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

Micol Marchetti-Bowick

May 2020
CMU-ML-20-105

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Eric P. Xing, Chair
Jian Ma
Seyoung Kim
Su-In Lee

Submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy.

This research was supported by the National Science Foundation award number DGE1252522 and by the National Institutes of Health award numbers RGM093156 and P30DA035778.

Key Words: structured sparse regression, high-dimensional multivariate regression, computational genomics, GWAS, eQTL mapping, gene network estimation, pan-cancer survival analysis

To my parents, Cristina Marchetti and Mark Bowick

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 yields increased statistical power to detect associations between different components of a biological system (e.g., SNPs and genes). This in turn provides greater insight into complex biological processes and more accurate predictions of disease phenotypes, ultimately leading to improved diagnosis and treatment of human diseases.

Acknowledgements

First, I would like to thank my advisor, Eric Xing, for guidance and support throughout my PhD. Thank you also to my other committee members, Jian Ma, Seyoung Kim, and Su-In Lee, for their encouragement and helpful discussions during the last part of this thesis. I have also greatly enjoyed collaborating with Wei Wu, Yaoliang Yu, Ankur Parikh, and Ben Lengerich over the years. Thank you for your ideas and insights.

The highlights of my time at CMU were always spent with my MLD classmates, Junier Oliva, Willie Neiswanger, Kirstin Early, Nicole Rafidi, Benjo Cowley, and David Wei Dai. Thank you for making me laugh. I also want to thank Avinava Dubey, Zhe Zhang, Matt Barnes, Calvin Murdock, Maria De-Arteaga, Benedikt Boecking, and Jing Xiang for additional laughter and friendship over the years in Pittsburgh.

The last few years of my PhD were completed while I was on leave of absence and working at Uber ATG. I want to sincerely thank all of my amazing colleagues at ATG for making our work both meaningful and fun, but I am particularly grateful to my past and current managers, Jeff Schneider and Clark Haynes, for their enthusiastic support of finishing this thesis.

To my family, thank you for everything you do. I would especially like to thank my parents for being an example to aspire to, and wish good luck to my sister, who is starting her own PhD journey just as I am finishing mine.

Finally, but most importantly, thank you to Tim, without whose unwavering love, support, and encouragement I would never have made it this far.

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 [81].

The information contained within these datasets has the potential to answer age-old questions about how the human body works. 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 studying the problem manually 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.

$$\min_{\theta} \ \text{loss}(\theta; x) + \lambda \ \text{sparsity-penalty}(\theta) + \gamma \ \text{structure-penalty}(\theta; \alpha) \tag{1.1}$$

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. Many existing models, including many variants of the classic Lasso, 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 main body of this thesis, 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 [56]. 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 [58]. 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 [57]. 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.

In the last part of this thesis, we take a step back and re-examine the premise, namely, that incorporating structure into statistical models can help boost the signal-to-noise ratio when working with biological datasets. To do this, we focus on predicting cancer survival rates from gene expression data, using a pan-cancer dataset comprised of patients with 10 distinct types of cancer.

This is an ideal problem to study because the statistical challenges that arise from high data heterogeneity and low sample sizes are particularly extreme in this setting, for two reasons. First, malignant cancers have extremely high molecular heterogeneity, even within a single tumor in a single individual [59]. This is due in part to the nature of cancerous growth, where runaway cell multiplication provides ample opportunities for mutations to arise and accumulate. This problem is further exacerbated by aggregating data from patients with multiple cancer types that originate in completely different organs within the body. Second, although most genomic datasets are already high-dimensional, containing many more dimensions of variation (e.g. genes in a gene expression dataset) than samples (e.g. cancer patients), survival prediction suffers from the additional challenge of data censorship, in which the survival outcome is only observed for a subset of patients [46]. The patients that survive past the end of the study or stop responding to follow-up requests have a censored outcome, which means that we only observe a lower bound on their survival time. Learning from censored data is particularly challenging because we have incomplete information about many of the samples.

Using the pan-cancer dataset, we perform an empirical analysis in which we share information across the cancer types and impose increasingly strict structural assumptions about the relationships between the cancers. We examine the effects of transfer learning on survival prediction and ultimately demonstrate that using structured sparsity penalties to share information across cancer types has two significant benefits: it leads to better performance on the survival prediction task, and it reveals the rich and intricate structure of the problem being studied. We therefore conclude that structured sparsity is not only useful for boosting the signal-to-noise ratio for prediction tasks, but also leads to greater understanding of the underlying biological mechanisms.

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 [40]. 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 [21]. 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 [48, 91]. 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

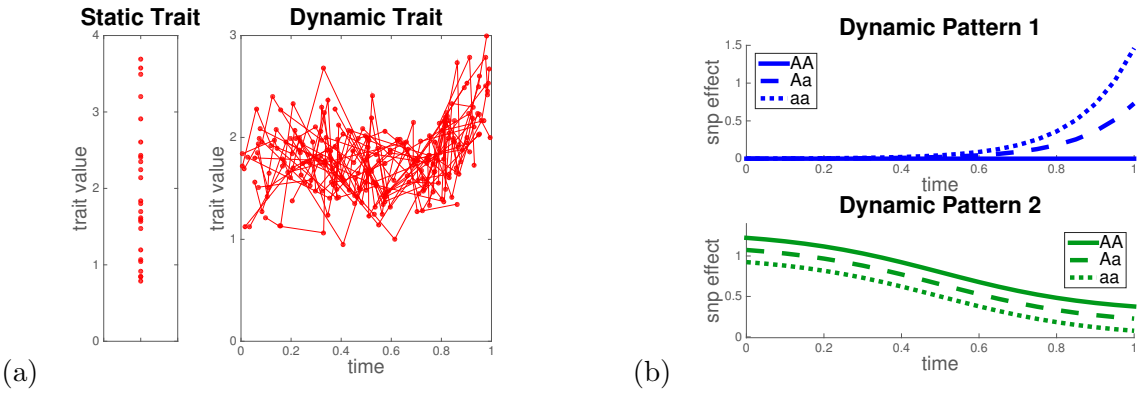


Figure 2.1: GWAS has greater statistical power with dynamic traits. (a) A toy dataset illustrating the difference between static and dynamic traits. (b) Two synthetic examples of time-dependent patterns of SNP influence on the trait that would be difficult to detect with a static model.

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 [25], 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 [49]. 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 [95], [32], [26], and [50]. 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 a Time-Varying Group Sparse Additive Model, or TV-GroupSpAM. Our method is based on GroupSpAM [97], 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 [25, 50].

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 [39] 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 [101]. 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 [97], in which a group sparse regularization penalty is applied to a standard additive model, and the Time-Varying Additive Model of [106], 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 [39]. 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 [39] 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 [97] 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 [97] 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 $j = 1, \ldots, p$

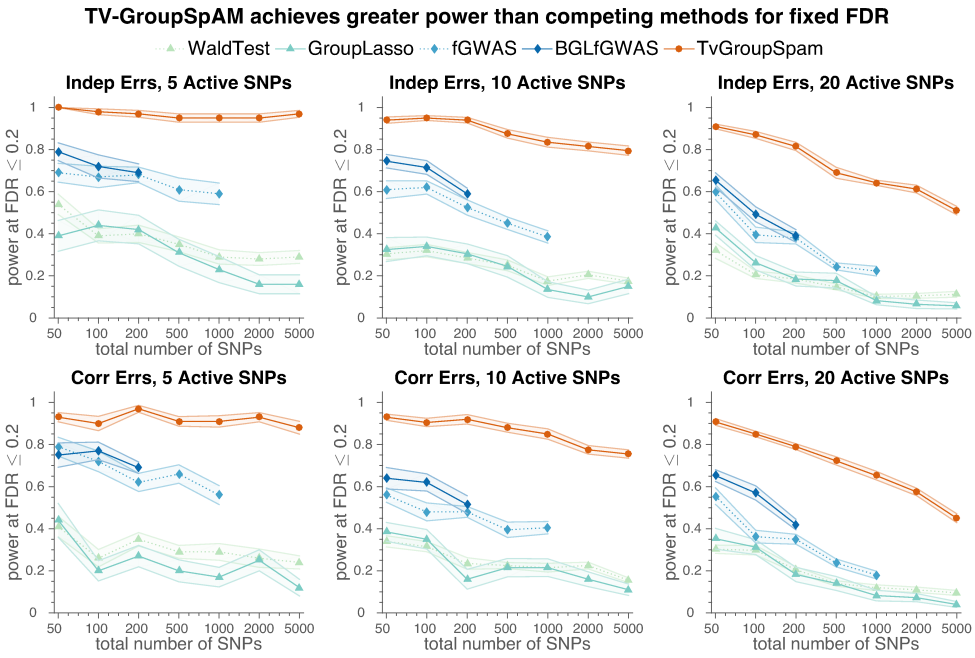


Figure 2.2: Comparison of TV-GroupSpAM to baseline methods shows that our approach achieves greater power for a fixed false discovery rate (FDR ≤ 0.2). The results are averaged over 20 random synthetic datasets for each setting, and the shaded region denotes the standard error.

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 [74] for the Wald test, the SLEP Matlab package [52] for lasso and group lasso, and the fGWAS2 R package [89] 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

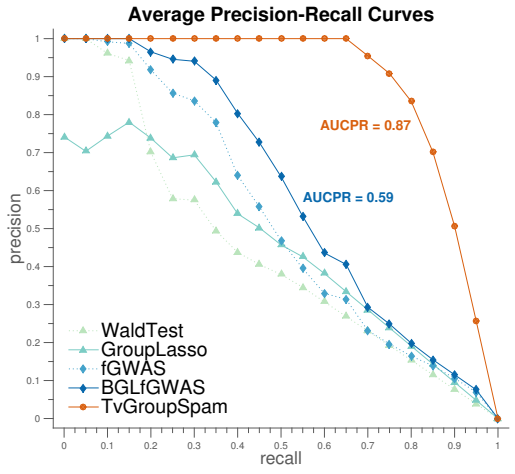


Figure 2.3: Comparison of precision-recall curves of TV-GroupSpAM to baseline methods shows that our approach has an average AUCPR of 0.87 ± 0.01, which is much higher than the most competitive baseline, BGL-fGWAS, which has average AUCPR 0.59 ± 0.02.

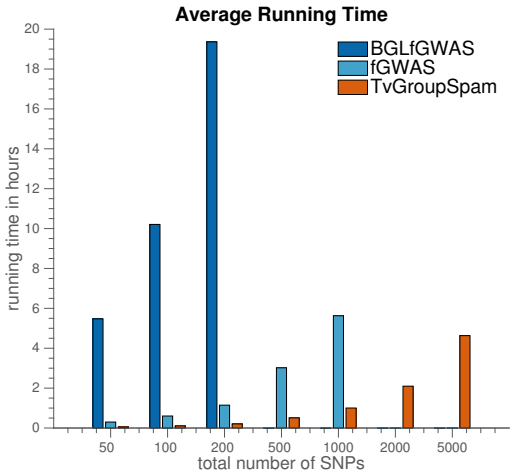


Figure 2.4: Comparison of the running time of TV-GroupSpAM to baseline methods shows that our approach runs much faster than both fGWAS and BGL-fGWAS. We were unable to run fGWAS for $p > 1000$ or BGL-fGWAS for $p > 200$ due to time constraints.

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 TV-GroupSpAM. 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 $p = 200$, TV-GroupSpAM 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 $1-\mathrm{precision} = \mathrm{FP}/(\mathrm{TP}+\mathrm{FP})$.

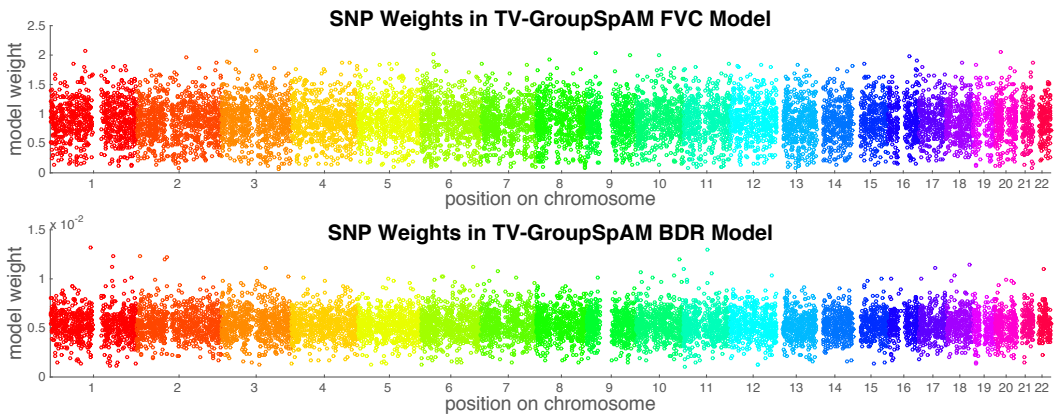


Figure 2.5: Manhattan plots of the model weights for each SNP that was selected in the FVC model (top) and BDR model (bottom) during the filtering stage.

