of the SNPs selected in the FVC and BDR models in Tables 2.1 and 2.2, along with the nearby genes that can be linked to asthma. Our model was able to identify several genetic loci that have a well-established connection to asthma. For example, SNP rs6116189 on chromosome 20 is located near the ADAM33 gene, which has been implicated in asthma by several independent studies [36]. In addition, SNP rs1450118 on chromosome 3 is located near IL1RAP, a gene that produces the Interleukin 1 receptor accessory protein needed for the binding of Interleukin 33 (a member of the Interleukin 1 family) to its receptor encoded by the IL1RL1 gene, which is

known to play an important role in asthma [23]. Finally, the locus on chromosome 7 at 139.3 Mb is particularly interesting because it was selected in both the FVC and BDR models. This SNP is located near the TBXAS1 gene, which encodes Thromboxane-A synthase, an enzyme that is known to play a role in asthma [24]. We plot some examples of the estimated time-varying effects of SNPs selected in our FVC and BDR models in Figure 2.6.

Finally, in order to evaluate the sensitivity of TV-GroupSpAM to noise in the data, we returned to the two filtered sets of $\sim 10,000$ SNPs each and reran the final selection step on multiple 90% subsamples of the data, then analyzed the stability of the set of selected SNPs. Because the stability naturally varies with the total number of SNPs being selected, we ran our algorithm on each subsample for a fixed set of λ values such that the fraction of selected SNPs ranged from 0.5% to nearly 100%. We then calculated the average stability for a particular value of λ as the average pairwise overlap among the selected SNP sets divided by the average number of SNPs selected across all subsamples. We plot the stability as a function of the average percentage of SNPs selected in Figure 2.7, with the shaded region showing the standard deviation of the pairwise stability. These results indicate that the stability of the FVC model when selecting 0.47% of SNPs (48 out of 10,118) is 32% and the stability of the BDR model when selecting 0.53% of SNPs (51 out of 9,621) is 39%.

2.4 Discussion

In this work, we propose a new approach to GWAS that bridges the gap between existing penalized regression methods, such as the lasso and group lasso, and dynamic trait methods, such as fGWAS. Our approach uses penalized regression to identify a sparse set of SNPs that jointly influence a dynamic trait. This is a challenging task for several reasons: first, we must contend with high-dimensional data, which requires that we regularize the model to perform variable selection; second, we do not know the true underlying model by which each SNP acts on the phenotype, and therefore we must avoid making parametric assumptions about these patterns; and third, we assume that SNP effects vary smoothly over time, which means that we cannot apply a standard multi-task regression model that treats the time series as a set of unordered traits.

Although TV-GroupSpAM achieves significantly better performance on synthetic data than existing methods, there are still certain challenging aspects of genome-wide association mapping that are not addressed by this approach. One of these is the task of rare variant detection. Although our method is robust to detecting spurious effects from rare variants, we are also not able to detect true effects from rare variants with high power. This is due to the lack of data available for the aa genotype in SNPs with very low minor allele frequency; because we estimate a separate effect function for each SNP genotype, we are unable to accurately estimate f^{aa} when there are very few data points with this genotype. Modifying TV-GroupSpAM to more accurately detect the effects of rare variants would be an interesting direction for future work.

Chapter 3

Inverse-Covariance Fused Lasso

3.1 Introduction

A critical task in the study of biological systems is understanding how gene expression is regulated within the cell. Although this problem has been studied extensively over the past few decades, it has recently gained momentum due to rapid advancements in techniques for high-throughput data acquisition. Within this task, two problems that have received significant attention in recent years are (a) understanding how various genetic loci regulate gene expression, a problem known as eQTL mapping [40], and (b) determining which have a direct influence on the expression of other genes, a problem known as gene network estimation [41]. Prior work on learning regulatory associations has largely treated eQTL mapping and gene network estimation as completely separate problems.

In this work, we pursue a holistic approach to discovering the patterns of gene regulation in the cell by integrating eQTL mapping and gene network estimation into a single model. Specifically, given a dataset that contains both genotype information for a set of single nucleotide polymorphisms (SNPs) and mRNA expression measurements for a set of genes, we aim to simultaneously learn the SNP-gene and gene-gene relationships. The key element of our approach is that we transfer knowledge between these two tasks in order to yield more accurate solutions to both problems.

In order to share information between tasks, we assume that two genes that are tightly linked in a regulatory network are likely to be associated with similar sets of SNPs in an eQTL map, and vice versa. Our assumption is motivated by the observation that genes participating in the same biological pathway or module are usually co-expressed or co-regulated, and therefore linked in a gene network [42]. Because of this, when the expression of one gene is perturbed, it is likely that the expression of the entire pathway will be affected. In the case of eQTL mapping, this suggests that any genetic locus that is associated with the expression of one gene is likely to influence the expression of the entire subnetwork to which the gene belongs. By explicitly encoding these patterns into our model, we can take advantage of this biological knowledge to boost our statistical power for detecting eQTLs. Ultimately, this allows us to leverage information about gene-gene relationships to learn a more accurate set of eQTL associations, and similarly to leverage information about SNP-gene relationships to learn a more accurate gene network.

Based on these key assumptions, we construct a unified model for this problem by formulating it as a multiple-output regression task in which we jointly estimate the regression coefficients and the inverse covariance structure among the response variables. Specifically, given SNPs $x = (x_1, \ldots, x_p)$ and genes $y = (y_1, \ldots, y_q)$, our goal is to regress y on x and simultaneously estimate the inverse covariance of y. In this model, the matrix of regression coefficients encodes the SNP-gene relationships in the eQTL map, whereas the inverse covariance matrix captures the gene-gene relationships in the gene network. In order to ensure that information is transferred between the

two components of the model, we incorporate a regularization penalty that explicitly encourages pairs of genes that have a high weight in the inverse covariance matrix to also have similar regression coefficient values. This structured penalty enables the two estimates to learn from one another as well as from the data.

3.2 Background

Before presenting our approach, we provide some background on the problems of penalized multiple-output regression and sparse inverse covariance estimation, which will form the building blocks of our unified model.

In what follows, we assume X is an n-by-p dimensional matrix of SNP genotypes, which we also call *inputs*, and Y is an n-by-q dimensional matrix of gene expression values, which we also call *outputs*. Here n is the number of samples, p is the number of SNPs, and q is the number of genes. The element $x_{ij} \in \{0, 1, 2\}$ represents the genotype value of sample i at SNP j, encoded as 0 for two copies of the minor allele, 1 for one copy of the minor allele, and 2 for two copies of the minor allele. Similarly $y_{ik} \in \mathbb{R}$ represents the expression value of sample i in gene k. We assume that the expression values for each gene are mean-centered.

Multiple-Output Lasso. Given input matrix X and output matrix Y, the standard ℓ_1-penalized multiple-output regression problem, also known as the multi-task lasso [43], is given by

$$\min_B \ \tfrac{1}{n}\|Y - XB\|_F^2 + \lambda\|B\|_1 \tag{3.1}$$

where B is a p-by-q dimensional matrix and β_{jk} is the regression coefficient that maps SNP x_j to gene y_k. Here $\|\cdot\|_1$ is an ℓ_1 norm penalty that induces sparsity among the estimated coefficients, and λ is a regularization parameter that controls the degree of sparsity. The objective function given above is derived from the penalized negative log likelihood of a multivariate Gaussian distribution, assuming $y \,|\, x \sim \mathcal{N}(x^T B, \varepsilon^2 I)$ where we let $\varepsilon^2 = 1$ for simplicity. Although this problem is formulated in a multiple-output framework, the ℓ_1 norm penalty merely encourages sparsity, and does not enforce any shared structure between the regression coefficients of different outputs. As a result, the objective function given in (3.1) decomposes into q independent regression problems.

Graph-Guided Fused Lasso. Given a weighted graph $G \in \mathbb{R}^{q \times q}$ that encodes a set of pairwise relationships among the outputs, we can modify the regression problem by imposing an additional fusion penalty that encourages genes y_k and y_m to have similar parameter vectors $\beta_{\cdot k}$ and $\beta_{\cdot m}$ when the weight of the edge connecting them is large. This problem is known as the graph-guided fused lasso [44; 45; 46] and is given by

$$\begin{aligned} \min_B \ & \tfrac{1}{n}\|Y - XB\|_F^2 + \lambda\|B\|_1 \\ & + \gamma \textstyle\sum_{k,m} |g_{km}| \cdot \|\beta_{\cdot k} - \operatorname{sign}(g_{km})\beta_{\cdot m}\|_1 \end{aligned} \tag{3.2}$$

Here the ℓ_1 norm penalty again encourages sparsity in the estimated coefficient matrix. In contrast, the second penalty term, known as a graph-guided fusion penalty, encourages similarity among the regression parameters for all pairs of outputs. The weight of each term in the fusion penalty is dictated by $|g_{km}|$, which encodes the strength of the relationship between y_k and y_m. Furthermore, the sign of g_{km} determines whether to encourage a positive or negative relationship between parameters; if $g_{km} > 0$ (i.e. genes y_k and y_m are positively correlated), then we encourage $\beta_{\cdot k}$ to be

equal to $\beta_{\cdot m}$, but if $g_{km} < 0$ (i.e. genes y_k and y_m are negatively correlated), we encourage $\beta_{\cdot k}$ to be equal to $-\beta_{\cdot m}$. If $g_{km} = 0$, then genes y_k and y_m are unrelated, and so we don't fuse their respective regression coefficients.

Sparse Inverse Covariance Estimation. In the graph-guided fused lasso model defined in (3.2), the graph G must be known ahead of time. However, it is also possible to learn a network over the set of genes. One way to do this is to estimate their pairwise conditional independence relationships. If we assume $y \sim \mathcal{N}(\mu, \Sigma)$, where we let $\mu = 0$ for simplicity, then these conditional independencies are encoded in the inverse covariance matrix, or precision matrix, defined as $\Theta = \Sigma^{-1}$. We can obtain a sparse estimate of the precision matrix using the graphical lasso [47] given by

$$\min_{\Theta} \ \tfrac{1}{n} \operatorname{tr}(Y^T Y \, \Theta) - \log\det(\Theta) + \lambda \|\Theta\|_1 \tag{3.3}$$

This objective is again derived from the penalized negative log likelihood of a Gaussian distribution, where this time the ℓ_1 penalty term encourages sparsity among the entries of the precision matrix.

3.3 Method

We now introduce a new approach for jointly estimating the coefficients in a multiple-output regression problem and the edges of a network over the regression outputs. We apply this technique to the problem of simultaneously learning an eQTL map and a gene regulatory network from genome (SNP) data and transcriptome (gene expression) data. Although we focus exclusively on this application, the same problem formulation appears in other domains as well.

3.3.1 Joint Regression and Network Estimation Model

Given SNPs $x \in \mathbb{R}^p$ and genes $y \in \mathbb{R}^q$, in order to jointly model the n-by-p regression parameter matrix B and the q-by-q inverse covariance matrix Θ, we begin with two core modeling assumptions,

$$x \sim \mathcal{N}(0, T) \tag{3.4}$$

$$y \,|\, x \sim \mathcal{N}(x^T B, E) \tag{3.5}$$

where T is the covariance of x and E is the conditional covariance of $y \,|\, x$. Given the above model, we can also derive the marginal distribution of y. To do this, we first use the fact that the marginal distribution $p(y)$ is Gaussian.[1] We can then derive the mean and covariance of y, as follows.

$$\begin{aligned}
\mathbb{E}_y(y) &= \mathbb{E}_x(\mathbb{E}_{y|x}(y|x)) = 0 \\
\mathbb{C}\mathrm{ov}_y(y) &= \mathbb{E}_x(\mathbb{C}\mathrm{ov}_{y|x}(y|x)) + \mathbb{C}\mathrm{ov}_x(\mathbb{E}_{y|x}(y|x)) = E + B^T T B
\end{aligned}$$

Using these facts, we conclude that the distribution of y is given by

$$y \sim \mathcal{N}(0, \Theta^{-1}) \tag{3.6}$$

where $\Theta^{-1} = E + B^T T B$ denotes the marginal covariance of y. This allows us to explicitly relate Θ, the inverse covariance of y, to B, the matrix of regression parameters. Lastly, we simplify our

[1] See Equation B.44 of Appendix B in [48].

model by assuming $T = \tau^2 I_{p\times p}$ and $E = \varepsilon^2 I_{q\times q}$. With this change, the relationship between B and Θ^{-1} can be summarized as $\Theta^{-1} \propto B^T B$ because B is now the only term that contributes to the off-diagonal entries of Θ and hence to the inverse covariance structure among the genes.[2]

3.3.2 Estimating Model Parameters with a Fusion Penalty

Now that we have a model that captures B and Θ, we want to jointly estimate these parameters from the data while encouraging the relationship $\Theta^{-1} \propto B^T B$. To do this, we formulate our model as a convex optimization problem with an objective function of the form

$$\text{loss}_{y|x}(B) + \text{loss}_y(\Theta) + \text{penalty}(B, \Theta) \tag{3.7}$$

where $\text{loss}_{y|x}(B)$ is a loss function derived from the negative log likelihood of $y \mid x$, $\text{loss}_y(\Theta)$ is a loss function derived from the negative log likelihood of y, and $\text{penalty}(B, -\Theta)$ is a penalty term that encourages shared structure between the estimates of B and Θ.

Given n i.i.d. observations of x and y, let X be a matrix that contains one observation of x per row and let Y be a matrix that contains one observation of y per row. Then we define the inverse covariance fused lasso (ICLasso) optimization problem as

$$\begin{aligned} \min_{B,\Theta} \; & \tfrac{1}{n}\|Y - XB\|_F^2 + \tfrac{1}{n}\operatorname{tr}(Y^T Y\,\Theta) - \log\det(\Theta) \\ & + \lambda_1 \|B\|_1 + \lambda_2 \|\Theta\|_1 \\ & + \gamma \textstyle\sum_{k,m} |\theta_{km}| \cdot \|\beta_{\cdot k} + \operatorname{sign}(\theta_{km})\beta_{\cdot m}\|_1 \end{aligned} \tag{3.8}$$

From a statistical perspective, the above formulation is unusual because we aim to simultaneously optimize the marginal and conditional likelihood functions of y. However, when we consider it simply as an optimization problem and divorce it from the underlying model, we see that it boils down to a combination of the objectives from the multiple-output lasso and the graphical lasso problems, with the addition of a graph-guided fused lasso penalty to encourage transfer learning between the estimates of B and Θ.

When Θ is fixed, our objective reduces to the graph-guided fused lasso with the graph given by $G = -\Theta$. When B is fixed, our objective reduces to a variant of the graphical lasso in which the ℓ_1 norm penalty has a different weight for each element of the inverse covariance matrix, i.e. the standard penalty term $p(\Theta) = \lambda \sum_{k,m} |\theta_{km}|$ is replaced by $p(\Theta) = \sum_{k,m} w_{km}|\theta_{km}|$ where the weights are given by $w_{km} = \lambda_2 + \gamma\|\beta_{\cdot k} + \operatorname{sign}(\theta_{km})\beta_{\cdot m}\|$.

We further deconstruct the ICLasso objective by describing the role of each term in the model:

- The first term $\frac{1}{n}\|Y - XB\|_F^2$ is the regression loss, and is derived from the conditional log likelihood of $y \mid x$. Its role is to encourage the coefficients B to map X to Y, i.e. to obtain a good estimate of the eQTL map from the data.

- The second term $\frac{1}{n}\operatorname{tr}(Y^T Y\,\Theta) - \log\det(\Theta)$ is the inverse covariance loss, and is derived from the marginal log likelihood of y. Its role is to encourage the network Θ to reflect the partial correlations among the outputs, i.e. to obtain a good estimate of the gene network from the data.

[2] Although we make this simplifying assumption in our model, we later demonstrate via simulation experiments that ICLasso still performs well in practice when these constraints are violated, namely when the dimensions of x are not independent and the dimensions of y have residual covariance structure once the effect of $x^T B$ is removed.

- The third term $\lambda_1\|B\|_1$ is an ℓ_1 norm penalty over the matrix of regression coefficients that induces sparsity among the SNP-gene interactions encoded in B.
- The fourth term $\lambda_2\|\Theta\|_1$ is an ℓ_1 norm penalty over the precision matrix that induces sparsity among the gene-gene interactions encoded in Θ.
- The final term $\gamma\sum_{k,m}|\theta_{km}|\cdot\|\beta_{\cdot k}+\text{sign}(\theta_{km})\beta_{\cdot m}\|_1$ is a graph-guided fusion penalty that encourages similarity between the coefficients of closely related outputs; specifically, when genes y_k and y_m have a positive partial correlation, it fuses β_{jk} towards β_{jm} for all SNPs x_j, and when genes y_k and y_m have a negative partial correlation, it fuses β_{jk} towards $-\beta_{jm}$ for all SNPs x_j.[3]

In the above objective, the loss functions come directly out of the modeling assumptions given in (3.5) and (3.6). The sparsity-inducing ℓ_1 norm penalties make estimation feasible in the high-dimensional setting where $p, q > n$, and contribute to the interpretability of the eQTL map and gene network.

3.3.3 Sparse Structure in B and Θ

In this section, we describe how the ICLasso model captures sparse structure that is shared between the eQTL map B and the gene network Θ and in doing so enables transfer learning.

We first prove that the graph-guided fused lasso penalty encourages the structure $\Theta^{-1} \propto B^TB$, thereby linking the two estimates. Consider the optimization problem $\hat{\Theta} = \arg\min_\Theta f(\Theta) \equiv \text{tr}(B^TB\Theta) - \log\det(\Theta)$. We can solve this problem in closed form by taking the gradient $\nabla_\Theta f(\Theta) = B^TB - \Theta^{-1}$ and setting it to 0, which yields the solution $\hat{\Theta}^{-1} = B^TB$. This suggests that the penalty $\text{tr}(B^TB\Theta)$ encourages the desired structure, while the log determinant term enforces the constraint that Θ be positive semidefinite, which is necessary for Θ to be a valid inverse covariance matrix.

However, instead of directly using this penalty in our model, we demonstrate that it encourages similar structure as the graph-guided fused lasso penalty. We compare the trace penalty, denoted TRP, and the graph-guided fused lasso penalty, denoted GFL, below.

$$\text{TRP}(B, \Theta) = \text{tr}(B^TB\Theta) = \sum_{k=1}^{q}\sum_{m=1}^{q}\theta_{km}\cdot\beta_{\cdot k}^T\beta_{\cdot m} \tag{3.9}$$

$$\text{GFL}(B, -\Theta) = \sum_{k=1}^{q}\sum_{m=1}^{q}|\theta_{km}|\cdot\|\beta_{\cdot k}+\text{sign}(\theta_{km})\beta_{\cdot m}\|_1 \tag{3.10}$$

We show that these penalties are closely related by considering three cases.

- When $\theta_{km} = 0$, the relevant terms in both TRP and GFL go to zero. In this case, nothing links $\beta_{\cdot k}$ and $\beta_{\cdot m}$ in either penalty.
- When $\theta_{km} < 0$, the relevant term in TRP is minimized when $\beta_{\cdot k}^T\beta_{\cdot m}$ is large and positive, which occurs when $\beta_{\cdot k}$ and $\beta_{\cdot m}$ point in the same direction. Similarly, the corresponding term in GFL is minimized when $\beta_{\cdot k} = \beta_{\cdot m}$. In this case, both penalties

[3]Note that θ_{km} is negatively proportional to the partial correlation between y_k and y_m, meaning that a negative value of θ_{km} indicates a positive partial correlation and vice versa (see, e.g., [49]). This explains why the sign is flipped in the fusion penalty in (3.8) relative to the one in (3.2).

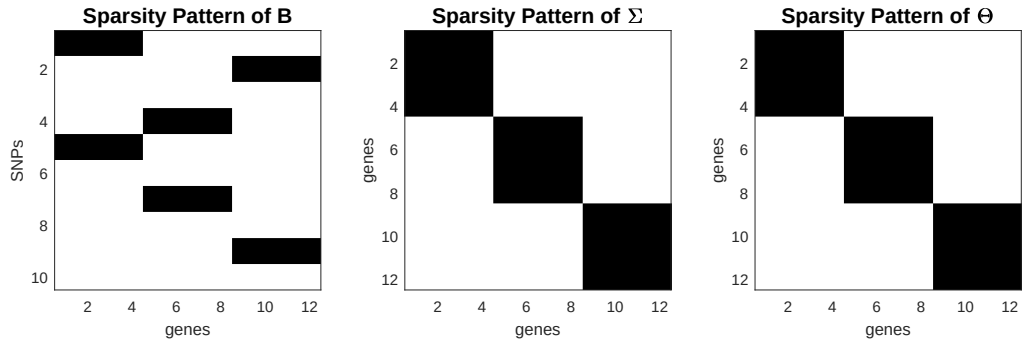


Figure 3.1: A toy example with 10 SNPs and 12 genes grouped into 3 modules. When B exhibits a certain type of sparse structure, $\Sigma = I + B^T B$ and $\Theta = \Sigma^{-1}$ will also be sparse.

encourage similarity between $\beta_{\cdot k}$ and $\beta_{\cdot m}$ with strength proportional to the magnitude of θ_{km}.

- When $\theta_{km} > 0$, the relevant term in TRP is minimized when $\beta_{\cdot k}^T \beta_{\cdot m}$ is large and negative, which occurs when $\beta_{\cdot k}$ and $\beta_{\cdot m}$ point in opposite directions. Similarly, the corresponding term in GFL is minimized when $\beta_{\cdot k} = -\beta_{\cdot m}$. In this case, both penalties encourage similarity between $\beta_{\cdot k}$ and $-\beta_{\cdot m}$ with strength proportional to the magnitude of θ_{km}.

We choose to use the graph-guided fused lasso penalty instead of the trace penalty because it more strictly enforces the relationship between B and Θ^{-1} by fusing the regression parameter values of highly correlated genes.

Next, we describe a set of conditions under which our assumptions on B and Θ are compatible with one another. Although we do not provide theoretical guarantees on what type of structure will be learned by our method, we illustrate via a toy example that certain biologically realistic scenarios will naturally lead to sparsity in both B and $\Theta = (B^T B)^{-1}$.

Consider a gene network that is organized into a set of densely connected sub-networks corresponding to functional gene modules (e.g., pathways). In this case, we might expect the true Θ to be block diagonal, meaning that there exist blocks $C_1, \ldots, C_d$ such that any pair of genes belonging to two different blocks are not connected in the gene network, i.e. $\theta_{km} = 0$ for any $y_k \in C_a$ and $y_m \notin C_a$. Furthermore, suppose our central assumption on the relationship between B and Θ is satisfied, namely genes that are linked in the gene network are associated with similar sets of SNPs in the eQTL map. Then we might expect that any pair of genes belonging to the same block will have the same SNP-gene associations, i.e. $\beta_{jk} = \beta_{jm}\ \forall j$ for any $y_k, y_m \in C_a$. Since we also assume that the true B is sparse, this would lead to a block sparse pattern in B in which each gene module is associated with only a subset of the SNPs.

A simple of example of this type of sparse structure is shown in Figure 3.1. Note that such a pattern in B would lead to block diagonal structure in $\Sigma = I + B^T B$ that preserves the blocks defined by $C_1, \ldots, C_d$. Furthermore, since the inverse of a block diagonal matrix is also block diagonal with the same blocks, this implies that $\Theta = \Sigma^{-1} = (I + B^T B)^{-1}$ will be block diagonal with blocks $C_1, \ldots, C_d$.

This provides an example of a scenario that occurs naturally in biological networks and satisfies our modeling assumptions. However, we note that our model is flexible enough to handle other types of sparse structure as well. In fact, one of the main advantages of our approach is that the sparsity pattern is learned from the data rather than specified in advanced.

3.3.4 Relationship to Other Methods

There are currently two existing approaches that jointly estimate regression coefficients and network structure: multivariate regression with covariance estimation (MRCE), from [50], and conditional Gaussian graphical models (CGGM), originally from [51] and further developed by [52] and [53]. In this section, we describe how our approach differs from these others.

All three methods, including ours, assume that the inputs X and outputs Y are related according to the basic linear model $Y = XB + E$, where E is a matrix of Gaussian noise. However, each approach imposes a different set of additional assumptions on top of this, which we discuss below.

MRCE. This method assumes that $E \sim \mathcal{N}(0, \Omega^{-1})$, which leads to $Y \mid X \sim \mathcal{N}(XB, \Omega^{-1})$. MRCE estimates B and Ω by solving the following objective:

$$\begin{aligned} \min_{B,\Omega} \tfrac{1}{n} \operatorname{tr}((Y - XB)^T (Y - XB)\,\Omega) \\ - \log\det(\Omega) + \lambda_1 \|B\|_1 + \lambda_2 \|\Omega\|_1 \end{aligned} \tag{3.11}$$

It's very important to note that Ω is the conditional inverse covariance of $Y \mid X$, which actually corresponds to the inverse covariance of the noise matrix E rather than the inverse covariance of the output matrix Y. We therefore argue that Ω doesn't capture any patterns that are shared with the regression coefficients B, since by definition Ω encodes the structure in Y that cannot be explained by XB.

CGGM. This approach makes an initial assumption that X and Y are jointly Gaussian with the following distribution:

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Gamma & \Lambda \\ \Lambda^T & \Omega \end{bmatrix} \right)$$

In this formulation, the distribution of $Y \mid X$ is given by $\mathcal{N}(-X\Lambda\Omega^{-1}, \Omega^{-1})$. This corresponds to the reparameterization of B as $-\Lambda\Omega^{-1}$, where Ω is the conditional inverse covariance matrix and Λ represents the "direct" influence of X on Y. CGGM estimates Λ and Ω by solving the following optimization problem, where sparsity penalties are applied to Λ and Ω instead of B and Ω as was the case in (3.11):

$$\begin{aligned} \min_{\Lambda,\Omega} \tfrac{1}{n} \operatorname{tr}((Y + X\Lambda\Omega^{-1})^T (Y + X\Lambda\Omega^{-1})\,\Omega) \\ - \log\det(\Omega) + \lambda_1 \|\Lambda\|_1 + \lambda_2 \|\Omega\|_1 \end{aligned} \tag{3.12}$$

Here the meaning of Ω has not changed, and it once again represents the inverse covariance of the noise matrix.

ICLasso. Our method implicitly assumes two underlying models: $Y \mid X \sim \mathcal{N}(XB, I)$ and $Y \sim \mathcal{N}(0, \Theta^{-1})$. In this case, Θ represents the marginal inverse covariance of Y rather than the conditional inverse covariance of $Y \mid X$, which was captured by Ω in (3.11) and (3.12). The optimization problem in (3.8) is obtained by combining the loss functions derived from the log likelihood of each model and then incorporating sparsity penalties over B and Θ and an additional graph-guided fusion penalty to encourage shared structure.

Both MRCE and CGGM have two important drawbacks that are not shared by our approach. First, both of these methods estimate Ω, the precision matrix of the noise term, rather than Θ, the precision matrix of the outputs Y. Second, neither method incorporates a structured

sparsity penalty that explicitly encourages shared structure between the network and the regression coefficients. In fact, it would not make sense for these methods to apply a joint penalty over B and Ω because, as discussed above, we wouldn't expect these parameters to have any shared structure. By comparison, our method learns the true output network Θ and uses a graph-guided fused lasso penalty to explicitly encourage outputs that are closely related in Θ to have similar parameter values in B.

3.3.5 Optimization via Alternating Minimization

Finally, we present an efficient algorithm to solve the inverse-covariance fused lasso problem defined in (3.8). We start by rewriting the fusion penalty as follows:

$$\begin{aligned}\mathrm{GFL}(B, -\Theta) &= \gamma \textstyle\sum_{k,m} |\theta_{km}| \cdot \|\beta_{\cdot k} + \mathrm{sign}(\theta_{km})\beta_{\cdot m}\|_1 \\ &= \gamma \textstyle\sum_{k,m} \max\{\theta_{km}, 0\} \cdot \|\beta_{\cdot k} + \beta_{\cdot m}\|_1 \\ &\qquad + \gamma \textstyle\sum_{k,m} \max\{-\theta_{km}, 0\} \cdot \|\beta_{\cdot k} - \beta_{\cdot m}\|_1,\end{aligned}$$

from which it is clear that GFL is biconvex in B and Θ. Thus, upon defining

$$\begin{aligned}g(B) &= \tfrac{1}{n}\|Y - XB\|_F^2 + \lambda_1\|B\|_1 \\ h(\Theta) &= \tfrac{1}{n}\,\mathrm{tr}(Y^T Y\,\Theta) - \log\det(\Theta) + \lambda_2\|\Theta\|_1,\end{aligned}$$

we can rewrite the original optimization problem as

$$\min_{B,\Theta}\ g(B) + h(\Theta) + \mathrm{GFL}(B, -\Theta). \tag{3.13}$$

Here $g(B)$ is the usual lasso formulation in (3.1), $h(\Theta)$ is the usual graphical lasso formulation in (3.3), and the graph-guided fusion penalty couples the two problems. Since GFL is biconvex, we can solve the joint problem (3.13) using an alternating minimization strategy. Next we describe how we leverage and extend state-of-the-art convex optimization routines to solve each sub-problem.

Fix Θ, Minimize B. When Θ is fixed, minimizing the objective over B reduces to the well-known graph-guided fused lasso problem,

$$f_\Theta(B) = g(B) + \mathrm{GFL}(B, -\Theta), \tag{3.14}$$

which we optimize using the proximal-average proximal gradient descent (PA-PG) algorithm from [54]. This algorithm is very simple. On each iteration, we first take a gradient step of the form $B - \eta X^\top(XB - Y)$ using some small step size η. Then we compute the weighted average of the component proximal operators for each pair of outputs, where the prox that corresponds to pair (k, m) is given by:

$$\hat{B} = \arg\min_B\ \tfrac{1}{2\eta}\|B - Z\|_F^2 + \|\beta_{\cdot k} + \mathrm{sgn}(\theta_{km})\beta_{\cdot m}\|_1 \tag{3.15}$$

and the weight of this term is given by $|\theta_{km}|/\theta_{\mathrm{tot}}$ where $\theta_{\mathrm{tot}} = \sum_{k,m}|\theta_{k,m}|$. Due to the separability of (3.15) over the rows of B, we can solve for each $\beta_{j\cdot}$ independently. Furthermore, it's clear that for any $i \notin \{k, m\}$, we have $\beta_{ji} = z_{ji}$. Solving for the remaining elements β_{jk} and β_{jm} leads to the following two-dimensional subproblem:

$$\begin{aligned}\hat{\beta}_{jk}, \hat{\beta}_{jm} = \arg\min_{\beta_{jk},\beta_{jm}}\ &\tfrac{1}{2\eta}(\beta_{jk} - z_{jk})^2 \\ &+ (\beta_{jm} - z_{jm})^2 + |\beta_{jk} + \mathrm{sgn}(\theta_{km})\beta_{jm}|.\end{aligned} \tag{3.16}$$

Algorithm 3.1 PA-PG for Graph-Guided Fused Lasso

input: data X, Y, graph Θ, step size η
initialize: $B = 0$
repeat
  $B \leftarrow B - \eta X^\top (XB - Y)$
  for each edge (k, m) with $\theta_{km} \neq 0$ **do**
    $d_{km} \leftarrow \beta_{\cdot k} + \text{sign}(\theta_{km})\beta_{\cdot m}$
    $\beta_{\cdot k} \leftarrow \beta_{\cdot k} - (\theta_{km}/\theta_{\text{tot}}) \cdot \min\{\eta, \frac{1}{2}|d_{km}|\}$
    $\beta_{\cdot m} \leftarrow \beta_{\cdot m} - (\theta_{km}/\theta_{\text{tot}}) \cdot \min\{\eta, \frac{1}{2}|d_{km}|\}$
  end for
until convergence

which can be solved in closed form. Therefore the full solution to the prox operator can be written compactly as follows, where $d_{km} = z_{\cdot k} + \text{sign}(\theta_{km}) z_{\cdot m}$.

$$\begin{aligned}
\hat{\beta}_{\cdot i} &= z_{\cdot i} \quad \text{for } i \notin \{k, m\} \\
\hat{\beta}_{\cdot k} &= z_{\cdot k} - \text{sign}(d_{km}) \cdot \min\{\eta, \tfrac{1}{2}\,|d_{km}|\} \\
\hat{\beta}_{\cdot m} &= z_{\cdot m} - \text{sign}(\theta_{km}) \cdot \text{sign}(d_{km}) \cdot \min\{\eta, \tfrac{1}{2}\,|d_{km}|\}
\end{aligned}$$

From these formulas, we can see that β_{jk} and $-\text{sign}(\theta_{km})\beta_{jm}$ are always "fused" towards each other. For example, when $\text{sign}(\theta_{km}) < 0$, we want to push β_{jk} and β_{jm} towards the same value. In this case, the larger of z_{jk} and z_{jm} will be decremented and the smaller value will be incremented by the same quantity.