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 [5]; CCR3 may play a role in airway inflammation [1] *PRSS42, PRSS46, PRSS45, PRSS50* – trypsin-like serine proteases; tryptases cause bronchoconstriction and have been implicated in asthma [105]
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 [55]
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 [69]
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 [70]
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 [53]
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 [33]
rs6056242	20	8.8 Mb	1.2298	*PLCB4* – involved in the endothelial cell signaling pathway [98] and plays a role in vascular inflammation [51]

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 [96]
				TNFRSF21 – tumor necrosis factor receptor superfamily member 21; plays a central role in regulating immune response and airway inflammation in mice [86]
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 [79]
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 [62]
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 [66]
				CYP2A6, CYP2A7, RAB4B, MIA, EGLND – genes located in a known COPD locus [10]
rs6116189	20	4.0 Mb	0.0067	*ADAM33* – disintegrin and metalloproteinase domain-containing protein 33; has been implicated in asthma by several independent studies [85, 68]
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 [29]

bronchodilator response (BDR), which measures lung response to bronchodilator drugs. For this analysis, we use data from the CAMP longitudinal study of childhood asthma [75] 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 [91]. 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.

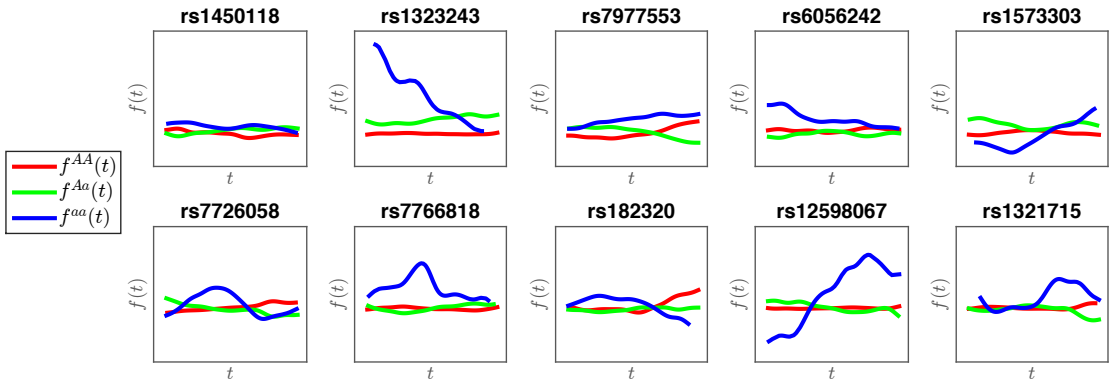


Figure 2.6: Examples of estimated dynamic SNP effects. Top row shows five SNPs selected from FVC model. Bottom row shows five SNPs selected from BDR model.

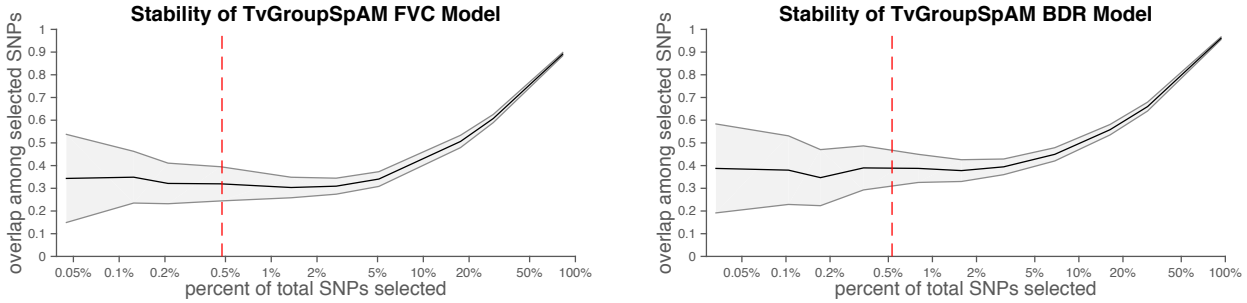


Figure 2.7: Average stability of the FVC model (left) and BDR model (right) for different fractions of selected SNPs. Shaded region shows standard deviation. Red line indicates the fraction of the filtered SNPs that we selected in our final analysis, which yielded 48 SNPs for FVC and 51 SNPs for BDR.

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 [8], 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 [69].

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 [68]. 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 [69]. 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 [70]. 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 ~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 [77], and (b) determining which have a direct influence on the expression of other genes, a problem known as gene network estimation [35]. 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 [4]. 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 [84], 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 [45, 44, 18] 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 [30] 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 [9].

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

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

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 $w_{km} = \lambda_2 + \gamma\|\beta_{\cdot k} + \text{sign}(\theta_{km})\beta_{\cdot m}\|$.

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., [71]). This explains why the sign is flipped in the fusion penalty in (3.8) relative to the one in (3.2).

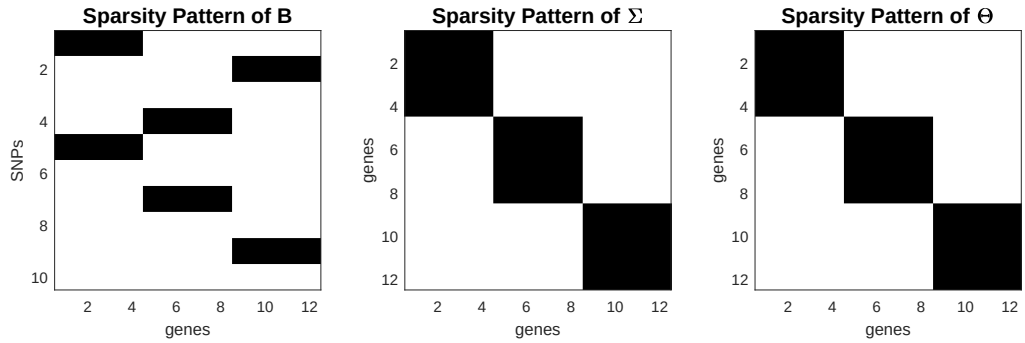


Figure 3.1: A toy example with 10 SNPs and 12 genes grouped into 3 modules. When B exhibits a certain type of sparse structure, $\Sigma = I + B^T B$ and $\Theta = \Sigma^{-1}$ will also be sparse.

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 [78], and conditional Gaussian graphical models (CGGM), originally from [80] and further developed by [92] and [102]. 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 [100]. 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

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

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:

$$\hat{\beta}_{jk}, \hat{\beta}_{jm} = \arg\min_{\beta_{jk}, \beta_{jm}} \frac{1}{2\eta}(\beta_{jk} - z_{jk})^2 + (\beta_{jm} - z_{jm})^2 + |\beta_{jk} + \text{sgn}(\theta_{km})\beta_{jm}|. \tag{3.16}$$

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 [100], 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 [30]. Indeed, we can rewrite the objective by introducing two $q \times q$ dimensional coefficient matrices U and L whose elements are defined as

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

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

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
$\Theta = \Xi^{-1}$

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.

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 traditional pairwise linear regression (LinReg), 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 a traditional pairwise correlation network (Corr) and the graphical lasso (GLasso).

For the two pairwise methods, LinReg and Corr, we use the permutation test proposed by [**?**] to select a global significance threshold that achieves a family-wise type I error rate of at most 5%. Because these methods are quite efficient, it is computationally feasible to perform a sufficient number of permutations to accurately estimate this threshold (we use 2,000 permutations). For all other methods 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 of Y 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. Using this approach, we search over a grid for the best parameter values.

Importantly, when running ICLasso, we actually fix the value of λ_2 to a small non-zero value rather than selecting it via a grid search. This works well in practice when B is not fully sparse, because the fusion penalty already applies shrinkage directly to the parameters of Θ. This also reduces the number of hyperparameters that we need to tune for ICLasso from 3 to 2, which is the same as several of the baselines. Before running all methods, we standardize our data by performing mean centering and variance normalization of X and mean centering of Y.

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 module size (a.k.a. group size) 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 modules and then selecting a random set of s SNPs that will be associated with each module. We also associate each gene with a small number of fixed SNPs (1-2) that are associated with every gene.

Given the sparsity structure, we generate the parameters of the nonzero values of B and Θ as

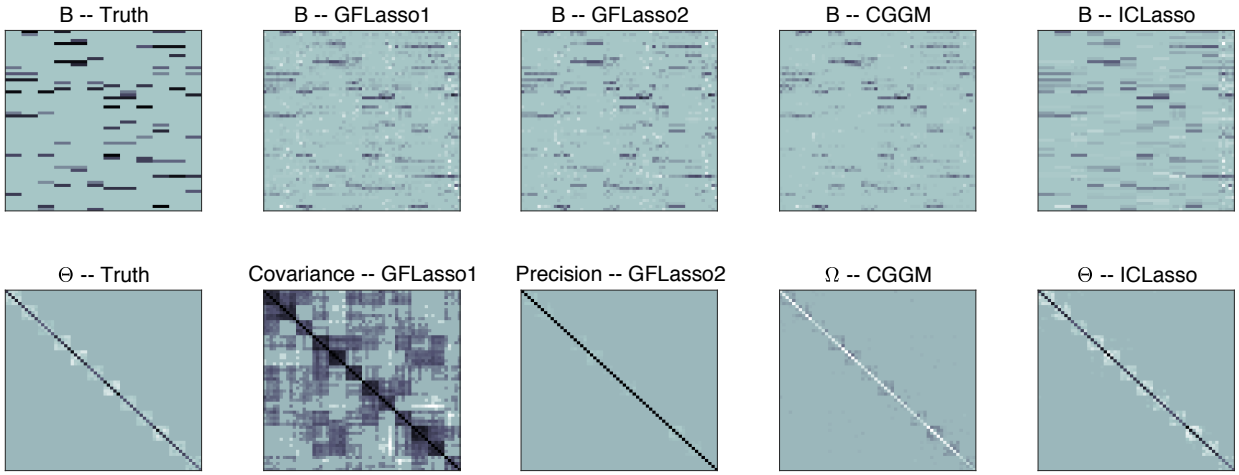


Figure 3.2: A comparison of results on a single synthetic dataset with $p = 60$ and $q = 60$. The far left panel contains the ground truth for B and Θ. The remaining panels show the estimates of the regression coefficients for each method (top) along with the graph structure that was used or estimated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

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 $E \neq I_{q\times q}$, and (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).

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 we conduct, 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,

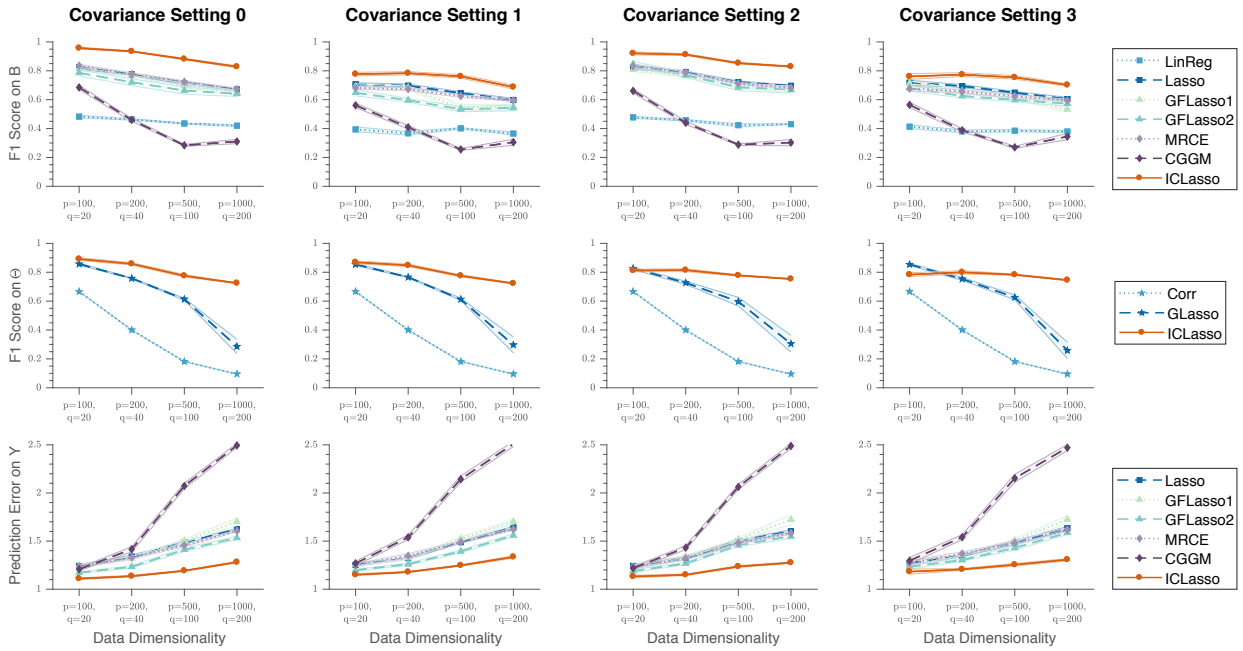


Figure 3.3: A comparison of results on synthetic data generated with each of the four different types of covariance structure and with several different values of p and q. We fix the group size to $g = 10$ and the number of SNP associations per gene to $s = 5$. The top row shows the F1 score on the recovery 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.