We summarize this procedure in Algorithm 3.1. In practice, we use the accelerated version of the algorithm, PA-APG. Using the argument from [54], we can prove that this accelerated algorithm converges to an ϵ-optimal solution in at most $O(1/\epsilon)$ steps, which is significantly better than the $O(1/\sqrt{\epsilon})$ converge rate of subgradient descent.

Fix B, Minimize Θ. When B is fixed, minimizing the objective over Θ reduces to a variation of the well-known graphical lasso problem,

$$f_B(\Theta) = h(\Theta) + \text{GFL}(B, -\Theta), \tag{3.17}$$

which can be optimized by adapting the block coordinate descent (BCD) algorithm of [47]. Indeed, we can rewrite the objective by introducing two $q \times q$ dimensional coefficient matrices U and L whose elements are defined as

$$U_{km} = \tfrac{1}{n} Y_{\cdot k}^\top Y_{\cdot m} + \lambda_2 + \gamma \|\beta_{\cdot k} + \beta_{\cdot m}\|_1 \tag{3.18}$$

$$L_{km} = \tfrac{1}{n} Y_{\cdot k}^\top Y_{\cdot m} - \lambda_2 - \gamma \|\beta_{\cdot k} - \beta_{\cdot m}\|_1. \tag{3.19}$$

Using this notation, we collect all linear terms involving $\Theta_+ := \max\{\Theta, 0\}$ and $\Theta_- := \max\{-\Theta, 0\}$ and reformulate the objective given in (3.17) as

$$\min_{\Theta} -\log\det(\Theta) + \langle \Theta_+, U \rangle - \langle \Theta_-, L \rangle\,. \tag{3.20}$$

The graphical lasso is a special case of the above problem in which $U = L$. In our case, U and L differ because of the structure of the GFL penalty. Nevertheless, we can derive a block coordinate algorithm for this more general setting.

Algorithm 3.2 BCD for the Generalized Graphical Lasso

input: sample covariance matrix $S = \frac{1}{n}Y^TY$, coefficient matrices U, L
initialize: $\Xi = U$
repeat
  for $j = 1$ **to** q **do**
    $\xi \leftarrow \Xi_{\setminus j,j}$, $u \leftarrow U_{\setminus j,j}$, $l \leftarrow L_{\setminus j,j}$, $\tilde{\Xi} \leftarrow \Xi_{\setminus j,\setminus j}$
    $\alpha = 0$
    repeat
      for $j = 1$ **to** $q - 1$ **do**
        $\delta = \tilde{\Xi}_{jj}\alpha_j + \sum_{k \neq j} \tilde{\Xi}_{jk}\alpha_k$
        if $\delta \geq -l_j$ **then** $\alpha_j = (-\delta - l_j)/\tilde{\Xi}_{jj}$
        else if $\delta \leq -u_j$ **then** $\alpha_j = (-\delta - u_j)/\tilde{\Xi}_{jj}$
        else $\alpha_j = 0$
      end for
    until convergence
    $\Xi_{\setminus j,\setminus j} = -\tilde{\Xi}\,\alpha$
  end for
until convergence
$\Theta = \Xi^{-1}$

First we dualize (3.20) to get the following problem:

$$\max_{L \leq \Xi \leq U} \log\det \Xi. \tag{3.21}$$

where $\Theta = \Xi^{-1}$. Then it can be shown that the diagonal of the covariance Ξ must attain the upper bound, i.e. we must have $\Xi_{jj} = U_{jj}\ \forall\ j = 1, \ldots, q$. Next, we perform block coordinate descent by cycling through each column (or row, due to symmetry) of Ξ. We denoted an arbitrary column of Ξ by ξ_j, with corresponding columns u_j and ℓ_j in U and L, respectively. Let $\tilde{\Xi}_j$ be the submatrix of Ξ obtained by deleting column j and row j. Then, by applying Schur's complement, maximizing (3.21) with respect to ξ_j with all other columns fixed amounts to:

$$\min_{\ell_j \leq \xi_j \leq u_j} \ \tfrac{1}{2}\xi_j^\top \tilde{\Xi}_j^{-1}\xi_j. \tag{3.22}$$

Dualizing again, with $\xi_j = -\tilde{\Xi}_j\alpha$, we obtain

$$\min_{\alpha} \ \tfrac{1}{2}\alpha^\top \tilde{\Xi}_j\alpha + u^\top\alpha_+ - \ell^\top\alpha_-, \tag{3.23}$$

which is essentially a lasso problem that we can solve using any known algorithm. We outline the procedure for solving (3.17) in Algorithm 3.2. We use coordinate descent and apply a variant of the soft-thresholding operator to solve for each coordinate. This algorithm converges very quickly because there is no tuning of the step size, and each iteration involves only a matrix-vector product.

3.4 Experiments

In this section, we present the results from a series of experiments on both synthetic and real data. We compare our method to several baselines and demonstrate that it achieves better recovery of the underlying structure of B and Θ than existing methods.

3.4.1 Simulation Study

We begin by evaluating our model on synthetic data so that we can directly measure how accurately the sparse structure of the eQTL map and the gene network are recovered. We compare our eQTL map estimates $\hat{B}$ to several baselines, including standard multi-task lasso (Lasso), graph-guided fused lasso using a sparse covariance matrix as its graph (GFLasso1), graph-guided fused lasso using a sparse precision matrix as its graph (GFLasso2), sparse multivariate regression with covariance estimation (MRCE), and the conditional Gaussian graphical model (CGGM). We compare our network estimates $\hat{\Theta}$ to the graphical lasso (GLasso).

For each method except GLasso, we select hyperparameter values via a two-step procedure in which we first refit B using only the selected inputs $x_j : \beta_{jk} \neq 0$ for each output y_k, and then choose the hyperparameter setting that minimizes the prediction error from the regression model on a held-out validation set. Since GLasso does not produce an estimate of B, we choose the value of λ that minimizes the graphical lasso loss on the validation set.

In our synthetic data experiments, we focus on recovering block-structured networks in which the genes are divided into a set of modules, or groups. In order to generate data according to our model, we assume that the genes within each module only regulate one another and are associated with the same set of eQTLs. Specifically, this means that if genes k and m belong to the same module, we will have $\theta_{km} \neq 0$ and $\beta_{\cdot k} \approx \beta_{\cdot m}$. Although we focus on this data setting because it makes intuitive biological sense and satisfies our modeling assumptions, we note that our approach is flexible enough to handle other types of structure among the SNPs and genes.

We generate synthetic data according to the following procedure. Given sample size n, input dimensionality p, and output dimensionality q, we first fix the number of modules in the gene network, g, and the number of SNPs that each gene will be associated with, s. Note that the density of the true B will be given by s/p and the density of the true Θ will be given by g/q. Next we fix the sparsity pattern in the eQTL map and gene network by randomly assigning each gene to one of the g modules and then selecting a random set of s SNPs that will be associated with each module.

Given the sparsity structure, we generate the parameters of the nonzero values of B and Θ as follows. For each module, we randomly designate a primary gene in that module and generate its association strengths according to $\beta_{jk} \sim \text{Uniform}(0.2, 0.8)$ for each SNP x_j with which the primary gene y_k is associated. Next, for all other genes y_m that belong to the same module as y_k, we draw $\beta_{jm} \sim \mathcal{N}(\beta_{jk}, \rho^2)$ for the same set of SNPs, where $\rho = 0.1$ is a small standard deviation. Lastly, assuming the covariance matrices E and T are given, we generate Θ by setting it equal to $(E + B^T T B)^{-1}$ and then zeroing out all entries that correspond to pairs of genes belonging to different modules.

In order to investigate a wide range of data scenarios, we consider four different settings of E and T in our experiments. These are:

0. $T = I_{p \times p}$ and $E = I_{q \times q}$
1. $T \neq I_{p \times p}$ and $E = I_{q \times q}$
2. $T = I_{p \times p}$ and $E \neq I_{q \times q}$
3. $T \neq I_{p \times p}$ and $E \neq I_{q \times q}$

To generate the non-identity covariance matrices with a random covariance pattern, we first set the element at position j, k equal to $0.7^{|j-k|}$ and then we randomly reshuffle the rows and columns (using the same shuffling for rows and columns to maintain symmetry).

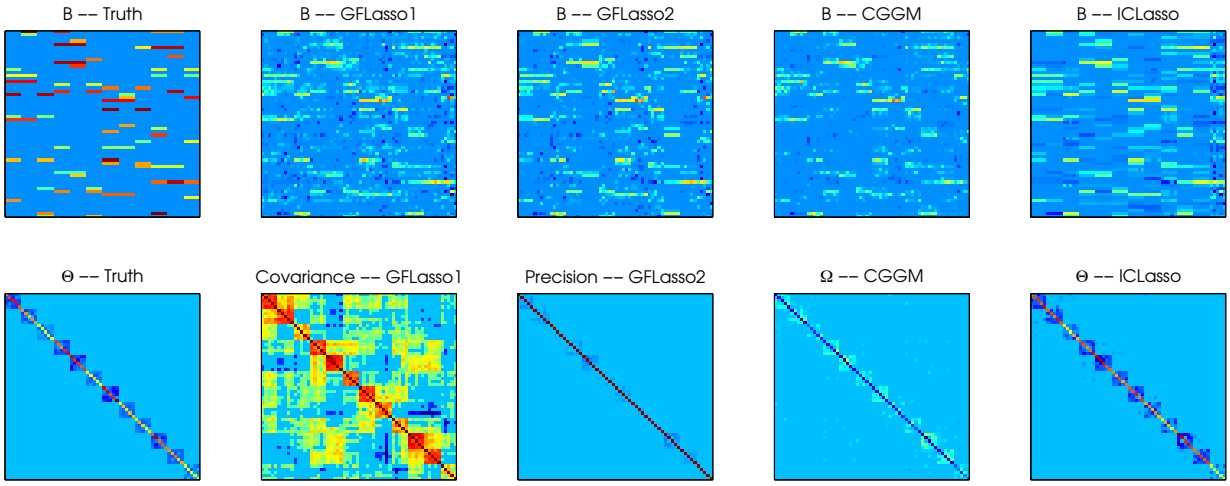


Figure 3.2: A comparison of results on a single synthetic dataset with $p = 60$ and $q = 60$. The far left panel contains the ground truth for B and Θ. The remaining panels show the estimates of the regression coefficients for each method (top) along with the graph structure that was used or estimated by the method (bottom).

Finally, once we have fixed all of the model parameters, we generate the data according to $X \sim \mathcal{N}(0, \tau^2 T)$ and $Y|X \sim \mathcal{N}(X^T B, \epsilon^2 E)$. In all of the experiments that follow, we fix $n = 100$ and use the same sample size for the training, validation, and test sets.

A synthetic data example is shown in Figure 3.2. The ground truth for both B and Θ is given in the far left panel. The next three columns show the estimated values of B for three competing methods, and the results of our method are shown on the far right. In this example, the drawbacks of each of the baseline methods are evident. The covariance matrix used for the network structure in GFLasso1 captures many spurious patterns in Y that don't correspond to true patterns in the regression map, which confuses the estimate of B. The precision matrix used for the network structure in GFLasso2 does not accurately capture the true inverse covariance structure because of the low signal-to-noise ration in Y. This prevents the fusion penalty from effectively influencing the estimate of B. Finally, although CGGM gets a reasonable estimate of the network, despite the fact that it learns the conditional inverse covariance Ω instead of the marginal inverse covariance Θ, this structure is not explicitly enforced in B, which still leads to a poor estimate of the regression parameters. In contrast, the cleanest estimate of both $\hat{B}$ and $\hat{\Theta}$ comes from ICLasso.

The main results of our synthetic experiments are shown in Figures 3.3 and 3.4. We evaluate our approach according to three metrics. In the top two rows, we show the F1 score on the recovery of the true nonzero elements of B and Θ, respectively. This reflects the ability of each method to learn the correct structure of the eQTL map and the gene network. In the bottom row, we show the prediction error of Y on an out-of-sample test set. We note that this test set is completely separate from both the training set (used to estimate the model parameters) and the validation set (used to select the best values of the hyperparameters).

In our first experiment, we jointly vary the number of SNPs and genes, keeping their ratio fixed. We show that ICLasso achieves the best performance even when we violate our modeling assumptions by introducing covariance among the SNPs, introducing conditional covariance among the genes, or both. In our second experiment, we vary the density of B, the density of Θ, and the number of SNPs while keeping the number of genes fixed. Our results clearly demonstrate that

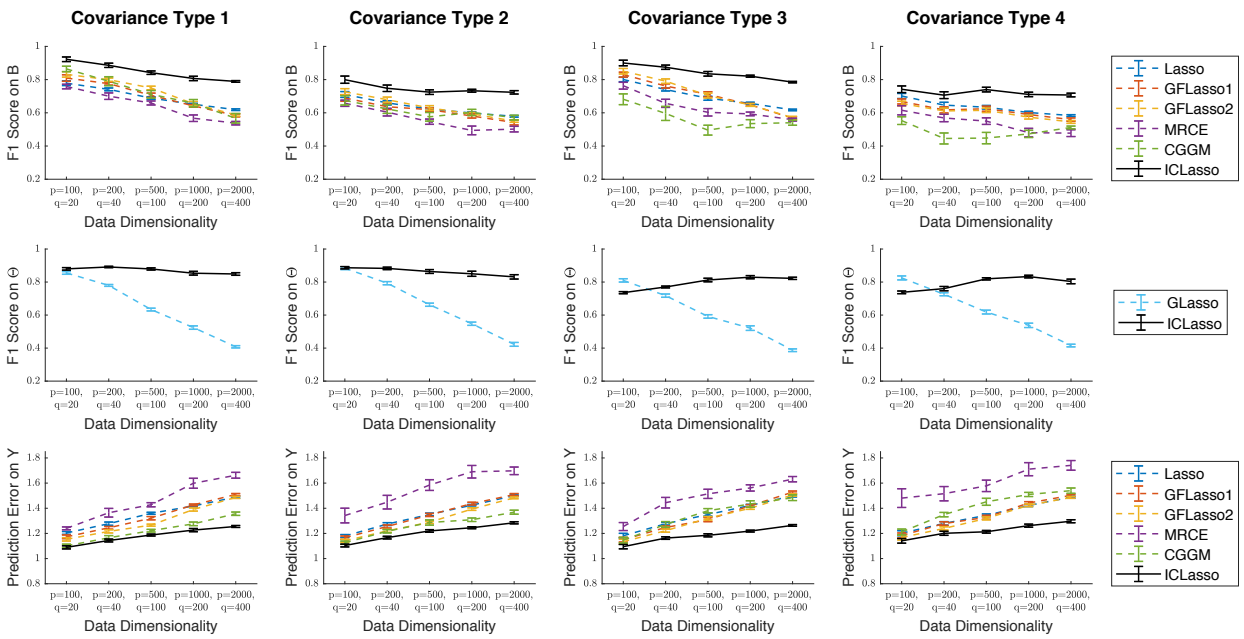


Figure 3.3: A comparison of results on synthetic data generated with each of the four different types of covariance structure and with several different values of p and q. We fix the group size to $g = 10$ and the number of SNP associations per gene to $s = 5$. The top row shows the F1 score on the recovery of the true nonzero elements of B. The second row shows the F1 score on the recovery of the true nonzero elements of Θ. The bottom row shows the prediction error on a held out test set. All results are averaged over 20 simulations, and the error bars show the standard error.

Table 3.1: Regression Error on Yeast Data

	density	training error	validation error
Lasso	1.65%	0.502	0.718
GFLasso	2.87%	0.392	0.715
ICLasso	6.88%	0.395	0.703

ICLasso outperforms all baselines in nearly all of the settings we consider.

3.4.2 Yeast eQTL Study

In order to evaluate our approach in a real-world setting and provide a proof of concept for our model, we applied ICLasso to a yeast eQTL dataset from [55] that consists of $2,956$ SNP genotypes and $5,637$ gene expression measurements across 114 yeast samples. To preprocess the data, we removed SNPs with duplicate genotypes and retained only the 25% of genes with the highest variance in expression, leaving $p = 1,157$ SNPs and $q = 1,409$ genes in our analysis.

We used our approach to jointly perform eQTL mapping and gene network inference on the yeast dataset, treating the the SNPs as inputs X and the genes as outputs Y. We trained our model on 91 samples and used the remaining 23 samples as a validation set for tuning the hyperparameters. Given the trained model, we read the eQTL associations from the regression coefficient matrix $\hat{B}$,

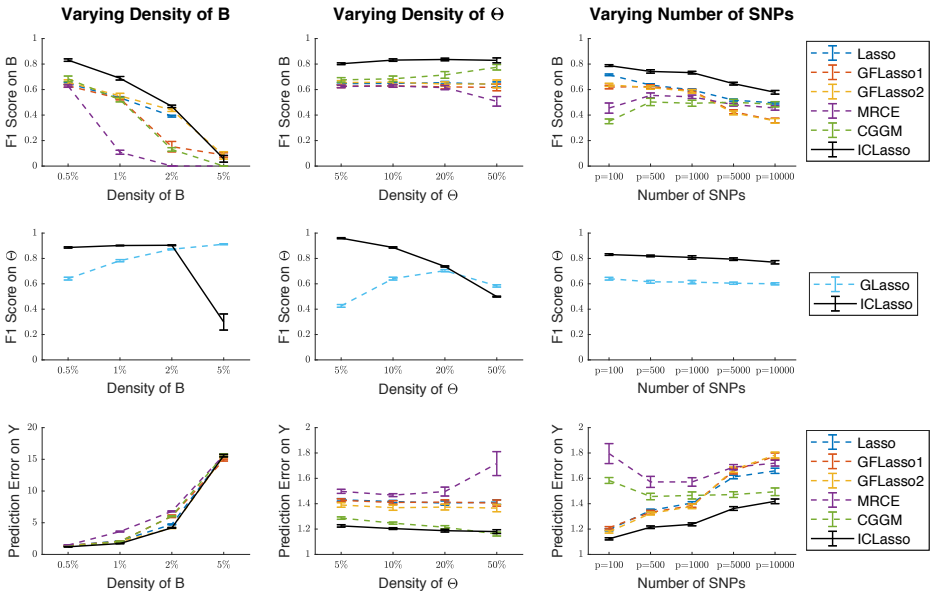


Figure 3.4: A comparison of results on synthetic data generated with different settings. In the first two columns, we use covariance type 0, $p = 1000$, $q = 100$, and vary the density of B and Θ. In the third column, we use covariance type 3, $q = 100$, $g = 10$, $s = 5$, and vary the number of SNPs. The top row shows the F1 score on the recovery of the true nonzero elements of B. The second row shows the F1 score on the recovery of the true nonzero elements of Θ. The bottom row shows the prediction error on a held out test set. All results are averaged over 20 simulations, and the error bars show the standard error.

which encodes SNP-gene relationships, and obtained the gene network from the inverse covariance matrix $\hat{\Theta}$, which encodes gene-gene relationships. In addition to ICLasso, we ran Lasso and GFlasso on the yeast data to obtain two additional estimates of B, and ran GLasso1 to obtain another estimate of Θ. Note that we chose not to compare to MRCE and CGGM because these methods performed worse than the other baselines in the most realistic data settings that we tested in our simulation experiments. Furthermore, we did not compare to GFLasso2 because the performance of the two variants of GFLasso that we evaluated were comparable.

Table 3.1 shows the density of $\hat{B}$ obtained with each method, along with the prediction error of Y on the training set and on the held-out validation set, which were calculated using $\|Y_{\text{train}} - X_{\text{train}}\hat{B}\|_F^2$ and $\|Y_{\text{valid}} - X_{\text{valid}}\hat{B}\|_F^2$, respectively. We chose not to sacrifice any data for a test set, but these results indicate that ICLasso achieves an equivalent or better fit to the training and validation sets than Lasso and GFLasso.

Quantitative Analysis. Because the true yeast eQTLs and gene network structure are not known, there is no ground truth for this problem. We instead analyzed the output of each method by performing a series of enrichment analyses that together provide a comprehensive picture of the biological coherence of the results. An enrichment analysis uses gene annotations to identify specific biological processes, functions, or structures that are over-represented among a group of genes relative to the full set of genes that is examined [56]. To evaluate our yeast data results, we performed three types of enrichment analyses: biological process and molecular function enrichment using annotations from the Gene Ontology (GO) database [57], and pathway enrichment using

Table 3.2: GO and KEGG Enrichment Analysis on Yeast eQTL Map

	number of enriched terms			avg. change	number of enriched SNPs			avg. change
	GO-BP	GO-MF	KEGG		GO-BP	GO-MF	KEGG	
Lasso	1862	804	205	—	198	132	127	—
GFLasso	3499	1528	312	+**77**%	286	211	172	+**47**%
ICLasso	8046	3147	1025	+**155**%	590	453	441	+**126**%

Table 3.3: GO and KEGG Enrichment Analysis on Yeast Gene Network

	number of enriched terms			avg. change	number of enriched clusters			avg. change
	GO-BP	GO-MF	KEGG		GO-BP	GO-MF	KEGG	
GLasso	173	77	31	—	14	12	11	—
ICLasso	321	127	41	+**61**%	29	26	22	+**108**%

annotations from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database [58]. We used a hypergeometric test to compute a p-value for each term, and then adjusted the values to account for multiple hypothesis testing. Significance was determined using an adjusted p-value cutoff of 0.01.

We first analyzed $\hat{B}$ by performing a per-SNP enrichment analysis. For each SNP j, we used the nonzero elements in $\beta_{j\cdot}$ to identify the set of genes associated with the SNP. Next we performed GO and KEGG enrichment analyses on this group of genes by comparing their annotations to the full set of $1,409$ genes that we included in our study. We repeated this procedure for each SNP, and calculated the total number of terms that were enriched over all SNPs to obtain a global measure of enrichment for $\hat{B}$. In addition, we calculated the total number of SNPs that were enriched for at least one term in each category. These results are summarized in Table 3.2. It is evident that ICLasso outperforms both GFLasso and Lasso on estimating the regression coefficients, since it has more than twice as many enriched terms in GO biological process, GO molecular function, and KEGG than either baseline.

Next we used a similar approach to evaluate the structure present in $\hat{\Theta}$. We first obtained groups of genes by using spectral clustering to perform community detection among the genes using the inferred network structure. After clustering the genes into 100 groups,[4] we performed GO and KEGG enrichment analyses on each cluster and calculated the total number of enriched terms along with the total number of clusters that were enriched for at least one term. These results are summarized in Table 3.3. Once again, our approach has more enrichment than the baseline in every category, which implies that the gene network estimated by ICLasso has a much more biologically correct structure than the network estimated by GLasso.

Qualitative Analysis. The quantitative results in Tables 3.2 and 3.3 indicate that, compared to other methods, our approach identifies more eQTLs that are associated with genes significantly enriched in certain biological processes and pathways. A more detailed examination of our results

[4]We also clustered with 25, 50, and 200 groups and obtained similar results.

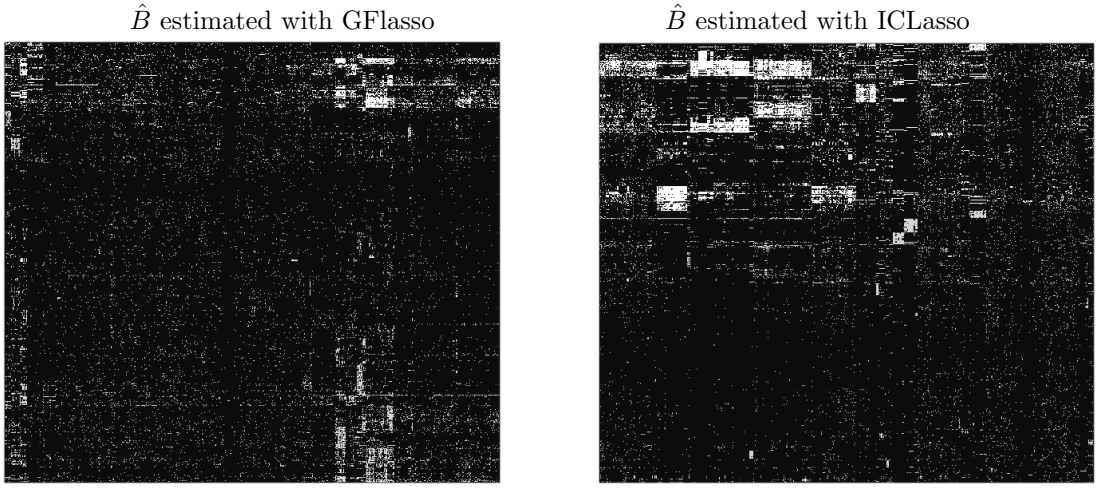


Figure 3.5: Binary heatmap of associations between SNPs (one per row) and genes (one per column), estimated with GFLasso and ICLasso. In each image, the SNPs and genes are ordered to maximize the visual clustering of associations.

revealed that many of the enriched terms correspond to metabolic pathways, and that the eQTLs we identified agree with those discovered in a previous study that analyzed the effect of genetic variations on the yeast metabolome.

Breunig et al. [59] identified the metabolite quantitative trait loci (mQTLs) for 34 metabolites and then examined each mQTL for the presence of metabolic genes in the same pathway as the linked metabolite. We found that 10 of these 34 metabolites were linked to metabolic genes where our identified eQTLs reside. For example, Breunig et. al. determined that the metabolite valine is linked to an mQTL in a region spanned by the ILV6 gene, which encodes a protein involved in valine biosynthesis. In our study, we also identified an eQTL located in ILV6. Moreover, we found that the eQTL in ILV6 is associated with 365 genes that are significantly enriched for pathways involved in the metabolism and biosynthesis of various amino acids. This is consistent with the fact that the metabolism and biosynthesis of amino acids in the cell needs to be coordinated.

Furthermore, our enrichment analysis shows that the eQTL-associated genes we identified are enriched for various metabolic pathways (e.g. sulfur, riboflavin, protein, starch, and sucrose metabolism, oxidative phosphorylation, glycolysis), as well as more general pathways, such as cell cycle pathways, and MAPK pathways. This is consistent with the roles of the mQTLs identified by Breunig et al. Interestingly, among these genes, SAM1, encoding an S-adenosylmethionine synthetase, is also among the eQTLs in our list. Our results show that the eQTL we found in SAM1 is associated with 252 genes that are enriched for cytoplasmic translation and ribosome functions, consistent with the fact that SAM is the methyl donor in most methylation reactions and is essential for DNA methylation of proteins, nucleic acids, and lipids [60].

Finally, to illustrate our results, we visualized the SNP-gene associations discovered by GFLasso and ICLasso by plotting a binary heatmap of the two estimates of B in Figure 3.5. Within each heatmap, both the SNPs and genes are sorted to maximize the clustering of associations. From these plots, it's clear that the associations discovered by ICLasso contain more interesting block structure than those discovered by GFLasso.

Table 3.4: Known Alzheimer's Disease Genes

Gene Symbol
APP, APOE, PLD3, TREM2, SORL1, GAB2, BIN1, CLU, CD33, CR1, PICALM, ABCA7, CD2AP, MS4A6A, MS4A4E

3.4.3 Human eQTL Study of Alzheimer's Disease

Finally, we applied our method to a human eQTL dataset in order to identify a set of interesting genomic loci that may play a role in Alzheimer's disease. For this study, we used a dataset from [61] that contains $n = 540$ case and control samples of patients with Alzheimer's disease, genotypes of $p = 555,091$ SNPs across all chromosomes, and mRNA expression values of $q = 40,638$ gene probes measured in the cerebellum, a region of the brain that governs motor control and some cognitive functions.

We preprocessed this data by selecting a subset of interesting SNPs and genes to include in our analysis. To filter genes, we calculated the marginal variance of the expression of each gene, the fold change in each gene's expression between the case and control samples, and the p-value of a t-test with the case-control status. We then selected all genes with variance in the top 10%, fold change in the top 10%, or p-value in the bottom 10%, along with a set of 15 genes known to be associated with Alzheimer's disease. These genes are listed in Table 3.4. To filter SNPs, we calculated the p-value of a chi-square test with the case-control status. We then selected all SNPs with uncorrected p-value < 0.05, along with all SNPs located within 500kb of any of the Alzheimer's genes. This filtering yielded $p = 24,643$ SNPs and $q = 9,692$ genes.

Applying ICLasso to this dataset yielded an estimate of $\hat{B}$ with 4.07% density and an estimate of $\hat{\Theta}$ with 4.87% density. To analyze the results, we first constructed a set of candidate SNPs comprised of the top 10 SNPs associated with each of the Alzheimer's genes based on association strength. Since some of the genes are represented by multiple probes in the dataset, there are 25 gene expression values corresponding to to the 15 Alzheimer's genes. From these, we identified 185 unique candidate eQTLs.

Next we performed an enrichment analysis for each of these SNPs by looking at the set of genes linked to each SNP in the eQTL map and determining whether these are enriched for any GO biological process terms relative to the full universe of $9,692$ genes. Among these, 58 (31%) are enriched for at least one term using a corrected p-value cutoff of 0.01. When analyzing the results, we noticed three categories of candidate eQTLs that might play a role in Alzheimer's disease: SNPs associated with genes enriched for immune functions, SNPs associated with genes enriched for metabolic functions, and SNPs associated with genes enriched for neural functions. A selected set of interesting results from each category are highlighted in Tables 3.5, 3.6, and 3.7.

One particularly interesting observation is that many of the SNPs in the first category are associated with genes implicated in myeloid cell activated immune response. This is notable because Alzheimer's disease has previously been linked to acute myeloid leukemia [62].

3.5 Discussion

In this work, we propose a new model called the *inverse-covariance fused lasso* which jointly estimates regression coefficients B and an output network Θ while using a graph-guided fused lasso penalty to explicitly encourage shared structure. Our model is formulated as a biconvex

Table 3.5: Candidate Alzheimer's Disease SNPs Linked to Immune Response

SNP	Chrom	Associated Alzheimer's Genes	Enriched Biological Process GO Terms (adjusted p-value < 0.01, category size > 5)
rs12058997	1	*CR1, CD33*	neutrophil degranulation; neutrophil activation; myeloid cell activation involved in immune response; myeloid leukocyte mediated immunity; hematopoietic or lymphoid organ development
rs1464401	3	*CR1, CD33, GAB2, APOE, TREM2*	myeloid cell activation involved in immune response; immune system development; circulatory system development; neutrophil activation; neutrophil degranulation; myeloid leukocyte mediated immunity
rs2187204	6	*CR1, CD33, APOE, TREM2, PICALM*	neutrophil degranulation; neutrophil activation, myeloid cell activation involved in immune response; immune system development; cardiovascular system development
rs780382	11	*APP, CD33, TREM2*	neutrophil activation; neutrophil degranulation; myeloid cell activation involved in immune response; myeloid leukocyte mediated immunity
rs11174276	12	*CLU, APOE*	leukocyte mediated immunity; negative regulation of cell death; positive regulation of NF-kappaB transcription factor activity
rs4072111	15	*CD33, TREM2, PICALM, GAB2*	neutrophil degranulation; neutrophil activation; myeloid cell activation involved in immune response; myeloid leukocyte mediated immunity

Table 3.6: Candidate Alzheimer's Disease SNPs Linked to Metabolic Processes

SNP	Chrom	Associated Alzheimer's Genes	Enriched Biological Process GO Terms (adjusted p-value < 0.01, category size > 5)
rs3795550	1	*APP, BIN1*	SRP-dependent cotranslational protein targeting to membrane; nuclear-transcribed mRNA catabolic process, nonsense-mediated decay
rs7094118	10	*SORL1*	nucleoside triphosphate metabolic process; nucleoside monophosphate metabolic process; mitochondrial translational elongation; mitochondrial translational termination
rs4945276	11	*GAB2*	ATP synthesis coupled electron transport; ribonucleoside triphosphate metabolic process; purine nucleoside monophosphate metabolic process; mitochondrial electron transport, NADH to ubiquinone
rs1571376	14	*SORL1*	nucleoside triphosphate metabolic process; ribonucleoside monophosphate metabolic process; energy derivation by oxidation of organic compounds; purine nucleoside monophosphate metabolic process; organophosphate metabolic process

optimization problem, and we derive new, efficient optimization routines for each convex subproblem based on existing methods.