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 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 [11] that consists of 2,956 SNP genotypes

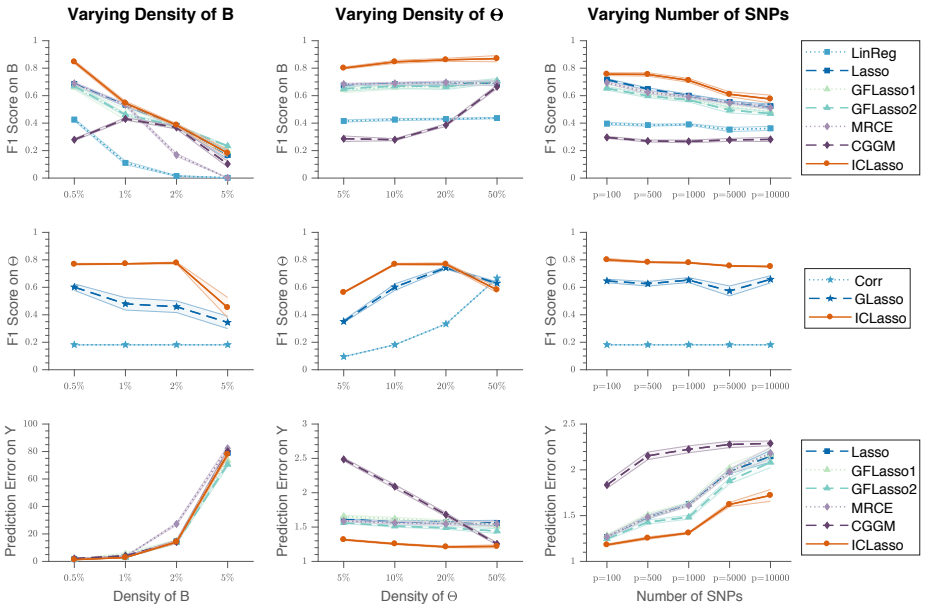


Figure 3.4: A comparison of results on synthetic data generated with different settings. In the first two columns, we use covariance type 0, $p = 1000$, $q = 100$, and vary the density of B and Θ. In the third column, we use covariance type 3, $q = 100$, $g = 10$, $s = 5$, and vary the number of SNPs. 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.

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

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 $\hat{B}$, 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

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**%

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 [82]. 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 [3], and pathway enrichment using annotations from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database [43]. 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.

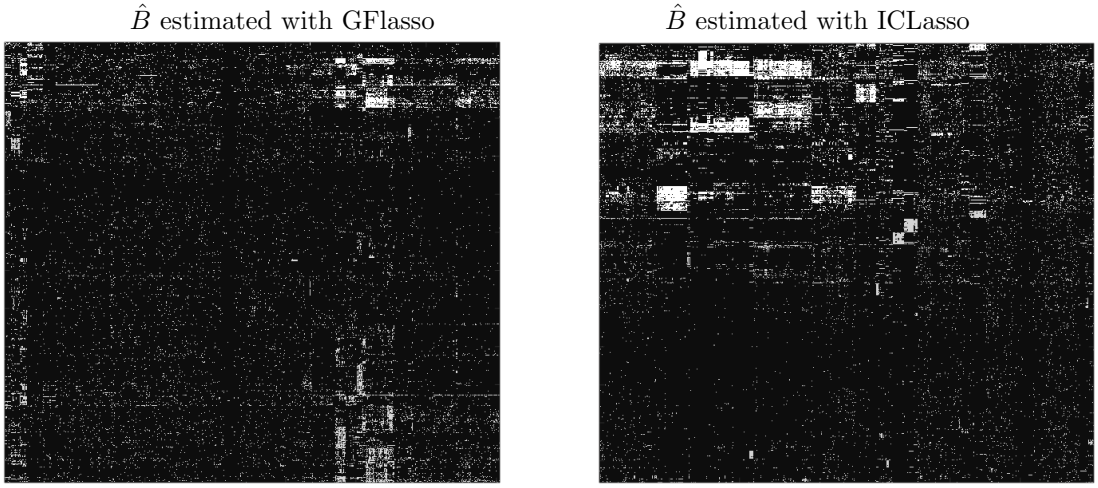


Figure 3.5: Binary heatmap of associations between SNPs (one per row) and genes (one per column), estimated with GFLasso and ICLasso. In each image, the SNPs and genes are ordered to maximize the visual clustering of associations.

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 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. [12] 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

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

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

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 [76].

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.

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 [104] 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 1.70% 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,

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

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

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 [54].

Finally, we compared our results to the two simple pairwise baselines (LinReg and Corr). These methods are commonly used for QTL mapping and gene network estimation, and are often favored for large datasets due to their efficient run time. For the purposes of this experiment, we selected significance thresholds for the baselines that yielded estimates of B and Θ that had the same density as the estimates produced by ICLasso.

The results of our comparative analysis are summarized in Figure 3.6. This plot shows the number of SNP associations per gene sorted by the degree of the gene in the network estimated by ICLasso. The trend lines clearly emphasize that ICLasso has more power to detect eQTL associations for genes that are highly connected in the estimated gene network. This result matches our intuition about the structural prior encoded in the ICLasso penalty term, which enables the model to detect SNP-gene associations that exhibit a weak signal in the data by leveraging information about other related genes. We also note that the gene connectivity estimated by ICLasso correlates well with the connectivity estimated by the correlation network.

To provide an additional view of the results, we plotted the distributions of the SNP and gene association counts in Figure 3.7. One significant difference between ICLasso and the baseline in the distribution of the number of gene associations (shown in the right panel) is that the ICLasso distribution clearly has 3 modes (one peaked at 0, one peaked at 750, and a third peaked at 1500). This suggests that the ICLasso estimate of the gene network identified at least two large interconnected sub-networks.

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 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 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

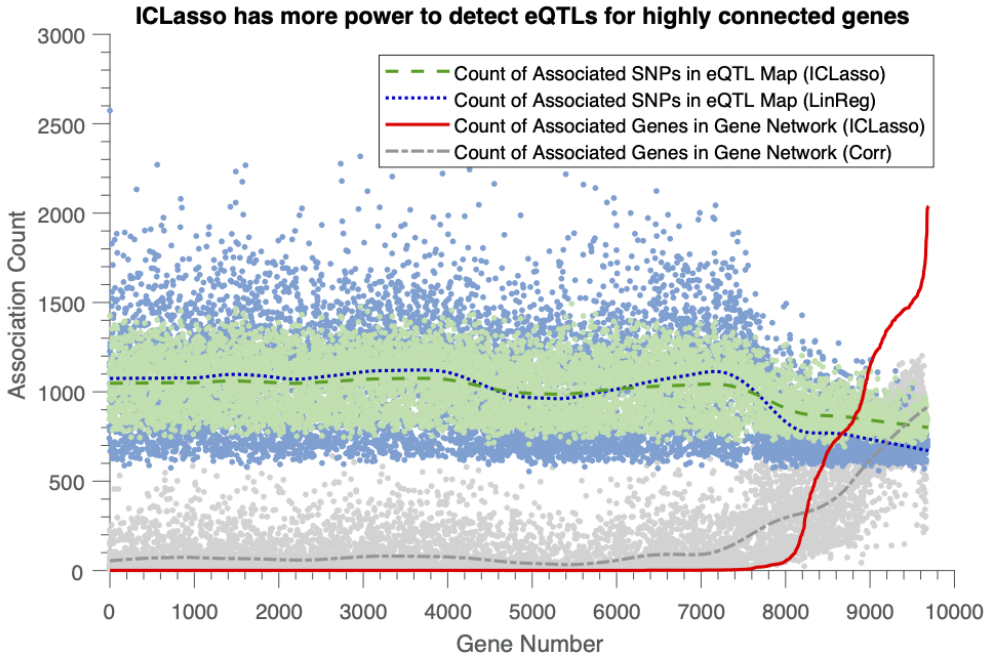


Figure 3.6: A comparison of the results obtained by ICLasso and the pairwise methods on Alzheimer's data. The blue and green lines show the number of SNP associations per gene, and the red and gray lines show the number of gene associations per gene. The genes on the horizontal axis are sorted according to their degree in the network estimated by ICLasso. The three dashed and dotted lines are smoothed versions of the corresponding scatter plots.

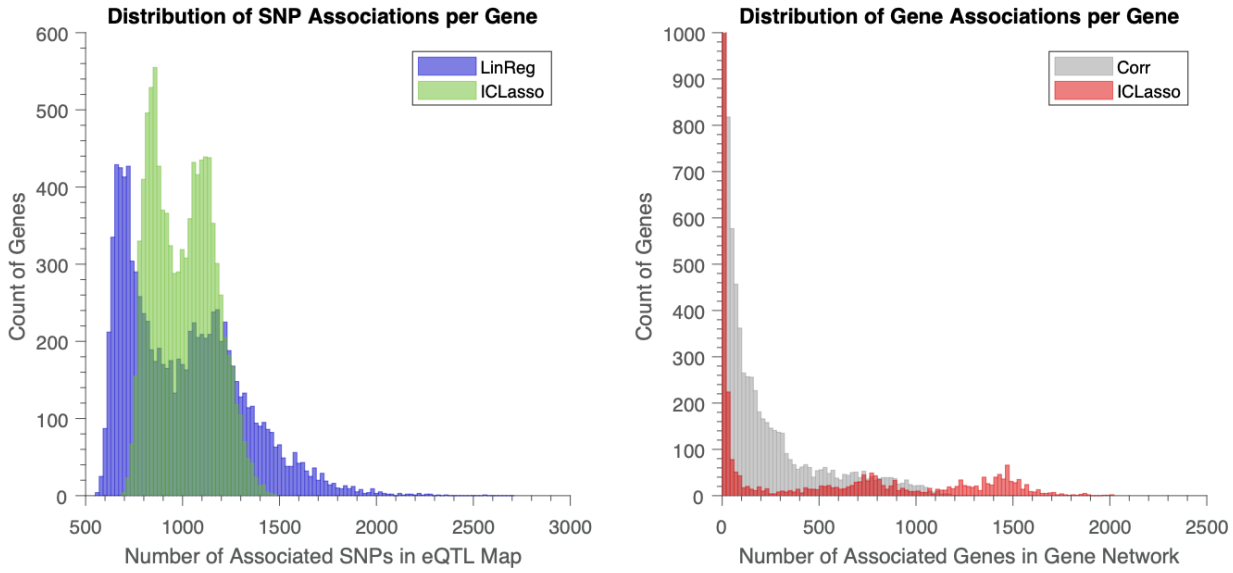


Figure 3.7: A comparison of the distribution of the number of estimated SNP associations per gene (left panel) and the number of estimated gene associations per gene (right panel).

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 [64] 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 [24] 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 [60], 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 [41].

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 [42]. 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 [108]. 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 [15]. Finally, there are approaches that propose richer models for the underlying latent representation of the data, involving multiple subspaces rather than just one [27].

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

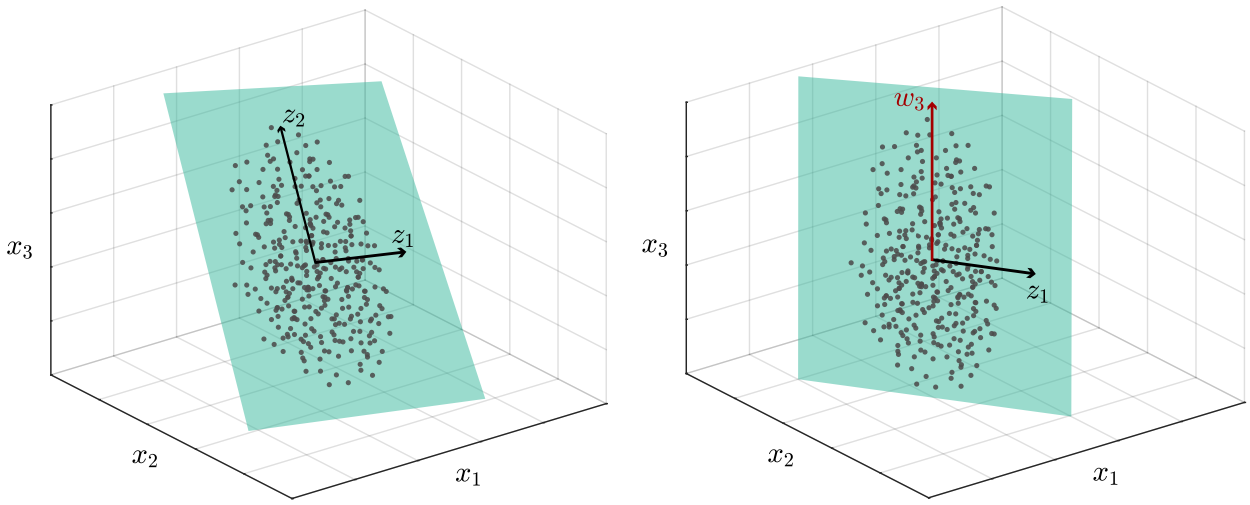


Figure 4.1: Toy datasets that illustrate the difference between fully low-rank data (left) and hybrid data (right). Here z_1 and z_2 represent latent features that are linear combinations of raw features, whereas w_3 represents a latent feature that is perfectly correlated with the raw feature x_3.