Our results on both synthetic and real data unequivocally demonstrate that our model achieves

Table 3.7: Candidate Alzheimer's Disease SNPs Linked to Neural Activity

SNP	Chrom	Associated Alzheimer's Genes	Enriched Biological Process GO Terms (adjusted p-value < 0.01, category size > 5)
rs11123605	2	*BIN1, GAB2, CD33*	trans-synaptic signaling
rs2855794	11	*CR1*	detection of chemical stimulus involved in sensory perception of smell
rs3895113	22	*BIN1, GAB2*	ensheathment of neurons

significantly better performance on recovery of the structure of B, recovery of the structure of Θ, and prediction error than all six baselines that we evaluated. In this paper, we demonstrated that our approach can effectively be used to perform joint eQTL mapping and gene network estimation on a yeast dataset, yielding more biologically coherent results than previous work. However, the same problem setting appears in many different applications, and the inverse-covariance fused lasso model can therefore be effectively used within a wide range of domains.

The primary disadvantage of our proposed method is that it is not scalable in the number of genes. One promising direction for future work would be to explore an approximation in the style of [63] that performs neighborhood selection for estimating the gene network instead of solving for the exact value of Θ. Furthermore, the screening rules for the graphical lasso proposed in [64] can be directly extended to our model, and would likely provide a significant speedup when working with block sparse gene networks.

Chapter 4

Hybrid Subspace Learning

4.1 Introduction

High-dimensional datasets, in which the number of features p is much larger than the sample size n, appear in a broad variety of domains. Such datasets are particularly common in computational biology [65], where high-throughput experiments abound but collecting data from a large number of individuals is costly and impractical. In this setting, many traditional machine learning algorithms lack sufficient statistical power to distinguish signal from noise, a problem that is known as the curse of dimensionality [66].

One way to alleviate this problem is to perform dimensionality reduction, either by choosing a subset of the original features or by learning a new set of features. In this work, we focus on the class of subspace learning methods, whose goal is to find a linear transformation that projects the high-dimensional data points onto a nearby low-dimensional subspace. This corresponds to learning a latent space representation of the data that captures the majority of information from the original features.

The most popular subspace learning method is principal component analysis (PCA), which learns a compact set of linearly uncorrelated features that represent the directions of maximal variance in the original data [67]. Since PCA was first introduced, many variants have been developed. For example, Sparse PCA uses an elastic net penalty to encourage element-wise sparsity in the projection matrix, resulting in more interpretable latent features [68]. Another method, Robust PCA, learns a decomposition of the data into the sum of a low-rank component and a sparse component, leading to increased stability in the presence of noise [69]. Finally, there are approaches that propose richer models for the underlying latent representation of the data, involving multiple subspaces rather than just one [70].

A significant limitation of existing subspace learning methods is their assumption that the data, except for noise terms, can be fully represented by an embedding in one or more low-dimensional subspaces. While this may hold true in some settings, we contend that in most high-dimensional, real-world datasets, only a subset of the features exhibit low-rank structure, while the remainder are best represented in the original feature space. Specifically, since the low-rank features will be highly intercorrelated, they can be accurately represented as the linear combination of a small set of latent features. However, if there are raw features that are largely uncorrelated with the others, it's clear that including them in the latent space model would require adding one new dimension for each such feature. We therefore argue that these features, which we describe as exhibiting *high-dimensional* rather than *low-rank* structure, should be excluded from the low-dimensional subspace representation.

We illustrate this intuition with a simple example. Figure 4.1 shows two toy datasets that each

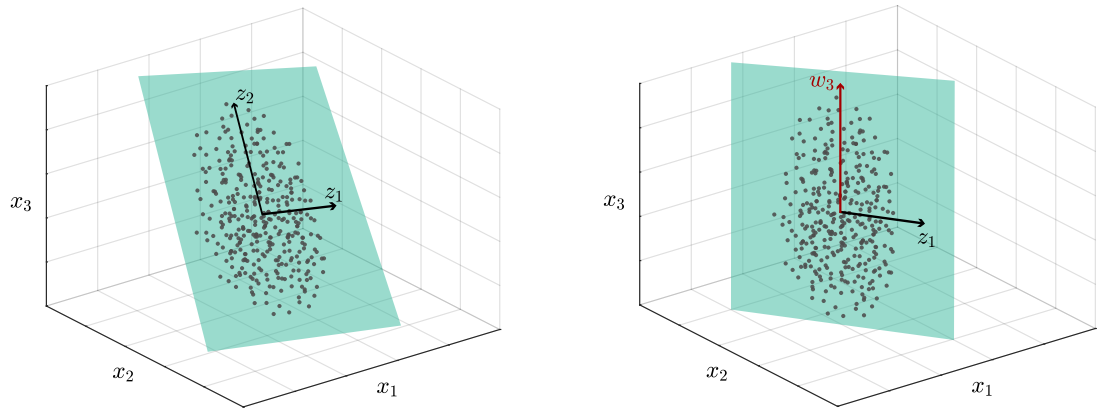


Figure 4.1: Toy datasets that illustrate the difference between fully low-rank data (left) and hybrid data (right). Here z_1 and z_2 represent latent features that are linear combinations of raw features, whereas w_3 represents a latent feature that is perfectly correlated with the raw feature x_3.

lie on a different 2D plane in 3D space. In the left plot, all three of the raw dimensions exhibit low-rank structure because they are all correlated. However, in the right plot, the vertical axis x_3 is completely uncorrelated with x_1 and x_2, which causes the 2D subspace on which the data points lie to be axis-aligned with x_3. We say that this data exhibits *hybrid structure* because only two out of the three features are truly low-rank.

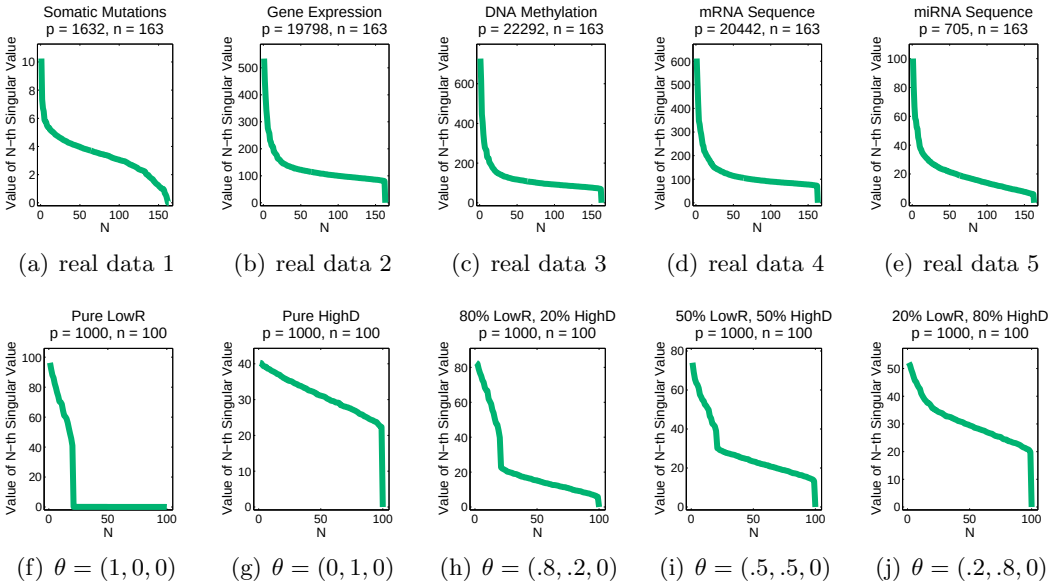
In this simple example, PCA would easily succeed on both of the datasets shown. However, in a high-dimensional and noisy setting, the data may not lie exactly on a low-rank subspace. In this case, we can boost the signal to noise ratio in the data by identifying a sparse set of high-dimensional features that do not contribute to the low-rank structure of the dataset and eliminating them from the low-rank projection. This is the core motivation for our approach.

In this work, we introduce a new method called *hybrid subspace learning* that learns a latent representation of the data in which some features are mapped to a low-rank subspace but others remain in the original high-dimensional feature space. To enforce this structure, we propose a novel regularization scheme that encourages each variable to choose between participating in the low-rank or high-dimensional component of the model. The resulting problem is biconvex, and we propose an efficient alternating minimization scheme using proximal gradient descent.

The goal of our hybrid method is to perform dimensionality reduction for high-dimensional datasets in a way that allows flexibility in the proportion of low-rank vs. high-dimensional structure that is present in the data, and is also robust to noise. This approach has connections to Outlier Pursuit [71], a variant of PCA that attempts to learn a latent space representation of the data in the presence of outliers (*i.e.* points that do not lie on the same low-rank subspace as the others). However, in our case, we treat features as outliers instead of points.

This work has two main contributions. First, we propose the idea of learning a partial low-rank representation of the data by identifying features that are outliers. We demonstrate that certain high-dimensional datasets naturally exhibit hybrid structure, indicating that our idea is useful for solving real-world tasks. Second, we introduce a new regularization term that encourages mutually exclusive sparsity. We show that this penalty outperforms the simple $l_{1,2}$ norm in our setting, and we provide practical guidelines for optimizing it.

Notation. We use lowercase bold symbols for vectors $\mathbf{x}$ and uppercase bold symbols for matrices $\mathbf{X}$. The i^{th} element of $\mathbf{x}$ is denoted $\mathbf{x}(i)$, the i^{th} row and j^{th} column of $\mathbf{X}$ are denoted $\mathbf{X}(i,:)$ and



(a) real data 1 (b) real data 2 (c) real data 3 (d) real data 4 (e) real data 5

(f) $\theta = (1, 0, 0)$ (g) $\theta = (0, 1, 0)$ (h) $\theta = (.8, .2, 0)$ (i) $\theta = (.5, .5, 0)$ (j) $\theta = (.2, .8, 0)$

Figure 4.2: Singular value spectra of real and synthetic datasets; (a)-(e) five real biological datasets collected from tumor samples of 163 leukemia patients; (f) synthetic data with pure low-rank structure; (g) synthetic data with pure high-dimensional structure; (h)-(j) synthetic data with varying degrees hybrid structure.

$\mathbf{X}(:, j)$, respectively, and $\text{diag}(\mathbf{x})$ denotes a diagonal matrix $\mathbf{X}$ s.t. $\mathbf{X}(i, i) = \mathbf{x}(i)$. We use $\|\cdot\|_1$ for the element-wise l_1 norm of a vector or matrix, $\|\cdot\|_2$ for the l_2 norm of a vector, $\|\cdot\|_F$ for the Frobenius norm of a matrix, and $\|\cdot\|_{1,p}$ to denote an $l_{1,p}$ column-wise block norm of a matrix s.t. $\|\mathbf{X}\|_{1,p} = \sum_j \|\mathbf{A}(:, j)\|_p$.

4.2 Motivation

In this section, we motivate our approach by demonstrating that certain properties of several real-world datasets naturally hint at a hybrid model. To do this, we use a series of simulations to show that hybrid structure causes the singular value spectrum of a dataset to become long-tailed, *i.e.* to have a distribution in which much of the probability mass is far from the mean. We then provide examples of real datasets that possess long-tailed singular value spectra, which implies that it is not appropriate to attempt to capture all of the information contained in these datasets with a low-dimensional feature representation.

Consider a dataset $\mathbf{X} \in \mathbb{R}^{n \times p}$ with n samples and p features. The top row of Figure 4.2 shows the singular value spectra of five genomic datasets that consist of measurements taken from tumor samples of cancer patients. In all of these datasets, the singular values start out large but then decay very quickly. However, instead of going directly to zero, the spectrum has a long tail. This points to the presence of structure in the data that does not fit into a low-rank space. As a result, if we ignored the tail by projecting the data to a low-rank subspace, it is likely that we would only capture a very coarse-grained representation of the data.

We compare these real datasets with several simulated datasets to demonstrate how certain underlying modeling assumptions affect the singular value spectrum of the data. We generate synthetic data as follows. Let $\mathbf{Z}$ be an $n \times k$ matrix with full column rank, $\mathbf{A}$ be a $k \times p$ matrix with full row rank, and $\mathbf{W}$ be an $n \times p$ matrix whose elements are independent. Define a probability

vector $\theta = (\theta_1, \theta_2, \theta_3)$ that specifies the likelihood that each feature participates in only a low-rank component, only a high-dimensional component, or both, respectively. For simplicity, we consider only the case of $\theta_3 = 0$ for now. For each variable $j \in \{1, \ldots, p\}$, we draw $C_j \sim \text{Categorical}(\theta)$. Then if $C_j = (1, 0, 0)$, we set $\mathbf{X}(:, j) \sim \mathcal{N}(\mathbf{ZA}(:, j), \sigma^2 \mathbf{I}_{n \times n})$ and if $C_j = (0, 1, 0)$, we set $\mathbf{X}(:, j) \sim \mathcal{N}(\mathbf{W}(:, j), \sigma^2 \mathbf{I}_{n \times n})$.

For our simulations, we use $n = 100$, $p = 1000$, $k = 20$, and σ^2 close to 0. We plot the spectra of synthetic datasets generated for multiple values of θ in the bottom row of Figure 4.2. In panel (f), we set $\theta = (1, 0, 0)$ such that $\mathbf{X}$ is rank k with some random noise. In this case the singular value spectrum drops sharply after k, but the tail that appears in the real data is missing. While it is possible that the tail could only contain noise, we postulate that it contains some important information that is ignored by subspace learning methods that focus purely on low-rank structure. In panel (g), we set $\theta = (0, 1, 0)$ such that $\mathbf{X}$ has rank n. In this case, the singular value spectrum of $\mathbf{X}$ decays slowly, again unlike the real data. This implies that methods that use the full data matrix $\mathbf{X}$ without alteration are not exploiting its intrinsic structure.

Panels (h)-(j) display three "hybrid" settings of θ. The spectra of these datasets exhibit structure that is much more similar to the real data, with a few large singular values and a tail that decays slowly. In these cases, forcing all of the variables to fit into a subspace would necessitate including a large number of dimensions in that subspace, many of which would be highly underutilized. This is the motivation for our hybrid approach that can model both the head and tail of the singular value spectrum.

4.3 Method

4.3.1 Hybrid Matrix Factorization Model

Given a dataset $\mathbf{X} \in \mathbb{R}^{n \times p}$, traditional subspace learning aims to solve the following problem:

$$\min_{\mathbf{Z}, \mathbf{A}} \ \|\mathbf{X} - \mathbf{ZA}\|_F^2 \tag{4.1}$$

where $\mathbf{Z} \in \mathbb{R}^{n \times k}$ is a k-dimensional representation of each point and $\mathbf{A} \in \mathbb{R}^{k \times p}$ is a transformation that maps the latent space to the observed feature space. The above model, which is equivalent to PCA when the columns of $\mathbf{Z}$ are constrained to be orthogonal, implicitly assumes that all of the information in $\mathbf{X}$ can be captured by its embedding in a low-rank subspace. However, as previously discussed, this assumption is inappropriate for high-dimensional data with a long-tailed singular value spectrum.

To overcome this limitation, we propose a new, flexible model for subspace learning that allows each feature in $\mathbf{X}$ to choose between participating in a low-rank representation, $\mathbf{Z}$, or a high-dimensional representation, $\mathbf{W}$. With this formulation, the goal is to have the low-rank ("low-r") component capture the head of the singular value spectrum while the high-dimensional ("high-d") component captures the tail. This leads naturally to the following problem:

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{A}, \mathbf{W}, \mathbf{b}} \ & \|\mathbf{X} - \mathbf{ZA} - \mathbf{W} \operatorname{diag}(\mathbf{b})\|_F^2 + \lambda \|\mathbf{b}\|_0 \\ \text{s.t.} \quad & \|\mathbf{A}(:, j)\|_2 \cdot \mathbf{b}(j) = 0 \quad \forall\, j \\ & \|\mathbf{W}\|_F \leq 1 \end{aligned} \tag{4.2}$$

Here, $\mathbf{Z} \in \mathbb{R}^{n \times k}$ is the low-rank component (as before) and $\mathbf{W} \in \mathbb{R}^{n \times p}$ is the high-dimensional component. Furthermore, $\mathbf{b} \in \{0, 1\}^p$ is a vector of indicator variables, each of which dictates

whether or not a particular feature j participates in the high-d component. We apply an l_0 norm regularizer to restrict the total number of features that are captured by the high-d component. Finally, we constrain the problem such that each feature belongs to exactly one component.

However, this problem is intractable for two reasons. First, the l_0 penalty is highly nonconvex and difficult to optimize. Secondly, since $\mathbf{A}$ and $\mathbf{b}$ are coupled in the constraint, they cannot be optimized jointly. Performing alternating minimization on (4.2) would yield degenerate solutions, since initializing $\mathbf{b}(j)$ to non-zero would always force $\mathbf{A}(:, j)$ to be zero and vice-versa. We therefore propose the following relaxation:

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{A},\mathbf{W},\mathbf{b}} \quad & \|\mathbf{X} - \mathbf{Z}\mathbf{A} - \mathbf{W}\,\mathrm{diag}(\mathbf{b})\|_F^2 \\ & + \gamma\|\mathbf{A}\,\mathrm{diag}(\mathbf{b})\|_{1,2} + \lambda\|\mathbf{b}\|_1 \\ \text{s.t.} \quad & \|\mathbf{Z}\|_F \leq 1 \quad \|\mathbf{W}\|_F \leq 1 \end{aligned} \tag{4.3}$$

We make two changes in order to arrive at (4.3). First, as is common in the sparsity literature, we relax $\mathbf{b} \in \{0, 1\}^p$ to $\mathbf{b} \in \mathbb{R}^p$, and replace the l_0 penalty on $\mathbf{b}$ with an l_1 penalty. Second, and more unique to our problem, we replace the hard constraint on $\mathbf{A}$ and $\mathbf{b}$ in (4.2) with a structured sparse regularizer that encourages each feature to participate in either the low-r component ($\mathbf{Z}$) or the high-d component ($\mathbf{W}$), but not both. This is achieved with an $l_{1,2}$ norm penalty of the form $\|\mathbf{A}\,\mathrm{diag}(\mathbf{b})\|_{1,2} = \sum_{j=1}^{p} \mathbf{b}(j)\|\mathbf{A}(:, j)\|_2$. Notice that sparsifying either the jth element of $\mathbf{b}$ or the jth column of $\mathbf{A}$ will completely zero out the jth term of the penalty. This regularization scheme therefore encourages mutually exclusive sparsity over the columns of $\mathbf{A}$ and the elements of $\mathbf{b}$. Furthermore, once the jth term of the penalty is zero, there is no longer any shrinkage applied to the jth column of $\mathbf{A}$, which yields a better estimate of the model parameters and eliminates the need for refitting the low-rank model after the high-d features have been identified.

As γ tends to ∞, the model shown in (4.3) will enforce the hard constraint in (4.2). Conveniently, as we will see in the next section, this relaxation also permits us to develop a much more effective optimization procedure that is less likely to be trapped in local optima. At the same time, the new model is more flexible than (4.2) in that it can allow some overlap between $\mathbf{A}$ and $\mathbf{b}$ at the cost of having an additional tuning parameter.

Our approach, hybrid subspace learning (HSL), is closely related to Robust PCA (RPCA) [69] and its variants, which learn a decomposition of the data $\mathbf{X}$ into the sum of a low-rank component $\mathbf{L}$ and a sparse component $\mathbf{S}$. In particular, while RPCA encourages element-wise sparsity in $\mathbf{S}$, Outlier Pursuit (OP) [71] is a more structured approach that encourages row-wise sparsity in $\mathbf{S}$ in order to identify points in the dataset that are outliers, and allow them to be ignored by the low-rank representation $\mathbf{L}$. The OP model can just as easily be applied to a transposed data matrix to identify features that are "outliers" because they can't easily be embedded in a low-rank subspace. Although this is conceptually very similar to the core idea of HSL, there are several key differences.

First, and most importantly, HSL also strictly enforces sparsity in the projection matrix $\mathbf{A}$, which causes some features to be completely excluded from the low-rank representation. In OP, although $\mathbf{S}$ can be made column-wise sparse, there is nothing to prevent the features that participate in $\mathbf{S}$ from also participating in $\mathbf{L}$. Second, we learn an exact rank k low-rank representation, whereas OP aims to minimize the nuclear norm of $\mathbf{L}$.

Finally, HSL also has some connections to Sparse PCA (SPCA) [68], which learns a rank k decomposition of $\mathbf{X}$ given by $\mathbf{ZA}$, where $\mathbf{A}$ is encouraged to be element-wise sparse.

Algorithm 4.1 Proximal Gradient Descent for HSL

1: **inputs:** data matrix $\mathbf{X}$; regularization parameters λ, γ; step size α; initial values $\mathbf{Z}$, $\mathbf{A}$, $\mathbf{W}$, $\mathbf{b}$
2: initialize $\hat{\mathbf{Z}}$, $\hat{\mathbf{A}}$, $\hat{\mathbf{W}}$, $\hat{\mathbf{b}}$ using provided initial values
3: **repeat**
4: fix $\mathbf{Z} = \hat{\mathbf{Z}}$, $\mathbf{b} = \hat{\mathbf{b}}$
5: initialize $\mathbf{W}^0 = \hat{\mathbf{W}}$, $\mathbf{A}^0 = \hat{\mathbf{A}}$
6: **repeat** ▷ Optimize $\{\mathbf{W}, \mathbf{A}\}$
7: $\mathbf{W}^+ = \mathbf{W}^t - \alpha \nabla_{\mathbf{W}} \ell(\mathbf{Z}, \mathbf{A}^t, \mathbf{W}^t, \mathbf{b})$
8: $\mathbf{W}^{t+1} = l_F\text{–project}\big(\mathbf{W}^+\big)$ ▷ Eq. (4.4)
9: $\mathbf{A}^+ = \mathbf{A}^t - \alpha \nabla_{\mathbf{A}} \ell(\mathbf{Z}, \mathbf{A}^t, \mathbf{W}^t, \mathbf{b})$
10: $\mathbf{A}^{t+1} = l_2\text{–prox}\big(\mathbf{A}^+, \alpha\,\gamma|\mathbf{b}|\big)$ ▷ Eq. (4.5)
11: **until** convergence
12: fix $\mathbf{W} = \hat{\mathbf{W}}$, $\mathbf{A} = \hat{\mathbf{A}}$
13: initialize $\mathbf{Z}^0 = \hat{\mathbf{Z}}$, $\mathbf{b}^0 = \hat{\mathbf{b}}$
14: **repeat** ▷ Optimize $\{\mathbf{Z}, \mathbf{b}\}$
15: $\mathbf{Z}^+ = \mathbf{Z}^t - \alpha \nabla_{\mathbf{Z}} \ell(\mathbf{Z}^t, \mathbf{A}, \mathbf{W}, \mathbf{b}^t)$
16: $\mathbf{Z}^{t+1} = l_F\text{–project}\big(\mathbf{Z}^+\big)$ ▷ Eq. (4.4)
17: $\mathbf{b}^+ = \mathbf{b}^t - \alpha \nabla_{\mathbf{b}} \ell(\mathbf{Z}^t, \mathbf{A}, \mathbf{W}, \mathbf{b}^t)$
18: $\mathbf{b}^{t+1} = l_1\text{–prox}\big(\mathbf{b}^+, \alpha\,(\gamma\|\mathbf{A}\|_{\cdot,2} + \lambda)\big)$ ▷ Eq. (4.6)
19: **until** convergence
20: **until** convergence
21: **outputs:** estimates $\hat{\mathbf{Z}}$, $\hat{\mathbf{A}}$, $\hat{\mathbf{W}}$, $\hat{\mathbf{b}}$

4.3.2 Optimization Algorithm

Our optimization objective consists of a differentiable, biconvex loss function,

$$\ell(\mathbf{Z}, \mathbf{A}, \mathbf{W}, \mathbf{b}) = \|\mathbf{X} - \mathbf{Z}\mathbf{A} - \mathbf{W}\,\mathrm{diag}(\mathbf{b})\|_F^2$$

and two non-smooth, biconvex regularizers,

$$\psi(\mathbf{A}, \mathbf{b}) = \|\mathbf{A}\,\mathrm{diag}(\mathbf{b})\|_{1,2} \quad \text{and} \quad \phi(\mathbf{b}) = \|\mathbf{b}\|_1.$$

The objective is jointly convex in $\{\mathbf{W}, \mathbf{A}\}$ when $\mathbf{Z}$ and $\mathbf{b}$ are fixed, and is jointly convex in $\{\mathbf{Z}, \mathbf{b}\}$ when $\mathbf{W}$ and $\mathbf{A}$ are fixed. We implement an alternating minimization scheme to solve this problem, in which we iteratively optimize each convex sub-problem until the complete objective converges. Since the objective function of each sub-problem consists of a smooth, convex loss function plus a non-smooth, convex regularizer, we can leverage well-known tools to optimize functions of this form. Specifically, we apply proximal gradient descent, which projects the gradient step back onto the solution space at each iteration. The complete optimization procedure is outlined in Algorithm 4.1. In practice, we employ accelerated proximal gradient descent with line search to achieve a convergence rate of $O(1/\sqrt{\epsilon})$ [72]. We also find that 25-50 outer iterations is typically sufficient to reach convergence.

The projection and proximal operators used on lines 8, 10, 16, and 18 of Algorithm 4.1 are defined as:

$$l_F\text{–project}\big(\mathbf{W}\big) = \mathbf{W}/\max\{1, \|\mathbf{W}\|_F\} \tag{4.4}$$
$$l_2\text{–prox}\big(\mathbf{a}, u\big) = \mathbf{a} \cdot \max\{0, \|\mathbf{a}\|_2 - u\}/\|\mathbf{a}\|_2 \tag{4.5}$$
$$l_1\text{–prox}\big(b, u\big) = \mathrm{sign}(b) \cdot \max\{0, |b| - u\} \tag{4.6}$$

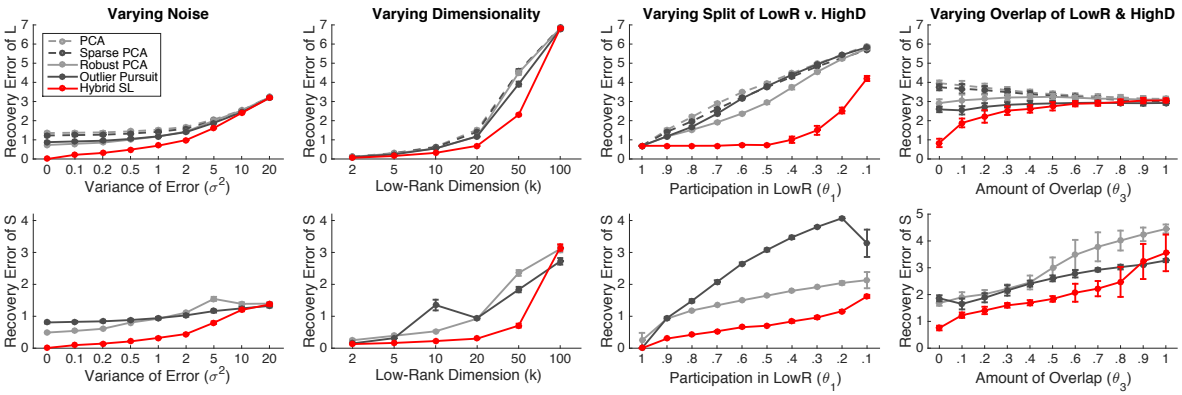


Figure 4.3: Results comparing the performance of our hybrid model against four baselines on synthetic data. The top row shows the recovery error for the low-rank component $\mathbf{L}$, and the bottom row shows the recovery error for the high-dimensional component $\mathbf{S}$. Results are averaged over 10 simulated datasets, and the error bars show the standard error over these trials.

These are applied column-wise or element-wise when given matrix arguments in place of vectors or vector arguments in place of scalars, respectively. We also use $|\mathbf{b}|$ to denote the element-wise absolute value of $\mathbf{b}$, and $\|\mathbf{A}\|_{\cdot,2}$ to denote the column-wise l_2 norm of $\mathbf{A}$.

Although this optimization procedure is quite efficient, the algorithm can easily get trapped in local optima. The joint regularization term compounds the problem by increasing the sensitivity of the algorithm to initialization, especially when the value of γ is very high. However, when γ is small, these local optima are substantially reduced. Therefore, to circumvent this problem, we fit our model to data by incrementally increasing the value of γ from 0 to $\gamma_{\max}$, while using warm starts to initialize the estimate of each successive model.[1] We define $\gamma_{\max}$ as the smallest value of γ that yields $\|\mathbf{A}\,\mathrm{diag}(\mathbf{b})\|_{1,2} = 0$. In the next section, we demonstrate empirically that using warm starts in place of cold starts leads to significant performance gains.

4.4 Experiments

4.4.1 Simulation Study

In order to quantitatively evaluate our approach, we perform a series of experiments on synthetic data. We compare HSL to four baseline methods: PCA, Sparse PCA [68], Robust PCA [69], and Outlier Pursuit [71]. Note that we apply OP to the transposed data matrix, $\mathbf{X}^T$.

We generate synthetic data as follows. Given raw feature space dimensionality p, latent space dimensionality k, and sample size n, we first generate low-rank features from $\mathbf{Z} \sim \mathcal{N}(0, \mathbf{I}_{k\times k})$ and high-dimensional features from $\mathbf{W} \sim \mathcal{N}(0, \mathbf{I}_{p\times p})$. We then generate coefficients for the low-r component $\mathbf{A}$ by drawing uniform random values in $[-1.5, -0.5] \cup [0.5, 1.5]$ and similarly generate coefficients for the high-d component $\mathbf{b}$ by drawing uniformly at random from $\sqrt{k}[-1.5, -0.5] \cup \sqrt{k}[0.5, 1.5]$. Next, given a probability vector $\theta = (\theta_1, \theta_2, \theta_3)$ whose elements denote the likelihood that a feature will participate in only the low-r component (θ_1), only the high-d component (θ_2), or both (θ_3), we incorporate sparsity by setting randomly chosen columns of $\mathbf{A}$ and elements of $\mathbf{b}$ to zero according to the proportions specified in θ. Finally we generate the data according to

[1] This is based on [73] who proposed using warm starts for a nonconvex sparse regularizer.

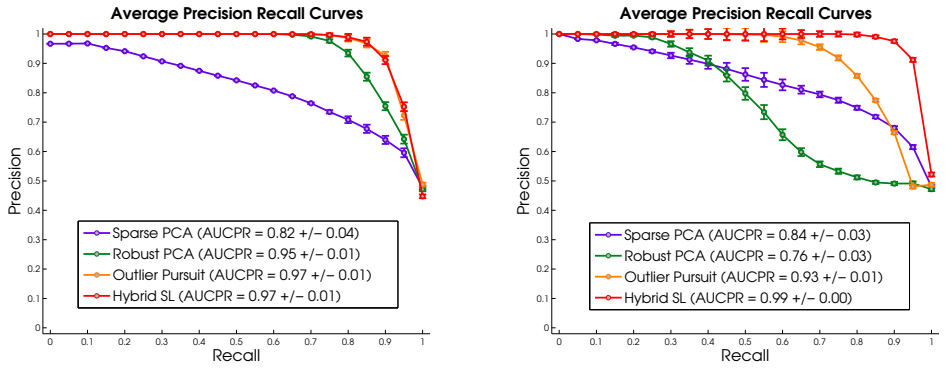


Figure 4.4: Average precision-recall curves for SPCA, RPCA, OP, and HSL calculated by varying hyperparameter values and evaluating recovery of the true set of high-dimensional features. Each curve is averaged over 20 simulated datasets.

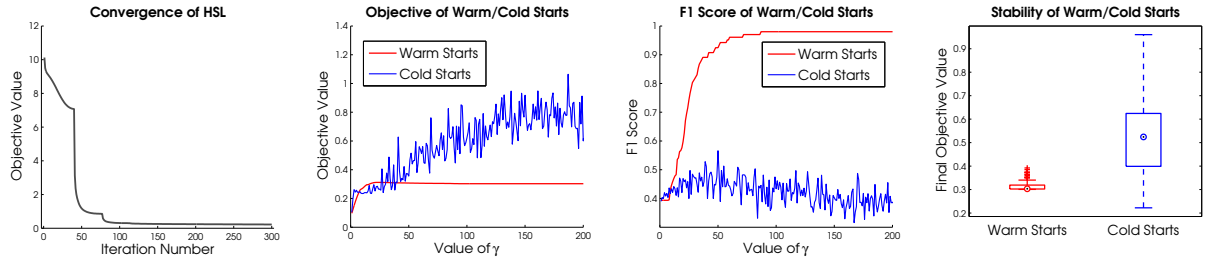


Figure 4.5: (a) Convergence of HSL. (b) The final objective value obtained from running HSL with each value of γ using warm and cold starts. (c) The F1 score on the selection of high-dimensional features obtained from running HSL with each value of γ using warm and cold starts. (d) The final objective value of HSL averaged over multiple simulations with warm and cold starts.