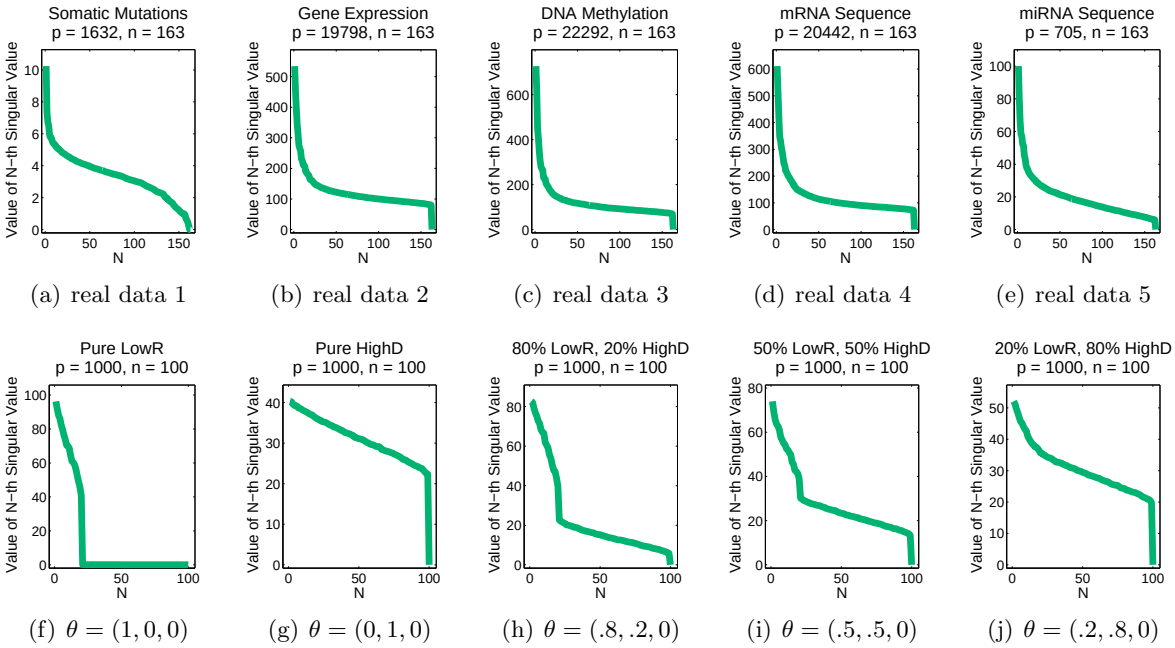
lie on a different 2D plane in 3D space. In the left plot, all three of the raw dimensions exhibit low-rank structure because they are all correlated. However, in the right plot, the vertical axis x_3 is completely uncorrelated with x_1 and x_2, which causes the 2D subspace on which the data points lie 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 [93], 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.



(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.

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 $\mathbf{X}(:,j)$, 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:

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

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 $\|\mathbf{A}\,\mathrm{diag}(\mathbf{b})\|_{1,2} = \sum_{j=1}^{p} |\mathbf{b}(j)| \cdot \|\mathbf{A}(:, j)\|_2$. 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) [15] 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) [93] 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) [108], 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 $O(1/\sqrt{\epsilon})$ [6]. 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:

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

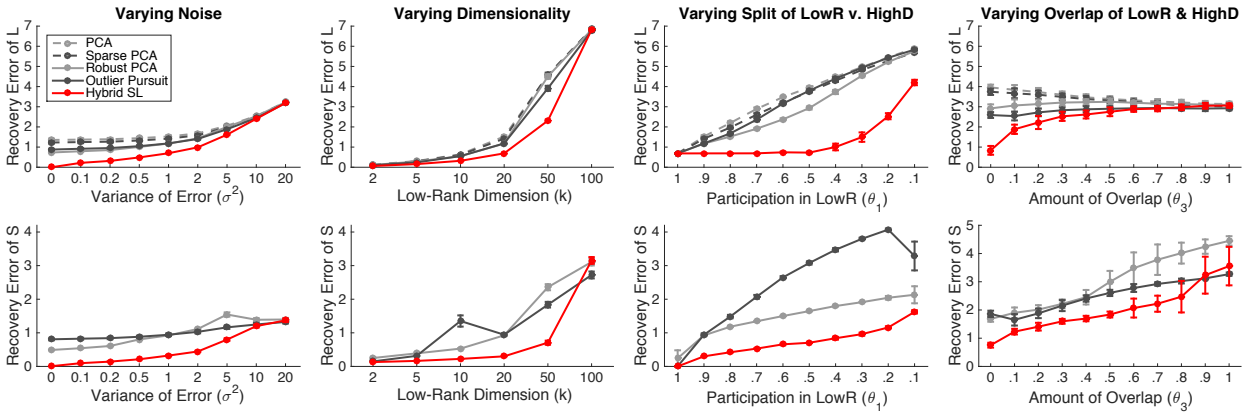


Figure 4.3: Results comparing the performance of our hybrid model against four baselines on synthetic data. The top row shows the recovery error for the low-rank component $\mathbf{L}$, and the bottom row shows the recovery error for the high-dimensional component $\mathbf{S}$. Results are averaged over 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 [108], Robust PCA [15], and Outlier Pursuit [93]. 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

[1] This is based on [63] who proposed using warm starts for a nonconvex sparse regularizer.

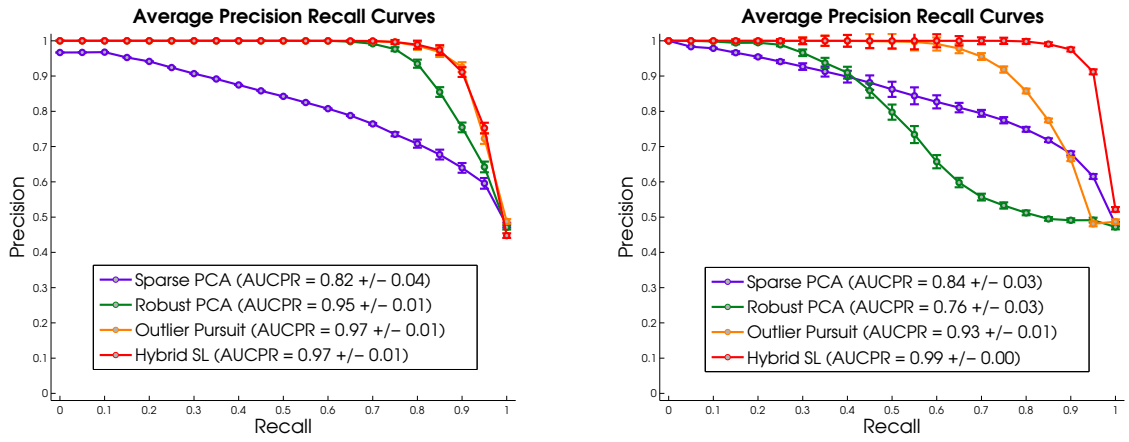


Figure 4.4: Average precision-recall curves for SPCA, RPCA, OP, and HSL calculated by varying hyperparameter values and evaluating recovery of the true set of high-dimensional features. Each curve is averaged over 20 simulated datasets.

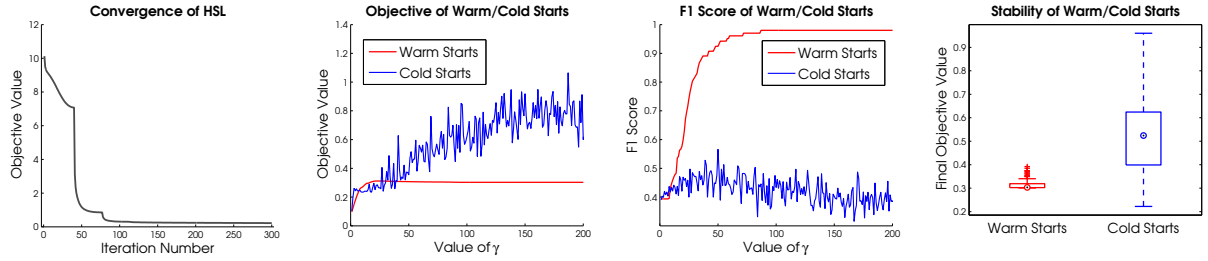


Figure 4.5: (a) Convergence of HSL. (b) The final objective value obtained from running HSL with each value of γ using warm and cold starts. (c) The F1 score on the selection of high-dimensional features obtained from running HSL with each value of γ using warm and cold starts. (d) The final objective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

$\mathbf{b}$ to zero according to the proportions specified in θ. Finally we generate the data according to $\mathbf{X} = \mathbf{ZA} + \mathbf{W}\,\mathrm{diag}(\mathbf{b}) + \mathbf{E}$, 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

[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

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 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 [15]. In keeping with our synthetic experiments, OP was run

the fact that the magnitude of the generated features depends on the value of k.

[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

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 [13].

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 [88], GBM [87], and colon [36] cancers and 5 clusters for kidney [73] and lung [90] 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 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

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

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

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 [28] of the features comprising the high-d component, and identified known oncogenes [7] 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 [65]. 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-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 [14]. 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

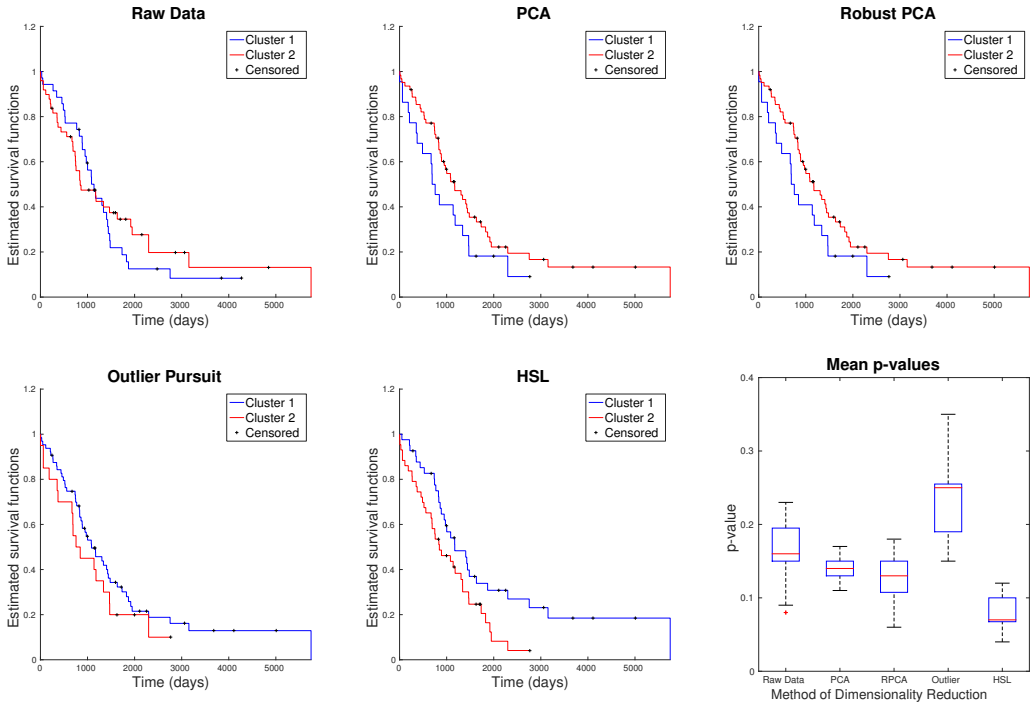


Figure 4.6: Results of survival analysis performed on a breast cancer gene expression dataset. (a-e) Examples of representative Kaplan-Meier survival function estimates. (f) Distribution of p-values over 100 clustering initializations.

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

Cancer Survival Analysis

5.1 Introduction

One of the central hypotheses of this thesis is that leveraging structure and sharing information across related learning tasks can help boost the signal-to-noise ratio when learning from high-dimensional biological data. In the final part of this thesis, we set out to explore and systematically test this hypothesis. To do this, we focus on a particular task that has significant clinical relevance: survival prediction for cancer patients. In particular, we seek to combine genomic and clinical data from patients with multiple different cancer types, and share information across the distinct but closely related tasks of predicting survival rates for each cancer type.

In addition to its clinical relevance, we choose this problem for two key reasons. First, cancer data is extremely heterogeneous. Even across patients with the same cancer type, molecular and genetic diversity abounds. This problem is only exacerbated by aggregating data from patients with multiple distinct types of cancer. Heterogeneity across samples means that the i.i.d. assumption that is central to many statistical learning models is violated. Second, survival data is particularly prone to small sample sizes due to censorship, which occurs when the survival outcome is not observed for patients that survive past the end of the study or stop responding to follow-up requests. In practice, although survival models are able to cope with censored data, they still primarily learn from uncensored samples where the outcome is observed. This further exacerbates the curse of dimensionality, which is already a problem due to the high-dimensional nature of genomic datasets, in which we typically have many more features (e.g. genes) than samples (e.g. human patients).

Using the pan-cancer dataset described in the next section, we systematically evaluate the effects of incorporating different degrees of information sharing between the survival models, ranging from estimating completely independent models for each cancer type all the way to estimating a single fully joint model that does not distinguish between cancer types. This concept is illustrated in Figure 5.1, where we show a hypothetical set of 5 survival prediction tasks, each of which involves predicting y_i from x_i. In the left-most diagram, a completely independent model is learned for each cancer type, with no information sharing. This method would work well if there were no similarities between the tasks. In the right-most diagram, a single unified model is learned for all cancer types, with no differentiation between them. This method would work well if there were no differences between the tasks. The middle diagram shows a hybrid approach in which we estimate multiple related models by sharing information between them.

Although this diagram only shows three distinct possibilities, in reality there are multiple dimensions over which in which information can be shared. In this chapter, we explore several different types of information sharing in order to determine whether any of them lead to improvements over the independent models that are traditionally used for cancer survival prediction [103].

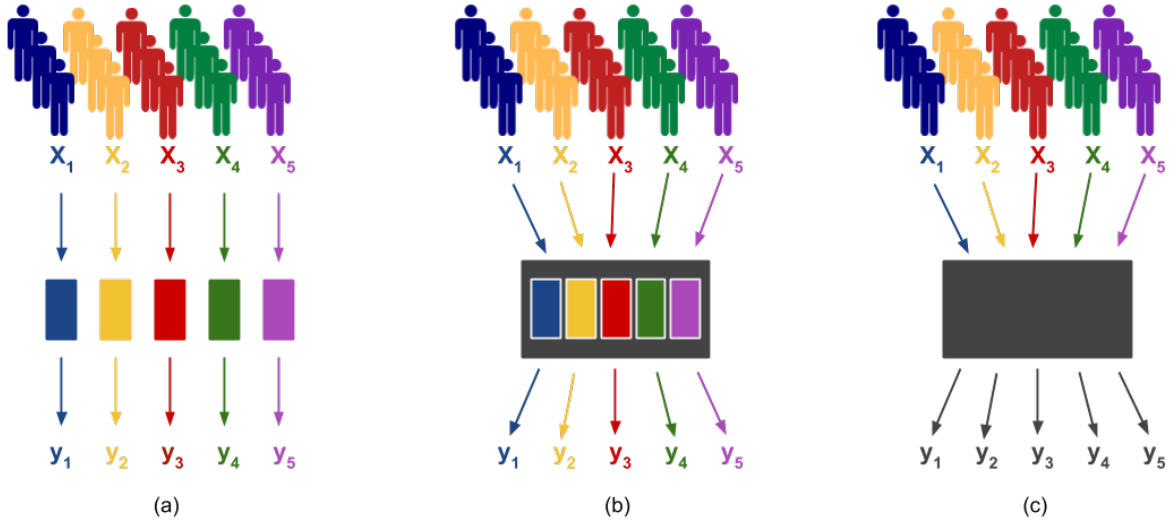


Figure 5.1: An illustration of multiple degrees of information sharing. The people represent cancer patients, with the colors indicating different cancer types. Each diagram shows a different approach for aggregating data across cancer types. (a) Independent models: estimate completely separate models for each task, and do not share any information between the tasks. (b) Related models: estimate multiple models that are related but still capture some differences between the tasks. (c) Single model: estimate a single model for all tasks, do not differentiate between the tasks at all.

5.2 Dataset

We conduct our analysis on a pan-cancer dataset collected from The Cancer Genome Atlas (TCGA). Our dataset spans 10 cancer types and contains a total of $n = 4{,}610$ patients. For each patient, we downloaded and pre-processed gene expression data, survival data, and some additional clinical variables. After filtering out missing data and retaining only genes that had measurements for all patients in our cohort, we ended up with expression values for $p = 60{,}483$ genes. The survival outcome for each patient includes their survival time (measured in days to death if observed or days to last followup) and censorship status (whether the outcome was observed or censored). We only kept patients for whom both survival outcomes and gene expression data was available. We also extracted the patient's age and gender from their clinical file so that we could incorporate these additional non-genomic covariates into our survival models. We did not use any additional clinical variables, such as tumor grade, because these tend to vary quite a bit across cancer types and have high rates of missing data.

For our experiments, we split our data into a training set (80%) and test set (20%). Table 5.1 shows a list of the cancer types in our dataset along with the training and test sample sizes for each cancer type. Overall, the vast majority (76%) of the patients have censored outcomes. Some cancer types have much higher censorship rates than others. In particular there are two cancer types, Prostate Adenocarcinoma (PRAD) and Thyroid Carcinoma (THCA), that respectively only have 7 and 11 uncensored samples in their training sets. Unfortunately this means that the test sets for these cancer types are even smaller and may cause high variance in the results that we obtain. However, the extreme disparity between the dimensionality of the problem ($p \approx 60{,}000$) and the uncensored sample size ($n \approx 10$) for these cancer types clearly underscores the need for leveraging information from additional samples in order to mitigate the curse of dimensionality.

Table 5.1: 10-Cancer TCGA Gene Expression Dataset

Cancer Type	Sample Sizes: Total (Uncensored)	
	Training Set	Test Set
Breast Invasive Carcinoma (BRCA)	760 (106)	191 (27)
Glioblastoma Multiforme (GBM)	117 (94)	29 (23)
Head and Neck Squamous Cell Carcinoma (HNSC)	387 (169)	96 (42)
Kidney Renal Clear Cell Carcinoma (KIRC)	263 (62)	65 (15)
Brain Lower Grade Glioma (LGG)	396 (98)	100 (25)
Lung Adenocarcinoma (LUAD)	379 (136)	95 (34)
Ovarian Serous Cystadenocarcinoma (OV)	203 (121)	51 (30)
Prostate Adenocarcinoma (PRAD)	382 (7)	96 (2)
Thyroid Carcinoma (THCA)	387 (11)	97 (3)
Uterine Corpus Endometrial Carcinoma (UCEC)	413 (69)	103 (17)
Total	3687 (873)	923 (218)

5.3 Related Work

Prior work in pan-cancer analysis for survival prediction has been somewhat limited, but has nonetheless shown significant promise. As part of a broader analysis of the utility of genomic data in predicting patient survival for 4 different cancer types, [103] 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, [99] 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.

Another recent large-scale study of cancer survival prediction analyzed a pan-cancer dataset containing tumors aggregated from 32 cancer types to obtain a cohort with 9,000 patients [94]. In this study, the authors fit a series of univariate Cox proportional hazards models on the aggregated pan-cancer dataset to identify a list of the top 10 adverse and top 10 favorable prognostic genes. They subsequently calculated a risk score for each patient using a weighted sum of the individual gene expression values weighted by their regression coefficients from the univariate models. This analysis was then repeated using only the data for each individual cancer type. Although the authors estimated both pan-cancer and individual cancer models, they did not evaluate whether the pan-cancer model led to better survival prediction. However, they briefly mentioned that the individual models are too limited by small sample sizes to be of real clinical value.

The approaches described above all focus 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 an unsupervised

manner by integrating datasets across cancer types, and then use the latent representation to train separate survival models for individual cancer types [19, 16]. One of these approaches led to the winning submission in the Sage Bionetworks DREAM Breast Cancer Prognosis Challenge [20].

In [2], the authors conduct a pan-cancer analysis of prognostic genes by first fitting a series of univariate Cox models to each gene on each individual cancer type and then using the results to perform an in-depth analysis of the similarities and differences across cancer types. One of the key conclusions from this study is that although cancer types do not have much overlap among the specific sets of genes with the smallest p-values, they do share co-regulated gene sets, which provides additional evidence to suggest that joint survival models can outperform individual models.

Finally, a few recent methods have been proposed to use deep learning for pan-cancer survival prediction, including [17], which trains a pan-cancer survival model using data aggregated from 20 cancer types and from multiple different modalities of genomic data. The authors compare their pan-cancer model to models trained on individual cancer types and observe a significant performance improvement for the majority of cancer types, with only one performance regression. However, their unified pan-cancer model does not attempt to explicitly capture the differences between the cancer types and therefore may miss some of the key patterns that set some cancers apart from others.

5.4 Methodology

In this section, we describe the methodology we used to conduct a comprehensive study of transfer learning for multi-task cancer survival prediction. Given a set of covariates $X_i = (X_{i1}, \cdots, X_{ip})$ and a (possibly censored) survival outcome value Y_i, we want to predict the relative survival across patients. To achieve this, we use the standard multivariate Cox proportional hazards regression model [23] with an additional ℓ_1 regularization term. We refer to this model as the Cox Lasso. The regression parameters $\beta \in \mathbb{R}^p$ capture the influence of each gene on patient hazard, which is inversely proportional to survival time. The regression parameters can be estimated by solving the following optimization problem, which minimizes the partial log likelihood:

$$\min_{\beta} \sum_{i:E_i=1} \left(\log \sum_{j:Y_j \geq Y_i} \exp\{X_j^T \beta\} - X_i^T \beta \right) + \lambda \|\beta\|_1 \tag{5.1}$$

Here E_i is an indicator variable whose value is 1 if the event was observed and 0 if it was censored.

In our setting, each patient i belongs to a particular cancer type $T_i = t \in \mathcal{T}$. We therefore estimate one set of regression parameters $\beta^{(t)}$ for each cancer type. As discussed in the following sections, we experiment with different degrees of information sharing between these $|\mathcal{T}|$ tasks in order to arrive at the best possible estimate of each $\beta^{(t)}$.

Table 5.2 provides a brief overview of the different types of sharing that we examine. Additional details about each type can be found in Sections 5.4.1 through 5.4.4. We assume that we have expression measurements for the exact same set of genes for all cancer types, and we also have clinical covariates (age and gender) for each patient, which we include as features in all models.

Across all experiments, we evaluate the predictive power of each fitted model using Harrell's concordance index [38], which is a measure of ranking accuracy. Given an estimated set of regression parameters $\{\hat{\beta}^{(t)}\}_{t \in \mathcal{T}}$, the predicted hazard ratio of an individual patient i with cancer type $T_i = t$ is given by $\hat{H}_i = \exp\{X_i^T \beta^{(t)}\}$. Since the hazard ratio is inversely proportional to survival time, we report $1 - \text{concordance}(\hat{H}, Y)$ where $\hat{H} = (\hat{H}_1, \cdots, \hat{H}_n)$ are the predicted hazards and $Y = (Y_1, \cdots, Y_n)$ are the ground truth survival times for a set of patients.

Table 5.2: Transfer Learning Study for Cancer Survival Prediction

Type of Information Sharing	Experiment Purpose
Feature Selection	Assess the effect of performing separate vs. joint feature selection across cancer types.
Hyperparameter Tuning (λ)	Assess the effect of performing separate vs. joint hyperparameter tuning (to control the sparsity level) across cancer types.
Regression Parameters (β)	Assess the effect of encouraging or enforcing different degrees of similarity between the estimated regression parameters for each cancer type.
Objective Function	Assess the effect of minimizing the sum of separate Cox objectives for each cancer type vs. a single joint Cox objective across all patients.

5.4.1 Feature Selection

Before fitting any multivariate survival models, as is standard in the literature, we perform a preliminary feature selection step in order to select a relevant and compact set of genes that are predictive of the survival outcome. To select the genes, we fit a univariate Cox model to each gene in turn and calculate a p-value for each gene using a likelihood ratio test. We then select the m genes with the smallest p-values, for several different values of m. These genes may be either positively or negatively associated with survival.

In order to evaluate the effect of sharing information between cancer types during the feature selection process, we compare two different approaches:

- Separate Feature Selection: In this setting, feature selection is performed separately for each cancer type. This means that each $\beta^{(t)}$ may be estimated from a different set of m genes.
- Joint Feature Selection: In this setting, feature selection is performed jointly across all cancer types. This means that each $\beta^{(t)}$ will be estimated from the same set of m genes, although each $\beta^{(t)}$ is still estimated separately, i.e., we fit $|\mathcal{T}|$ independent Cox Lasso models.

When performing joint feature selection, instead of using a univariate model to estimate the p-value for each gene, we fit an 11-variable Cox model that includes 10 binary variables representing a one-hot encoding of the patient's cancer type. This allows us to identify genes that still provide prognostic information after accounting for the information encoded by the cancer type itself.