$\mathbf{X} = \mathbf{ZA} + \mathbf{W}\,\mathrm{diag}(\mathbf{b}) + \mathbf{E}$, where $\mathbf{E} \sim \mathcal{N}(0, \sigma^2)$ is i.i.d. Gaussian noise.

We compare the performance of our method against the baselines on three tasks: recovery of the low-rank subspace, recovery of the high-dimensional component, and selection of the set of high-dimensional features. We measure the recovery error using the Frobenius norm distance between estimated and true matrices, and evaluate the identification of the high-dimensional feature set using precision and recall. Since parameter selection is a challenging task in unsupervised learning, each method is run with the ground truth value of k, and tuning parameters are chosen by picking the values that yield the best recovery of the low-rank subspace. We believe this provides a fair comparison of all methods.

In our first set of experiments, we use default parameter values $n = 100$, $p = 200$, $k = 20$, $\sigma^2 = 1$, $\theta = (0.9, 0.1, 0)$, and then vary certain parameters in order to evaluate the performance of our model under a wide range of settings. In particular, we vary (a) the noise σ^2, (b) the dimensionality of the latent and feature space k, (c) the proportion of low-r and high-d participation (θ_1 v. θ_2) with no overlap, and (d) the amount of overlap (θ_3) with θ_1 and θ_2 set to the same value.[2] In the first three cases, we run HSL with $\gamma \to \gamma_{\max}$ to ensure no overlap between the low-r and high-d components. In the fourth case, we pick the optimal value of γ. The results of these experiments are shown in

[2]Note that in the second experiment, we also scale the variance of the noise σ^2 by a factor of $k/20$. This counteracts the fact that the magnitude of the generated features depends on the value of k.

Figure 4.3. They demonstrate that HSL significantly outperforms all baselines in most conditions.

Next, we generated precision-recall curves for the task of identifying the correct set of high-dimensional features. We compared the performance of SPCA, RPCA, OP, and HSL in Figure 4.4. The left panel shows the PR curve generated using the standard data generation approach that we previously described. Although HSL achieves a very high AUC, several other methods perform just as well. In order to increase the difficulty of this task, we generated data in which the average variance of the high-dimensional features is about half the average variance of the low-rank features, making them harder to distinguish. The right panel shows the PR curve generated from this data. In the second case, HSL achieves a significantly higher average area under the curve than all other methods.

Finally, we perform an empirical analysis of the effects of using cold starts versus warm starts to optimize our model. To do this, we train a series of models with different values of γ in two ways. Using cold starts, we randomly initialize each model. Using warm starts, we start with $\gamma = 0$ and then increase its value incrementally, each time initializing the model with the estimate obtained from the previous value, until we hit $\gamma_{\max}$. We evaluate the performance of HSL using these two approaches. The results are shown in Figure 4.5, and illustrate that using warm starts helps avoid local optima and leads to increased stability. Figure 4.5 also shows that HSL with warm starts exhibits good convergence properties.

4.4.2 Genomic Analysis of Cancer

Next we apply HSL to biomedical data, and provide both qualitative and quantitative results to illustrate its performance. A common data type in which $p \gg n$ is microarray data, in which the number of features measured typically far exceeds the number of patients for whom data is available. Here, we study the effectiveness of applying subspace learning methods to microarray data taken from cancer patients. We show that our approach outperforms several baselines on this data. Specifically, HSL produces subspace embeddings that achieve lower reconstruction error and lead to better performance on downstream tasks than competing methods. Finally, we demonstrate that HSL can also be used as a feature selection algorithm, since the features assigned to the high-dimensional component reflect biological characteristics of the original data.

To conduct our experiments, we used two datasets from TCGA.[3]. The first dataset contains miRNA expression levels for five types of cancer. We used this dataset to evaluate how well the low-rank embedding of HSL captures the original data and its characteristics. The second dataset contains gene expression data for breast cancer patients with matching tumor and control samples. We used this to analyze the high-dimensional component of HSL and to determine whether the information contained in the HSL estimate can differentiate between cancer and control samples. Additional details about these datasets are provided in Table 4.1.

For each dataset, the number of latent dimensions k was chosen by manually inspecting the singular value spectrum. This value was determined to be $k = 5$ for the miRNA datasets and $k = 30$ for the gene expression dataset. In all experiments, we selected hyperparameter values as follows. For RPCA, the value of λ was set to $\frac{1}{\sqrt{n}}$, which can optimally recover the low-rank structure under standard assumptions [69]. In keeping with our synthetic experiments, OP was run on the transposed data matrix. The value of λ for OP was chosen to produce a low-rank component with rank equal to k. For HSL, parameters were selected by performing a grid search and selecting the combination of parameters that minimized the AIC score [74].

[3]The Cancer Genome Atlas, http://cancergenome.nih.gov/.

Table 4.1: List of Genomic Datasets

Data Type	Cancer Type	Organ	Sample Size	Feature Size
miRNA expression	breast invasive carcinoma	breast	106	354
miRNA expression	glioblastoma multiforme	brain	93	354
miRNA expression	colon adenocarcinoma	colon	216	354
miRNA expression	kidney renal clear cell carcinoma	kidney	123	354
miRNA expression	lung adenocarcinoma	lung	107	354
gene (mRNA) expression	breast invasive carcinoma	breast	106	13,794

Table 4.2: Reconstruction Errors of the Low-Rank Component of miRNA Data

Tumor Type	PCA	Robust PCA	Outlier Pursuit	HSL
Breast	63.99	**29.46**	172.44	29.61
Colon	83.73	33.09	141.17	**31.32**
GBM	70.35	106.08	303.06	**40.69**
Kidney	54.77	45.56	179.56	**25.93**
Lung	54.74	**25.31**	172.97	25.73

Table 4.3: Silhouette Scores for Clusters Produced by k-Means

Tumor Type	Raw Data	PCA	Robust PCA	Outlier Pursuit	HSL
Breast	0.35 ± 0.07	0.51 ± 0.04	$.27 \pm .02$	0.17 ± 0.02	$\mathbf{0.65 \pm 0.08}$
Colon	0.37 ± 0.17	0.52 ± 0.07	0.30 ± 0.04	0.15 ± 0.05	$\mathbf{0.70 \pm 0.07}$
GBM	0.22 ± 0.05	0.45 ± 0.06	0.20 ± 0.03	0.15 ± 0.07	$\mathbf{0.48 \pm 0.06}$
Kidney	0.26 ± 0.04	0.43 ± 0.04	0.24 ± 0.02	0.13 ± 0.04	$\mathbf{0.59 \pm 0.08}$
Lung	0.29 ± 0.05	$\mathbf{0.53 \pm 0.05}$	0.28 ± 0.03	0.19 ± 0.05	0.52 ± 0.09

In our first experiment, we evaluated the quality of the low-r components estimated for each miRNA dataset. To do this, we measured the reconstruction errors of the low-r embeddings produced by each method. Reconstruction errors, calculated as the Euclidean distance between the original data $\mathbf{X}$ and the estimated low-r component $\hat{\mathbf{L}}$, are shown in Table 4.2. We see that HSL performs at least comparably, and frequently outperforms, all baseline methods on all datasets.

Next, we hypothesized that the low-r component of the HSL embedding may be more biologically informative than those estimated by traditional subspace learning methods. To study this, we used the estimated low-rank embeddings from each method to cluster the samples within each cancer type into subtypes. Since we do not have ground truth information about the subtypes, we evaluated the quality of the clusters by their silhouette scores, which provide a measure of how well the samples fit into their respective clusters. We performed k-means clustering using 4 clusters for breast [75], GBM [76], and colon [77] cancers and 5 clusters for kidney [78] and lung [79] cancers, where the number of clusters is based on the number of experimentally identified subtypes. The mean and standard deviation of the silhouette scores over 100 initializations of the clustering algorithm are shown in Table 4.3. From these results, we see that the features extracted from the low-r component of the hybrid model yield more coherent clusters than features extracted from baseline methods.

Since our hybrid model does not encode all the features of the original data in the low-rank

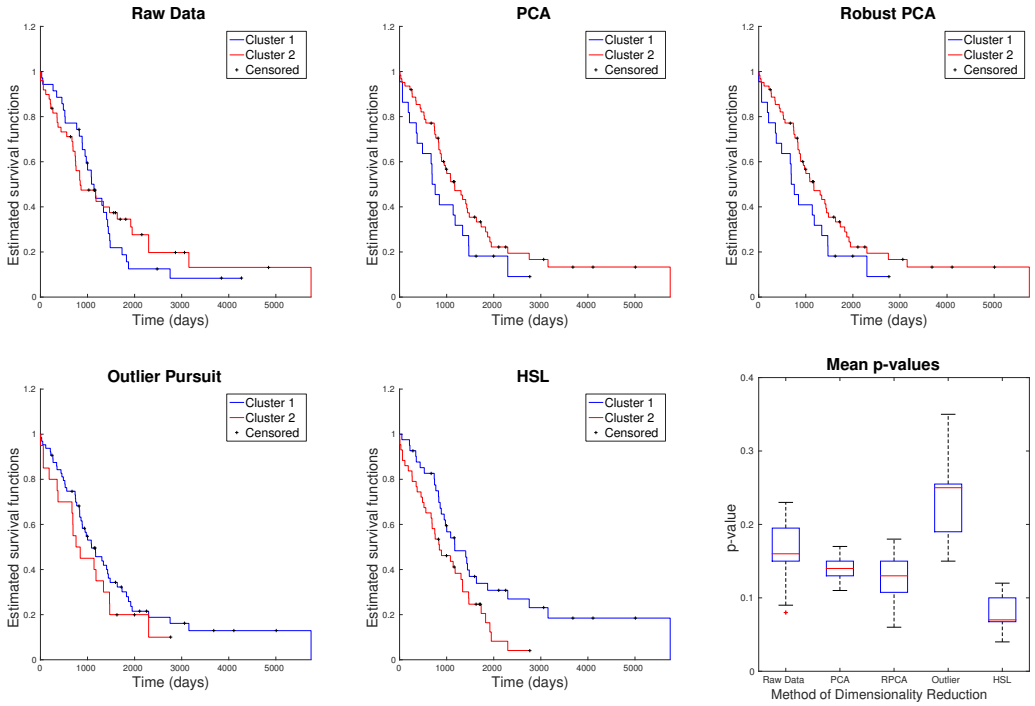


Figure 4.6: Results of survival analysis performed on a breast cancer gene expression dataset. (a-e) Examples of representative Kaplan-Meier survival function estimates. (f) Distribution of p-values over 100 clustering initializations.

subspace, using these features alone would not necessarily be expected to boost performance on downstream tasks. Furthermore, the features assigned to the high-d component of the model likely correspond to genes that display uncommon activity patterns, which is why they cannot be easily represented by the same low-rank structure as the other genes. Based on this reasoning, we hypothesized that, rather than being unimportant, some of these genes may actually have very important biological functions. This is particularly likely in the case of cancer data, since genes that are mutated in cancerous cells display highly aberrant activity that disrupts their normal correlations with other genes.

To test this hypothesis, we investigated whether genes assigned to the high-d component in HSL are enriched for oncogenes when the model is run on cancerous samples but not enriched for oncogenes when it is run on samples of healthy tissue. For this experiment, we used the breast cancer gene expression data with matching control samples. After estimating the latent subspaces, we identified gene ontology (GO) terms by performing an enrichment analysis [80] of the features comprising the high-d component, and identified known oncogenes [81] in the subsets. For both cancer and control samples, the three GO terms with the lowest p-value for each dataset, and their contained oncogenes, are shown in Table 4.4.

From these results, we see that HSL identifies a significant number of oncogenes when trained on tumor samples but selects non-oncogenic genes when trained on the healthy control samples. Notably, the high-d component estimated from the breast cancer tumor dataset selected features involved in the regulation of Interleukin-4, an enzyme that is known to be key in the growth of human breast cancer tumors [82]. In contrast, the high-d component learned from a control group did not include those features, instead assigning them to the low-rank space. In addition, the high-

Table 4.4: Differential Enrichment of the Features Assigned to High-Dimensional Components

Data Type	Gene Ontology Term	Selected Oncogenes
Tumor	interleukin-4 production	LEF1, CD83
	nucleoside-triphosphatase activity	TCIRG1, RAB31, ATP6V1C1, ATP6V1G3
	protein binding	NTRK3, HSPA1A, CCR5, ITGA2 + 10 more
Control	snRNA 3'-end processing	None
	epidermal growth factor receptor activity	ERRFI1, PSEN1
	acrosomal vesical exocytosis	None

d component for the cancerous samples is enriched for the GO term "nucleoside-triphosphatase activity", which includes both ATPase and GTPase activity. These processes are involved in regulation of the cell metabolism, a central mechanism in tumor growth [83]. Once again, the hybrid model assigned these features to the low-r component for non-cancerous samples. As the two datasets share the same set of features, the differential enrichment of oncogenes in the high-d component suggests that the assignment of features to either high-d or low-r component reflects characteristics of the original data.

Finally, we studied whether the subspaces estimated by HSL are more useful for downstream analysis than those of competing methods. To do this, we clustered the low-rank embeddings estimated from gene expression levels of both tumor and control samples into two groups using k-means. As seen in Figure 4.6, clusters formed in the subspace estimated by HSL have more differential survival patterns than clusters formed in the subspaces estimated by traditional methods. While the survival effect size is not large, HSL is the only dimensionality reduction technique that retains enough information to produce survival curves that are different at a significance level of $p < .05$. This indicates that the subspace estimated by HSL is both efficient and retains information for downstream analysis.

4.5 Discussion

In this work, we present a new subspace learning model that employs a novel regularization scheme to estimate a partial low-dimensional latent space embedding of a high-dimensional dataset and simultaneously identify features that do not easily fit in a low-rank space. This model addresses a critical gap in the existing literature on subspace learning, in which it is usually assumed that the high-dimensional data can be completely captured by a low-rank approximation, modulo some noise.

By comparing the singular value decompositions of real and synthetic datasets, we demonstrate that this assumption is not fulfilled in many real datasets. We therefore argue that our model is more appropriate for subspace learning on high-dimensional datasets that have a long-tailed singular value spectrum. Through applications to synthetic data, a video background subtraction task, and real gene expression data, we demonstrate that hybrid subspace learning can effectively learn a low-rank latent structure while assigning meaningful features to the high-dimensional component.

Chapter 5

Proposed Work

5.1 Problem

One of the most widespread problems currently facing genomic data is the curse of dimensionality, which occurs when a dataset contains many more features than samples. In this setting, accurate statistical inference is tremendously challenging, which is why dimensionality reduction techniques are often necessary and simple models tend to work best.

In the last part of this thesis, we propose to address this problem by designing methods that can learn from a dataset of heterogeneous samples obtained by combining multiple related datasets. Our core assumption is that the samples are identical and independently distributed (i.i.d.) within each dataset, but not necessarily across datasets. Furthermore, we assume that all samples have measurements for the same set of features and labels.

In particular, we will focus on developing methods that can learn from cancer genomic data by combining samples across cancer types. Within this domain, we plan to address two distinct but closely related problems: (1) Learning compact yet informative representations of cancer genomic data that can be used for a wide range of downstream tasks; (2) Predicting survival rates for cancer patients using censored outcome data. The first of these problems is unsupervised, whereas the second is supervised. Our aim is to develop one or more methods to address both of these problems, either together or separately.

Prior work in this area has been fairly limited, but has nonetheless shown significant promise. In an analysis of the utility of genomic data in predicting patient survival for 4 different cancer types, [84] found that a survival model trained on data from patients with ovarian cancer was more predictive of the survival rates of patients with kidney cancer than a model trained on kidney cancer patients themselves. Based on additional experiments, the authors hypothesized that the performance gain was primarily due to the larger sample size of the ovarian cancer training set. In a separate study of predicting the survival rates of breast cancer patients using a deep learning approach, [85] compared models trained purely on breast cancer data with those trained on data from multiple cancer types (including breast cancer, ovarian cancer, and uterine cancer). The authors found that the model trained on data combined from all three cancer types achieved the best performance when predicting breast cancer survival, and this result was consistent across multiple feature sets and model types.