5.4.2 Hyperparameter Tuning

In order to determine the value of the regularization hyperparameter λ from Equation 5.1, we follow the standard approach of performing a grid search over a range of possible values of λ and evaluating each fitted model on a held-out validation set, then choosing the value of λ that leads to the highest concordance index on the validation set. In order to evaluate the effect of sharing information during the hyperparameter tuning phase, we compare two different approaches:

- Separate Hyperparameter Tuning: In this setting, λ selection is performed separately for each cancer type. As a result, we are free to choose a different value of λ for each cancer type, leading to differing amounts of regularization in the estimate of each $\beta^{(t)}$.

- Joint Hyperparameter Tuning: In this setting, λ selection is performed jointly across all cancer types. As a result, we are forced to choose the same value of λ for all cancer types.

We conduct this experiment using the *Joint Feature Selection* setting from Section 5.4.1, since it does not necessarily make sense to use the same amount of regularization across cancer types when the set of features are different between them.

5.4.3 Regression Parameters

Next, we experiment with directly sharing information across the regression parameter values $\beta^{(t)}$ for $t \in \mathcal{T}$. We compare three different sharing settings:

- No Sharing: In this setting, we estimate the regression parameter values independently for each cancer type. Specifically, we use the model shown in Equation 5.2, which fully decomposes over cancer types.
- GFLasso Penalty: In this setting, we apply a graph-guided fused lasso penalty on the matrix $B \in \mathbb{R}^{m \times |\mathcal{T}|}$ formed by performing a column-wise concatenation of each $\beta^{(t)}$ vector. Specifically, we use the model shown in Equation 5.3. Because we do not have good prior knowledge about the relationships between cancer types, we set $w_{t_i t_j} = 1$ for all pairs of cancer types.
- Full Sharing: In this setting, we require that the regression parameter values be identical across cancer types, i.e. we enforce $\beta^{(t_i)} = \beta^{(t_j)} \ \forall \ t_i, t_j \in \mathcal{T}$. Specifically, we use the model shown in Equation 5.4, where the shared regression parameters are simply denoted by β.

The specific penalty function formulations we consider are given below.

$$\min_{\{\beta^{(t)}\}_{t \in \mathcal{T}}} \ \ell(B) + \lambda \sum_{t \in \mathcal{T}} \|\beta^{(t)}\|_1 \tag{5.2}$$

$$\min_{\{\beta^{(t)}\}_{t \in \mathcal{T}}} \ \ell(B) + \lambda \sum_{t \in \mathcal{T}} \|\beta^{(t)}\|_1 + \gamma \sum_{t_i \in \mathcal{T}} \sum_{t_j \in \mathcal{T}} w_{t_i t_j} \|\beta^{(t_i)} - \beta^{(t_j)}\|_1 \tag{5.3}$$

$$\min_{\beta} \ \ell(B) + \lambda \sum_{t \in \mathcal{T}} \|\beta\|_1 \tag{5.4}$$

Here B is used as shorthand for $\{\beta^{(t)}\}_{t \in \mathcal{T}}$ and $\ell(B)$ denotes the multi-task Cox loss function, which minimizes the partial log likelihood as in Equation 5.1 but considers all cancer types. Details about how the loss function is formulated are provided in the next section. We conduct this experiment using the *Joint Feature Selection* setting and the *Joint Hyperparameter Tuning* setting from Sections 5.4.1 and 5.4.2, respectively.

5.4.4 Objective Function

In our final experiment, we evaluate the effect of including all samples from all cancer types in the Cox loss function used to estimate the regression parameters for each individual cancer type. We hypothesize that considering all pairs of patients for every cancer type in the loss function itself leads to a richer prediction target, which may help the model estimate more robust associations. In particular, we compare two approaches:

- Separate Objective Functions: In this setting, we estimate each $\beta^{(t)}$ using only the samples from cancer type t. Specifically, we optimize the objective shown in Equation 5.5.

- Joint Objective Functions: In this setting, we estimate each $\beta^{(t)}$ using some information from all of the cancer types. Specifically, we optimize the objective shown in Equation 5.6. This is different from Equation 5.5 in that the second sum is over all j such that $Y_j \geq Y_i$, even if patient j belongs to a different cancer type than patient i. This effectively leads to an estimate of $\beta^{(t)}$ that is able to accurately rank patient i's survival against that of all other patients from all cancer types, not just those with the same cancer type as i.

The two objective function formulations we consider are given below. Both are slight variations of the general formulation given in Equation 5.1. Here B is used as shorthand for $\{\beta^{(t)}\}_{t \in \mathcal{T}}$ and $\psi(B)$ denotes a sparsity-inducing or structure-inducing penalty term over B. Refer to the previous section for details about how the penalty term is formulated.

$$\min_{\{\beta^{(t)}\}_{t \in \mathcal{T}}} \sum_{t \in \mathcal{T}} \sum_{i:E_i=1,T_i=t} \left(\log \sum_{j:Y_j \geq Y_i, T_j=t} \exp\{X_j^T \beta^{(t)}\} - X_i^T \beta^{(t)} \right) + \psi(B) \tag{5.5}$$

$$\min_{\{\beta^{(t)}\}_{t \in \mathcal{T}}} \sum_{t \in \mathcal{T}} \sum_{i:E_i=1,T_i=t} \left(\log \sum_{j:Y_j \geq Y_i} \exp\{X_j^T \beta^{(T_j)}\} - X_i^T \beta^{(t)} \right) + \psi(B) \tag{5.6}$$

$$\equiv \min_{\{\beta^{(t)}\}_{t \in \mathcal{T}}} \sum_{i:E_i=1} \left(\log \sum_{j:Y_j \geq Y_i} \exp\{X_j^T \beta^{(T_j)}\} - X_i^T \beta^{(T_i)} \right) + \psi(B)$$

When using the joint objective functions, we also include the cancer type itself as an additional covariate in the model. We do this by converting it into a set of 10 binary feature values using a one-hot encoding.

We conduct this experiment using the *Joint Feature Selection* setting and the *Joint Hyperparameter Tuning* setting from Sections 5.4.1 and 5.4.2, respectively. We experiment with using both objective function formulations with each of the parameter sharing settings described in Section 5.4.3, which we can easily do by substituting either the separate or joint loss for $\ell(B)$ in Equations 5.2 through 5.4.

5.5 Quantitative Results

5.5.1 Experimental Details

In order to determine the statistical significance of our results, we conducted our experiments by generating 25 random splits of the data into a training set and test set. In each split, the training set contains 80% of the data and the test set contains the remaining 20%. When splitting the data, we made sure to preserve both the distribution of cancer types and the ratio of censored to uncensored samples within each cancer type. We train all models on the training set and report performance on the test set, with error bars showing the standard error over the 25 splits.

In order to perform hyperparameter selection for each model, we further split the training set into a training and validation set, again respectively using 80% and 20% of the data. We trained models with each hyperparameter setting on the training set and then selected the hyperparameter values that led to the best performance on the held-out validation set. We then retrained a final model on the full training plus validation set using the chosen hyperparameter values.

We repeated our experiments using a wide range of different values of m, where m denotes the number of genes selected in the initial feature selection step. In particular, we experimented with $m \in \{10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$. This allows us to see how the results change

as we move from the low-dimensional setting, where $p < n$, to the high-dimensional setting, where $p > n$. In our dataset, the total number of uncensored samples in the training set across all cancer types is $n = 873$. We therefore consider $p = 1{,}000$ to be the smallest gene set size that falls into the high-dimensional setting when any sharing is used.

To measure performance on the validation and test sets, we used the following sample-weighted concordance index:

$$\text{weighted c-index} = \sum_{t \in \mathcal{T}} \frac{n_u^t}{n_u} \text{concordance}(Y^{(t)}, \hat{Y}^{(t)}) \tag{5.7}$$

Here n_u is the total uncensored sample size across all cancers and n_u^t is the uncensored sample size for cancer type t. $Y^{(t)}$ and $\hat{Y}^{(t)}$ denote the ground truth and predicted survival outcomes for the set of patients with cancer type t. This weighted concordance was used in order to assign equal weight to each uncensored sample in the full dataset rather than assigning equal weight to each cancer type, which matches the design of our overall objective function.

The remainder of this section presents the quantitative results from each of the experiments described in Section 5.4 and discusses their implications.

5.5.2 Sharing Feature Selection

In this experiment, our aim was to understand the effect of using shared vs. joint feature selection as a pre-processing step before fitting individual Cox models.

The main results of this experiment are shown in Figure 5.2. They indicate that shared feature selection helps for some cancer types but not others. Furthermore, for each cancer type, sharing appears to help in some data dimensionality regimes but not others. One cancer type of interest is

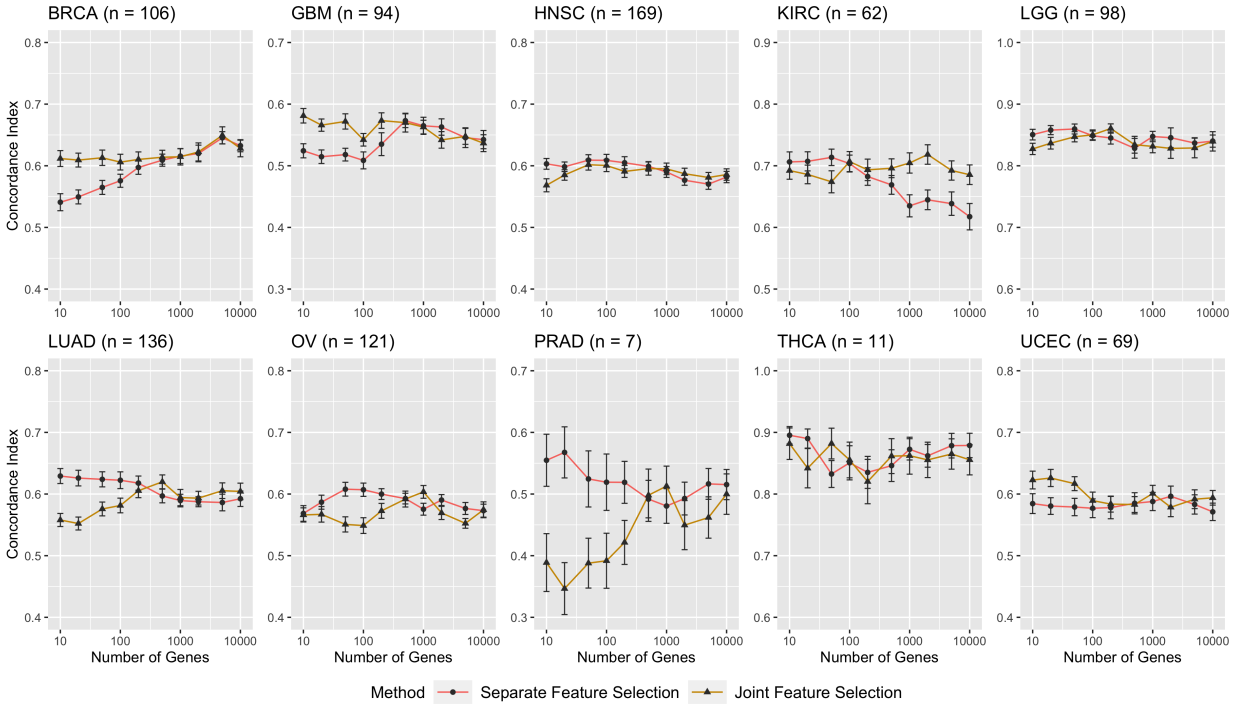


Figure 5.2: Comparison of separate vs. joint feature selection.

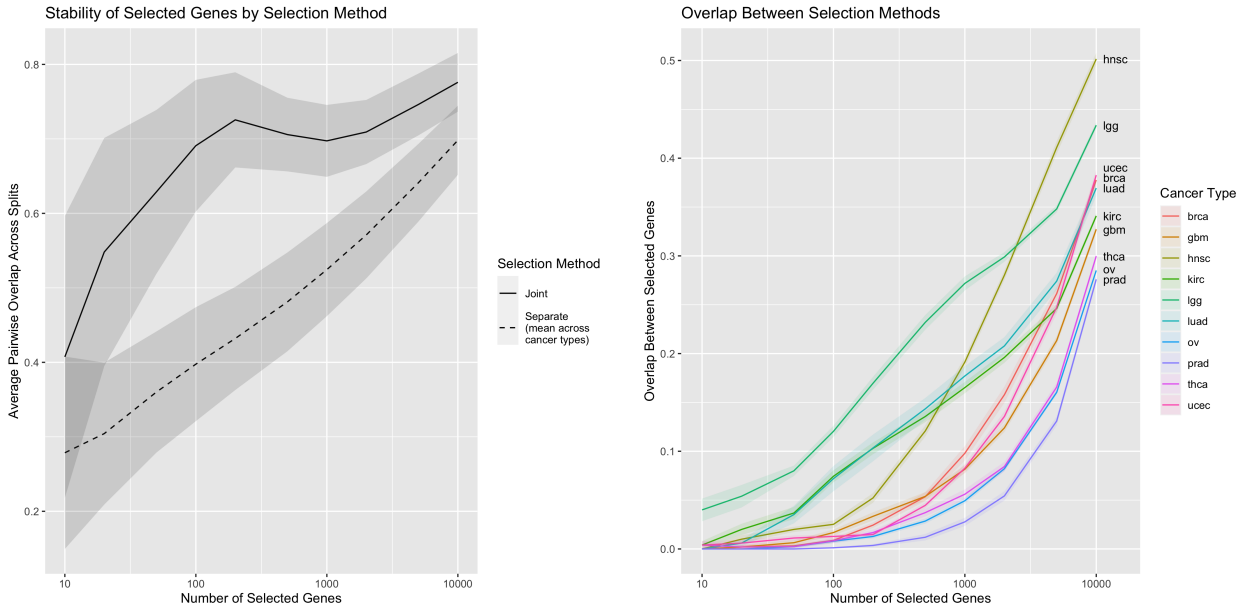


Figure 5.3: Left: The relative stability of separate and joint feature selection across random dataset splits. Right: The average overlap between features selected using the separate and joint methods.

BRCA, where sharing helps in the low-dimensional regime but the performance of separate feature selection catches up in the high-dimensional regime. Other cancer types, such as GBM and to a lesser extent UCEC, follow a similar pattern. In contrast, we see the opposite pattern in KIRC and some others, where sharing does not help in the low-dimensional regime but then starts to outperform no sharing in the high-dimensional regime.

The left panel of Figure 5.3 compares the stability of the set of genes selected using the separate and joint methods, measured as the pairwise overlap among the gene sets selected in each random split. Unsurprisingly, joint feature selection, which uses the full dataset and therefore estimates p-values over a larger set of samples, is much more stable than separate feature selection. This provides evidence to support the hypothesis that sharing information in the feature selection step boosts the signal-to-noise ratio. The right panel of Figure 5.3 shows the average overlap between the genes selected with the joint method and the separate method for each individual cancer type. Overall, the overlap is very low for small numbers of genes and never exceeds 50% even when selecting 10,000 out of the full set of 60,000 genes.

The stability and overlap results help explain the patterns observed in overall results from Figure 5.2. We hypothesize that joint feature selection leads to a more useful set of genes being selected for cancers such as BRCA that are more multi-genic and whose survival outcomes are less driven by a small set of high-impact genes. In contrast, there are several cancers where the joint feature selection method tends to miss some important genes, particularly when m is small. An interesting example is LUAD. If we examine the separate gene sets selected for LUAD, we see that there are a few key genes that are selected with high stability even when $m = 10$, the primary one being PITX, which is highly associated with LUAD survival. This gene is not selected in the joint set for small values of m.

Overall, the joint and separate methods tend to converge for larger values of m where there is generally more overlap among the selected gene sets. The notable exception to this pattern is KIRC, which benefits greatly from joint feature selection in the high-dimensional setting.

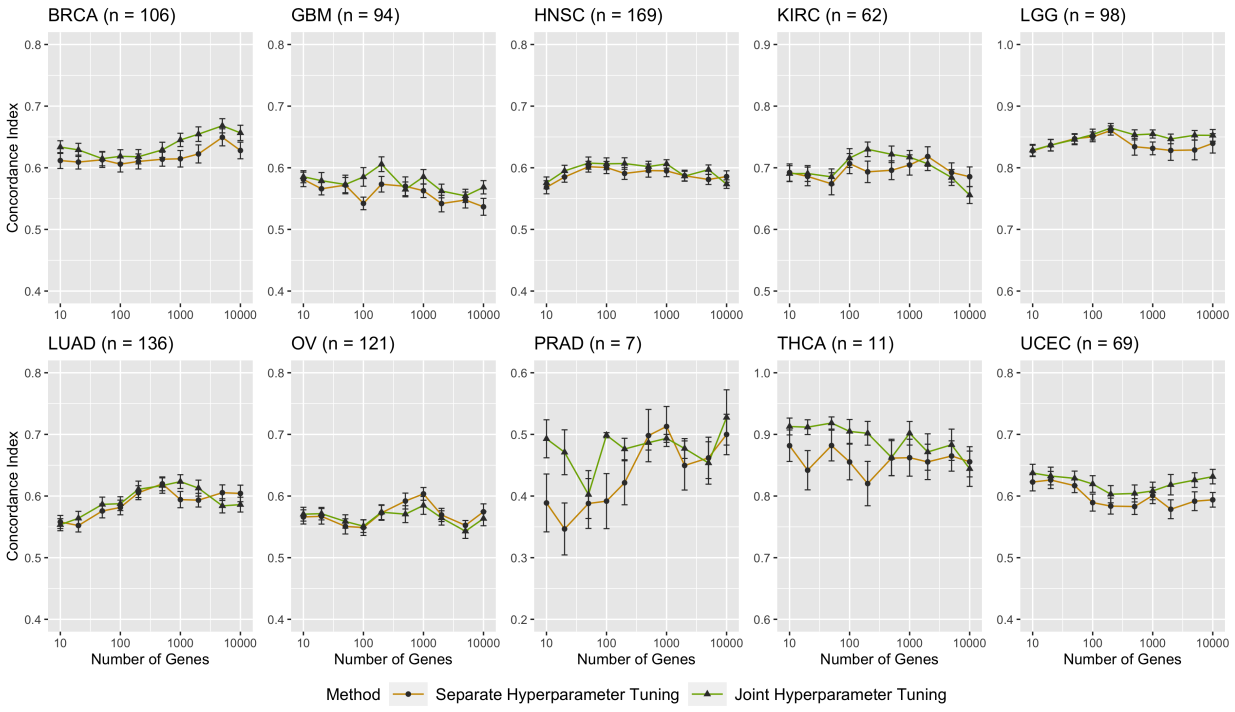


Figure 5.4: Comparison of separate vs. joint regularization hyperparameter tuning.

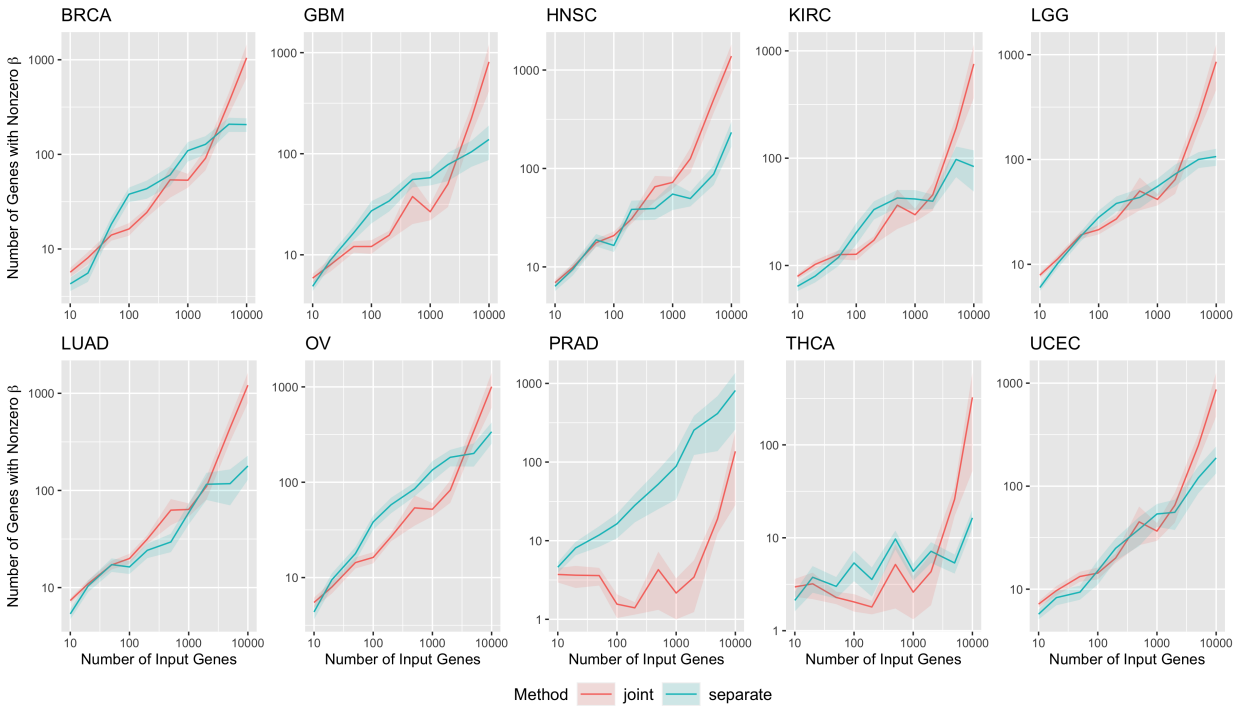


Figure 5.5: The counts of selected genes (i.e. those with nonzero β_j) in the regression parameter estimates when using separate vs. joint regularization hyperparameter tuning.

5.5.3 Sharing Hyperparameters

In this experiment, our goal was to understand the effect of using shared vs. joint hyperparameter selection in order to determine how much sparsity is appropriate for each regression parameter estimate $\beta^{(t)}$.

The results of this experiment are shown in Figure 5.4. Overall, we see that learning the value of λ jointly leads to a slight performance improvement in the majority of cancer types. We further investigate this by examining the overall sparsity of the estimated regression parameters when the hyperparameter is tuned separately vs. jointly. These results are shown in Figure 5.5, which plots the total number of genes with nonzero values in β as a function of the total number of input genes. Somewhat surprisingly, joint hyperparameter tuning leads to fewer genes being selected in the model for small numbers of input genes, but nearly always leads to more genes being selected in the high-dimensional setting.

A particularly interesting case is that of PRAD, for which we only have 7 uncensored samples in the training set. We see that joint hyperparameter selection leads to a much sparser model for PRAD than separate hyperparameter selection. This is likely because the separate approach leads to significant overfitting on this cancer type, whereas the joint approach is able to leverage information from the other cancer types to learn that more regularization is necessary in this case.

5.5.4 Sharing Regression Parameters

In this experiment, our goal was to evaluate the effects of directly sharing information between the hyperparameters for each cancer type, $\beta^{(t)}$ for $t \in \mathcal{T}$. The overall results of this experiment are shown in Figure 5.6. From here onward, we refer to the setting with no information sharing as

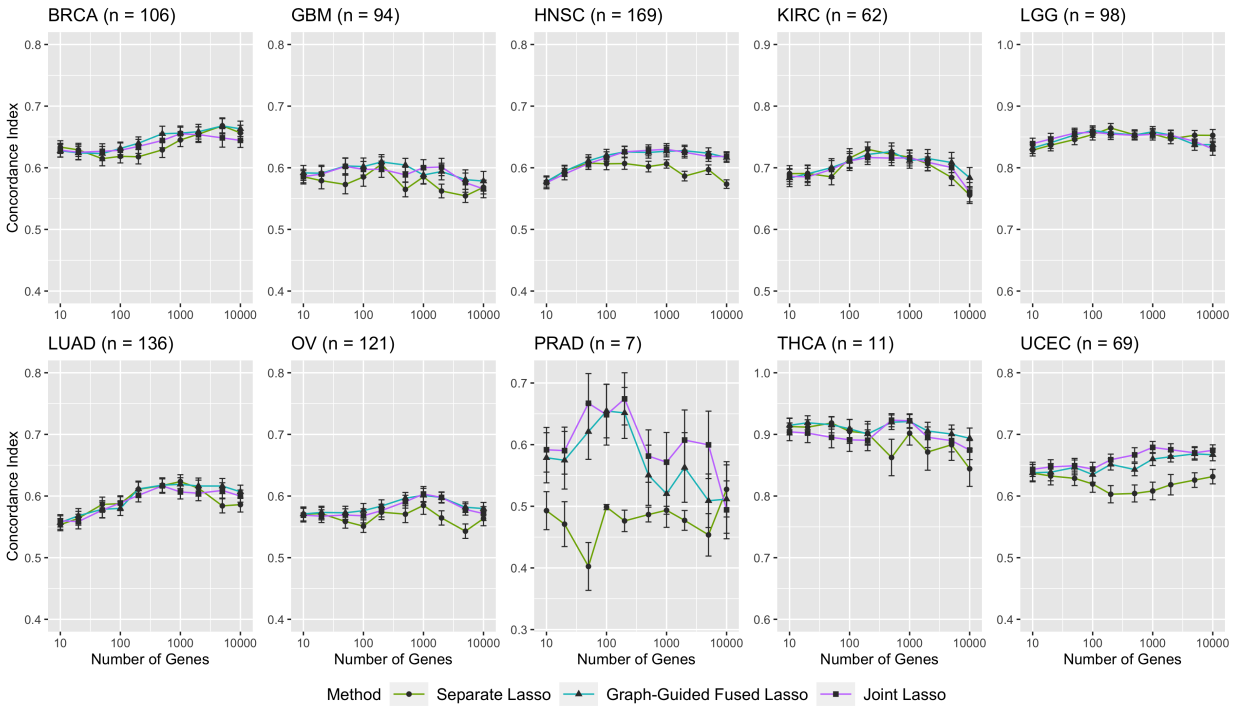


Figure 5.6: Comparison of different amounts of sharing among the regression parameters with the separate objective function formulation.