These approaches have focused directly on fitting survival models by combining data or sharing information across multiple cancer types. There have also been successful approaches that first learn an informative low-dimensional representation of genomic data in a completely unsupervised manner by integrating datasets across cancer types, and then use the latent representation to train separate survival models for individual cancer types [86; 87]. One of these approaches led to the

Table 5.1: TCGA Dataset

Cancer Type	Sample Size by Data Type			
	Somatic Mutation	Gene Expression	Copy Number	Survival (Censored)
Breast Invasive Carcinoma	1,044	1,092	1,096	1,096 (945)
Glioblastoma Multiforme	396	166	593	596 (105)
Ovarian Serous Cystadenocarcinoma	443	376	573	584 (236)
Lung Adenocarcinoma	569	515	518	513 (329)
Uterine Corpus Endometrial Carcinoma	542	555	547	547 (456)
Kidney Renal Clear Cell Carcinoma	339	530	532	537 (360)
Head and Neck Squamous Cell Carcinoma	510	501	521	527 (304)
Brain Lower Grade Glioma	513	511	514	514 (389)
Thyroid Carcinoma	496	502	505	507 (491)
Lung Squamous Cell Carcinoma	497	501	504	498 (283)
Prostate Adenocarcinoma	498	495	498	500 (490)
Colon Adenocarcinoma	433	456	458	458 (356)
Stomach Adenocarcinoma	441	380	443	438 (268)
Bladder Urothelial Carcinoma	412	408	412	411 (231)
Liver Hepatocellular Carcinoma	375	371	376	376 (244)
Total	7,508	7,359	8,090	8,102 (5,487)

winning submission in the Sage Bionetworks DREAM Breast Cancer Prognosis Challenge [88].

Although this existing work has already demonstrated success on a variety of survival prediction tasks, it has mainly focused on identifying generic patterns that persist across all cancer types. For final project in this thesis, we will aim to identify patterns that generalize across cancer types while also contrasting the differences between cancer types to identify clinically meaningful patterns that are specific to individual cancer types (using, for example, ideas from [89]).

5.2 Dataset

To facilitate this work, we have collected a large dataset from The Cancer Genome Atlas (TCGA) that spans 15 cancer types with more than 8,000 patients. For each patient, we downloaded and preprocessed 3 types of genomic data (somatic mutations, gene expression, copy number variation) along with survival outcomes, many of which are censored. The dataset is summarized in Table 5.1.

Since this dataset as a whole contains a much larger number of samples than what is typically used for fitting survival models,[1] we hope to use it to achieve state of the art performance on survival prediction tasks. Furthermore, having data for such a large number of cancer types provides a unique opportunity to investigate the impact of combining samples from heterogeneous sources.

[1]The model in [85] that combined three different cancer types was trained on 924 samples. The METABRIC breast cancer dataset used for the DREAM challenge, which is considered relatively large, contained 1,981 samples.

5.3 Timeline

An approximate timeline for carrying out the proposed work and completing this thesis is provided in Table 5.2.

Table 5.2: Timeline

Semester	Tasks
Winter 2018	data collection and preprocessing for cancer project; revisions for ICL paper
Spring 2018	method development and experiments for cancer project; revisions for HSL paper
Summer 2018	wrap up cancer project; write thesis document
Fall 2018	finish any remaining work; complete thesis defense

Bibliography

[1] ZD Stephens, SY Lee, F Faghri, et al. Big data: astronomical or genomical? *PLoS Biology*, 13(7):e1002195, 2015.

[2] LA Hindorff, J MacArthur, J Morales, et al. A catalog of published genome-wide association studies, 2015. URL: www.genome.gov/gwastudies.

[3] GM Clarke, CA Anderson, FH Pettersson, et al. Basic statistical analysis in genetic case-control studies. *Nature Protocols*, 6(2):121–133, 2011.

[4] J Li, K Das, G Fu, R Li, and R Wu. The Bayesian lasso for genome-wide association studies. *Bioinformatics*, 27(4):516–523, 2011.

[5] TT Wu, YF Chen, T Hastie, E Sobel, and K Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, 2009.

[6] K Das, J Li, Z Wang, et al. A dynamic model for genome-wide association studies. *Human Genetics*, 129(6):629–639, 2011.

[7] J Li, Z Wang, R Li, and R Wu. Bayesian group lasso for nonparametric varying-coefficient models with application to functional genome-wide association studies. *The Annals of Applied Statistics*, 9(2):640–664, 2015.

[8] J Yang, R Wu, and G Casella. Nonparametric functional mapping of quantitative trait loci. *Biometrics*, 65(1):30–39, 2009.

[9] NA Furlotte, E Eskin, and S Eyheramendy. Genome-wide association mapping with longitudinal data. *Genetic Epidemiology*, 36(5):463–471, 2012.

[10] K Das, J Li, G Fu, et al. Dynamic semiparametric Bayesian models for genetic mapping of complex trait with irregular longitudinal data. *Statistics in Medicine*, 32(3):509–523, 2013.

[11] Z Li and MJ Sillanpää. A Bayesian nonparametric approach for mapping dynamic quantitative traits. *Genetics*, 194(4):997–1016, 2013.

[12] J Yin, X Chen, and EP Xing. Group sparse additive models. In *International Conference on Machine Learning (ICML)*, pp. 871–878, 2012.

[13] T Hastie and R Tibshirani. *Generalized Additive Models*. Chapman & Hall, 1990.

[14] M Yuan and Y Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[15] X Zhang, BU Park, and JL Wang. Time-varying additive models for longitudinal data. *Journal of the American Statistical Association*, 108(503):983–998, 2013.

[16] S Purcell. PLINK 1.07, 2009. URL: http://pngu.mgh.harvard.edu/purcell/plink/.

[17] J Liu, S Ji, and J Ye. SLEP, 2009. URL: http://www.public.asu.edu/~jye02/Software/SLEP.

[18] Z Wang and J Li. fGWAS2, 2012. URL: http://statgen.psu.edu/software/fgwas-r2.html.

[19] J Batra and B Ghosh. Genetic contribution of chemokine receptor 2 (CCR2) polymorphisms towards increased serum total IgE levels in Indian asthmatics. *Genomics*, 94(3):161–168, 2009.

[20] Entrez Gene Database, 2005. URL: http://www.ncbi.nlm.nih.gov/gene.

[21] MQ Zhang and H Timmerman. Mast cell tryptase and asthma. *Mediators of Inflammation*, 6(5-6):311–317, 1997.

[22] ML Manni, KM Robinson, and JF Alcorn. A tale of two cytokines: IL-17 and IL-22 in asthma and infection. *Expert Review of Respiratory Medicine*, 8(1):25–42, 2014.

[23] C Ober and TC Yao. The genetics of asthma and allergic disease: a 21st century perspective. *Immunological Reviews*, 242(1):10–30, 2011.

[24] SH Oh, YH Kim, SM Park, et al. Association analysis of thromboxane a synthase 1 gene polymorphisms with aspirin intolerance in asthmatic patients. *Pharmacogenomics*, 12(3):351–363, 2011.

[25] T Liu, TM Laidlaw, C Feng, et al. Prostaglandin E2 deficiency uncovers a dominant role for thromboxane A2 in house dust mite-induced allergic pulmonary inflammation. *Proceedings of the National Academy of Sciences*, 109(31):12692–12697, 2012.

[26] P Gao, H Kawada, T Kasamatsu, et al. Variants of NOS1, NOS2, and NOS3 genes in asthmatics. *Biochemical and Biophysical Research Communications*, 267(3):761–763, 2000.

[27] J You, W Peng, X Lin, QL Huang, and JY Lin. PLC/CAMK IV–NF-κB involved in the receptor for advanced glycation end products mediated signaling pathway in human endothelial cells. *Molecular and Cellular Endocrinology*, 320(1):111–117, 2010.

[28] YJ Lin, JS Chang, X Liu, H Tsang, et al. Genetic variants in PLCB4/PLCB1 as susceptibility loci for coronary artery aneurysm formation in Kawasaki disease in Han Chinese in Taiwan. *Scientific Reports*, 5, 2015.

[29] MY Yang, MB Hilton, S Seaman, et al. Essential regulation of lung surfactant homeostasis by the orphan G protein-coupled receptor GPR116. *Cell Reports*, 3(5):1457–1464, 2013.

[30] C Venkataraman, K Justen, J Zhao, E Galbreath, and S Na. Death receptor-6 regulates the development of pulmonary eosinophilia and airway inflammation in a mouse model of asthma. *Immunology Letters*, 106(1):42–47, 2006.

[31] M Siedlinski, CC van Diemen, DS Postma, JM Vonk, and HM Boezen. Superoxide dismutases, lung function and bronchial responsiveness in a general population. *European Respiratory Journal*, 33(5):986–992, 2009.

[32] JA Mathews, J Ford, S Norton, et al. A potential new target for asthma therapy: A Disintegrin and Metalloprotease 10 (ADAM10) involvement in murine experimental asthma. *Allergy*, 66(9):1193–1200, 2011.

[33] K Nagpal, S Sharma, BR Chandrika, et al. TGFβ1 haplotypes and asthma in Indian populations. *Journal of Allergy and Clinical Immunology*, 115(3):527–533, 2005.

[34] Y Bossé. Updates on the COPD gene list. *International Journal of Chronic Obstructive Pulmonary Disease*, 7:607, 2012.

[35] P Van Eerdewegh, RD Little, J Dupuis, et al. Association of the ADAM33 gene with asthma and bronchial hyperresponsiveness. *Nature*, 418(6896):426–430, 2002.

[36] C Ober and S Hoffjan. Asthma genetics 2006: the long and winding road to gene discovery. *Genes and Immunity*, 7(2):95–100, 2006.

[37] MA Ferreira, L O'Gorman, P Le Souëf, et al. Robust estimation of experimentwise p values applied to a genome scan of multiple asthma traits identifies a new region of significant linkage on chromosome 20q13. *The American Journal of Human Genetics*, 77(6):1075–1085, 2005.

[38] CAMP Research Group et al. The childhood asthma management program (CAMP): design, rationale, and methods. *Controlled Clinical Trials*, 20(1):91–120, 1999.

[39] M Bijanzadeh, PA Mahesh, et al. An understanding of the genetic basis of asthma. *The Indian Journal of Medical Research*, 134(2):149, 2011.

[40] MV Rockman and L Kruglyak. Genetics of global gene expression. *Nature Reviews Genetics*, 7(11):862–872, 2006.

[41] TS Gardner and JJ Faith. Reverse-engineering transcription control networks. *Physics of Life Reviews*, 2(1):65–88, 2005.

[42] AL Barabasi and ZN Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

[43] R Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58(1):267–288, 1996.

[44] S Kim, KA Sohn, and EP Xing. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, 25(12):i204–i212, 2009.

[45] S Kim and EP Xing. Statistical estimation of correlated genome associations to a quantitative trait network. *PLoS Genetics*, 5(8):e1000587, 2009.

[46] X Chen, S Kim, Q Lin, JG Carbonell, and EP Xing. Graph-structured multi-task regression and an efficient optimization method for general fused lasso. *arXiv preprint*, arXiv:1005.3579, 2010.

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

[48] CM Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[49] J Peng, P Wang, N Zhou, and J Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746, 2009.

[50] AJ Rothman, E Levina, and J Zhu. Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 19(4):947–962, 2010.

[51] KA Sohn and S Kim. Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1081–1089, 2012.

[52] M Wytock and Z Kolter. Sparse Gaussian conditional random fields: algorithms, theory, and application to energy forecasting. In *International Conference on Machine Learning (ICML)*, pp. 1265–1273, 2013.

[53] XT Yuan and T Zhang. Partial Gaussian graphical model estimation. *IEEE Transactions on Information Theory*, 60(3):1673–1687, 2014.

[54] Y Yu. Better approximation and faster algorithm using the proximal average. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 458–466, 2013.

[55] RB Brem and L Kruglyak. The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proceedings of the National Academy of Sciences*, 102(5):1572–1577, 2005.

[56] A Subramanian, P Tamayo, VK Mootha, et al. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, 2005.

[57] M Ashburner, CA Ball, JA Blake, et al. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[58] M Kanehisa and S Goto. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

[59] JS Breunig, SR Hackett, JD Rabinowitz, and L Kruglyak. Genetic basis of metabolome variation in yeast. *PLoS Genetics*, 10(3):e1004142, 2014.

[60] CJ Roberts and EU Selker. Mutations affecting the biosynthesis of S-adenosylmethionine cause reduction of DNA methylation in Neurospora crassa. *Nucleic Acids Research*, 23(23):4818–4826, 1995.

[61] B Zhang, C Gaiteri, LG Bodea, et al. Integrated systems approach identifies genetic nodes and networks in late-onset Alzheimer's disease. *Cell*, 153(3):707–720, 2013.

[62] M Malik, J Chiles III, HS Xi, et al. Genetics of CD33 in Alzheimer's disease and acute myeloid leukemia. *Human Molecular Genetics*, 24(12):3557–3570, 2015.

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

[64] P Danaher, P Wang, and DM Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.

[65] V Marx. Biology: The big challenges of big data. *Nature*, 498(7453):255–260, 2013.

[66] GP Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1):55–63, 1968.

[67] I Jolliffe. *Principal Component Analysis*. Wiley Online Library, 2002.

[68] H Zou, T Hastie, and R Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

[69] EJ Candès, X Li, Y Ma, and J Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[70] RT Dorsam and JS Gutkind. G-protein-coupled receptors and cancer. *Nature Reviews Cancer*, 7(2):79–94, 2007.

[71] H Xu, C Caramanis, and S Sanghavi. Robust PCA via outlier pursuit. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2496–2504, 2010.

[72] A Beck and M Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[73] R Mazumder, JH Friedman, and T Hastie. SparseNet: coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495):1125–1138, 2011.

[74] KP Burnham and DR Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer Science & Business Media, 2003.

[75] K Voduc, M Cheang, S Tyldesley, et al. Breast cancer subtypes and the risk of local and regional relapse. *Journal of Clinical Oncology*, 28(10):1684–1691, 2010.

[76] R Verhaak et al. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell*, 17(1):98–110, 2010.

[77] J Guinney, R Dienstmann, X Wang, et al. The consensus molecular subtypes of colorectal cancer. *Nature Medicine*, 21(11):1350–1356, 2015.

[78] SR Prasad, PA Humphrey, JR Catena, et al. Common and uncommon histologic subtypes of renal cell carcinoma: imaging spectrum with pathologic correlation. *RadioGraphics*, 26(6):1795–1806, 2006.

[79] L West, SJ Vidwans, NP Campbell, et al. A novel classification of lung cancer into molecular subtypes. *PLoS ONE*, 7(2):1–11, 2012.

[80] E Eden, R Navon, I Steinfeld, D Lipson, and Z Yakhini. GOrilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists. *BMC Bioinformatics*, 10(1):48, 2009.

[81] GF Berriz, OD King, B Bryant, C Sander, and FP Roth. Characterizing gene sets with FuncAssociate. *Bioinformatics*, 19(18):2502–2504, 2003.

[82] S Nagai and M Toi. Interleukin-4 and breast cancer. *BMC Bioinformatics*, 7(3):181–186, 2000.

[83] RA Cairns, IS Harris, and TW Mak. Regulation of cancer cell metabolism. *Nature Reviews Cancer*, 11(2):85–95, 2011.

[84] Y Yuan, EM Van Allen, L Omberg, et al. Assessing the clinical utility of cancer genomic and proteomic data across tumor types. *Nature Biotechnology*, 32(7):644–652, 2014.

[85] S Yousefi, F Amrollahi, M Amgad, et al. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Scientific Reports*, 7(1), 2017.

[86] WY Cheng, THO Yang, and D Anastassiou. Biomolecular events in cancer revealed by attractor metagenes. *PLoS Computational Biology*, 9(2):e1002920, 2013.

[87] S Celik, BA Logsdon, S Battle, et al. Extracting a low-dimensional description of multiple gene expression datasets reveals a potential driver for tumor-associated stroma in ovarian cancer. *Genome Medicine*, 8(1):66, 2016.

[88] WY Cheng, THO Yang, and D Anastassiou. Development of a prognostic model for breast cancer survival in an open challenge environment. *Science Translational Medicine*, 5(181):181ra50–181ra50, 2013.

[89] A Abid, VK Bagaria, MJ Zhang, and J Zou. Contrastive principal component analysis. *arXiv preprint*, arXiv:1709.06716, 2017.