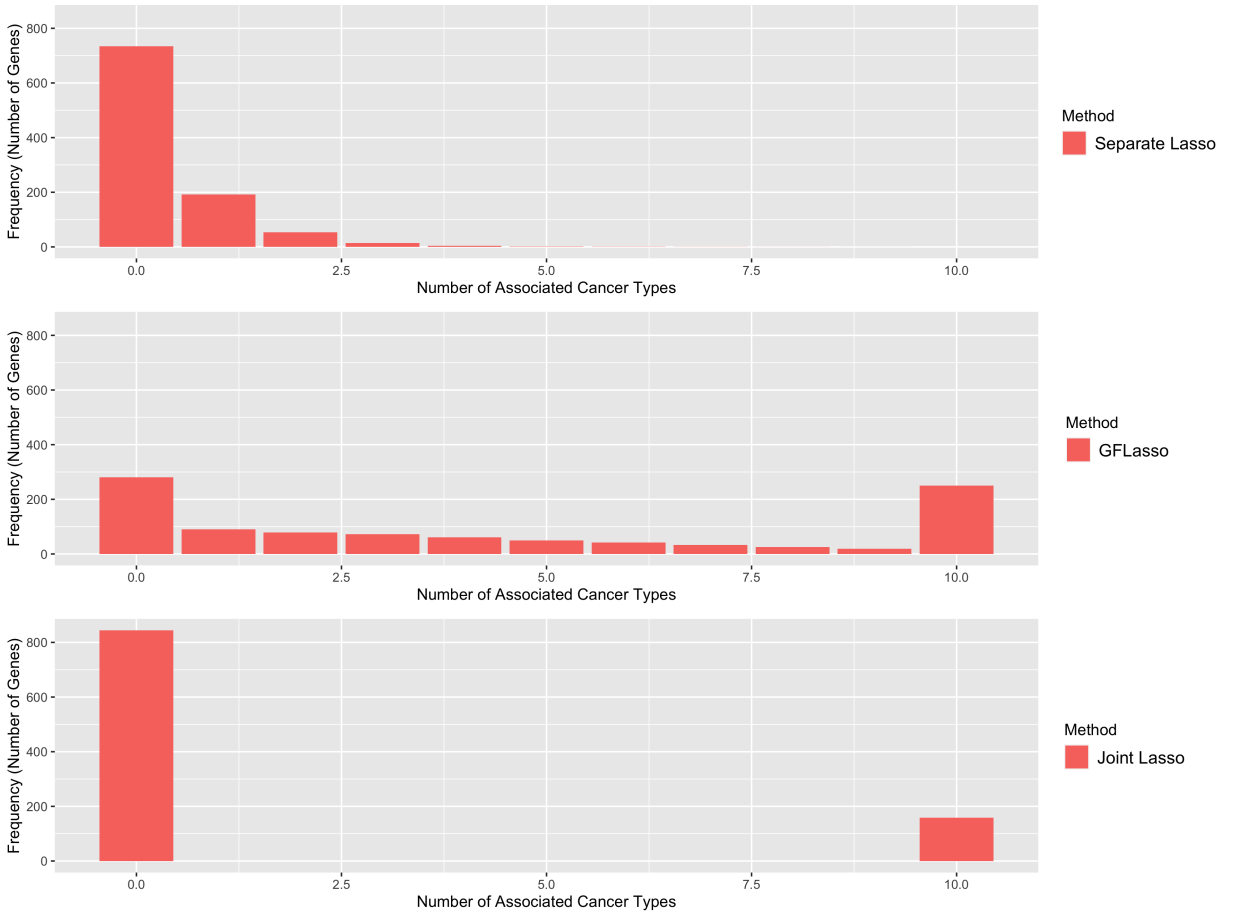


Figure 5.7: Distribution of the number of cancer types that each gene is associated with in the B estimates from Separate Lasso, GFLasso, and Joint Lasso with $p = 1000$ genes. Note that due to its design, Joint Lasso can only associate genes with either 0 cancer types or all 10 cancer types. In contrast, neither Separate Lasso nor GFLasso have any strict limitations on what they can express.

Separate Lasso (or SLasso for short), the setting with partial sharing using a graph-guided fused lasso penalty as GFLasso, the setting where we enforce exact equality of the regression parameters as Joint Lasso (or JLasso for short). The results shows that either GFLasso or Joint Lasso or both outperform Separate Lasso in the majority of cancer types, including BRCA, GBM, HNSC, OV, PRAD, THCA, and UCEC. Furthermore, in the three remaining cancer types, KIRC, LGG, and LUAD, all three methods perform comparably. Overall, this suggests that sharing information across regression parameters is a net win and never makes the performance worse.

In several cancer types, including HNSC and UCEC, we see that the performance gap between Separate Lasso and the other two methods widens as the number of genes included in the model increases. This provides evidence to support the hypothesis that sharing information across related tasks helps combat the curse of dimensionality.

To analyze these results in more detail, we also evaluated the sparsity pattern of the regression parameters that were estimated with each approach. Given an estimate of the full set of regression parameters B, each gene may have an association with one or more cancer types, where an association between gene j and cancer type t is indicated by a nonzero value of $\beta_j^{(t)}$. In order to gain

insight into the structure of B, we calculated the number of cancer types with which each gene was associated, given by $C_j = \sum_{t \in \mathcal{T}} \mathbb{I}\{\|\beta_j^{(t)}\| > 0\}$. We then plotted the distribution of C_j across genes for each method in Figure 5.7.

From these results, we can see that Separate Lasso primarily associates genes with only one or two cancer types and does not associate any genes with more than 7 cancer types. In contrast, by its very design, Joint Lasso can only associate genes with either no cancer types (in which case they are not selected in the model at all) or with all cancer types. However, interestingly, GFLasso yields a much more uniform distribution of association counts. This suggests that GFLasso is able to learn a much richer structure of gene associations with cancer types. In particular, it is able to identify genes that affect the multiple cancer types (including all 10 cancer types) but also identify genes that only affect a single cancer type. Based on its inherent limitations, Joint Lasso is only able to do the former. Based on empirical observations, Separate Lasso is only able to do the latter. We explore this hypothesis in more detail in Section 5.6.

5.5.5 Sharing the Objective Function

Finally, the aim of our last experiment was to investigate the effects of providing a richer and more challenging loss function for Cox regression by pooling information from all cancer types in the loss itself. This experiment was originally motivated by the observation that a fully joint model in which all data is pooled together and a single Cox model is trained also uses a joint loss function by default. We wanted to tease apart whether any performance improvements provided by a fully joint model were due to sharing information between the regression parameter estimates or due to

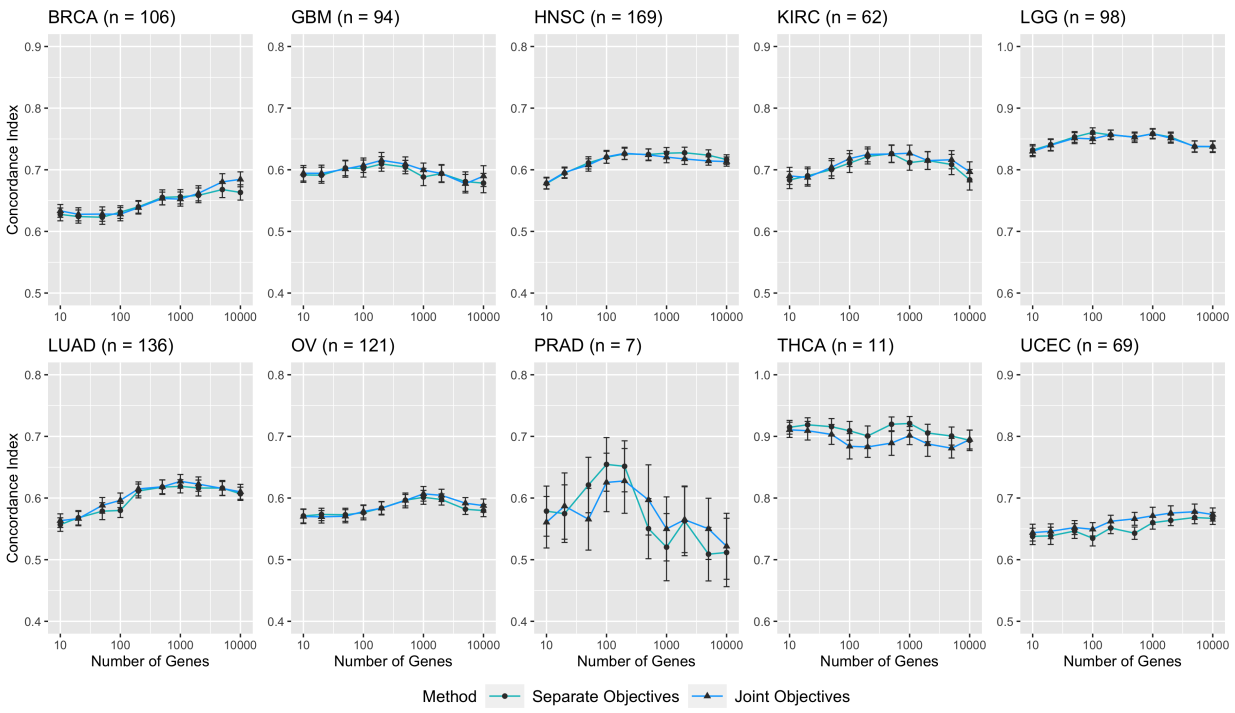


Figure 5.8: Comparison of separate vs. joint objective function formulations with GFLasso. We repeated the same experiment using both the Separate Lasso and Joint Lasso formulations and observed very similar results.

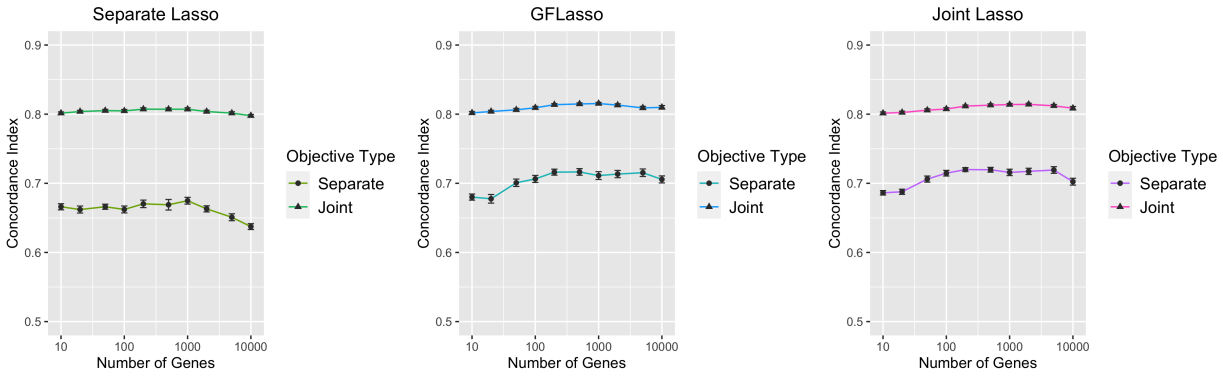


Figure 5.9: Comparison of separate vs. joint objective function formulations with Separate Lasso (left), GFLasso (middle), and Joint Lasso (right). The models are evaluated on the overall concordance index across all cancer types. Unsurprisingly, the joint objective far outperform the separate objective models because the separate models are not trained to perform well on this overall survival prediction task.

the loss function formulation.

The results from this experiment when using the GFLasso penalty are shown in Figure 5.8. Overall, we see that using the joint objective formulation does not help improve performance on the within-cancer survival prediction task. However, the results in Figure 5.9 show the overall test set concordance index measured on the full set of samples from all cancer types. Unsurprisingly, the models trained with a joint objective function significantly outperform the models trained with separate objective functions on this task.

We hypothesize that because the joint objective function provides a richer prediction target for the survival models, these models would also learn a more meaningful set of gene associations than models trained on the separate objectives. However, in order to restrict the scope of this analysis, we do not explore that direction any further in this thesis, and instead leave it for future work.

5.5.6 Analysis of Training vs Test Error

For completeness, we include the full set of experimental results, with concordance indices reported on both the training and test sets, in Figures 5.10 through 5.17. By comparing the training and test errors, we can clearly see that sharing information across related tasks leads to less overfitting on the training set, which generally translates to better performance on the test set.

One interesting observation from these results is that reducing the gap between the training and test error does not always correspond to improving the test error. For example, in nearly all settings, increasing the number of genes selected in the feature selection step leads to a much better fit to the training set. However, the effect on the test set is very mixed. In some cases, such as BRCA, including more genes seems to help its test set performance overall. In other cases, such as LUAD, the overall best performance is achieved with a small number of genes.

In general, the large gap between training and test error that is seen even with the full joint model indicates that there are still a lot of spurious patterns being estimated by the Cox model that do not generalize to unseen data. This suggests that more sophisticated or more structured approaches are needed in order to further improve the signal-to-noise ratio.

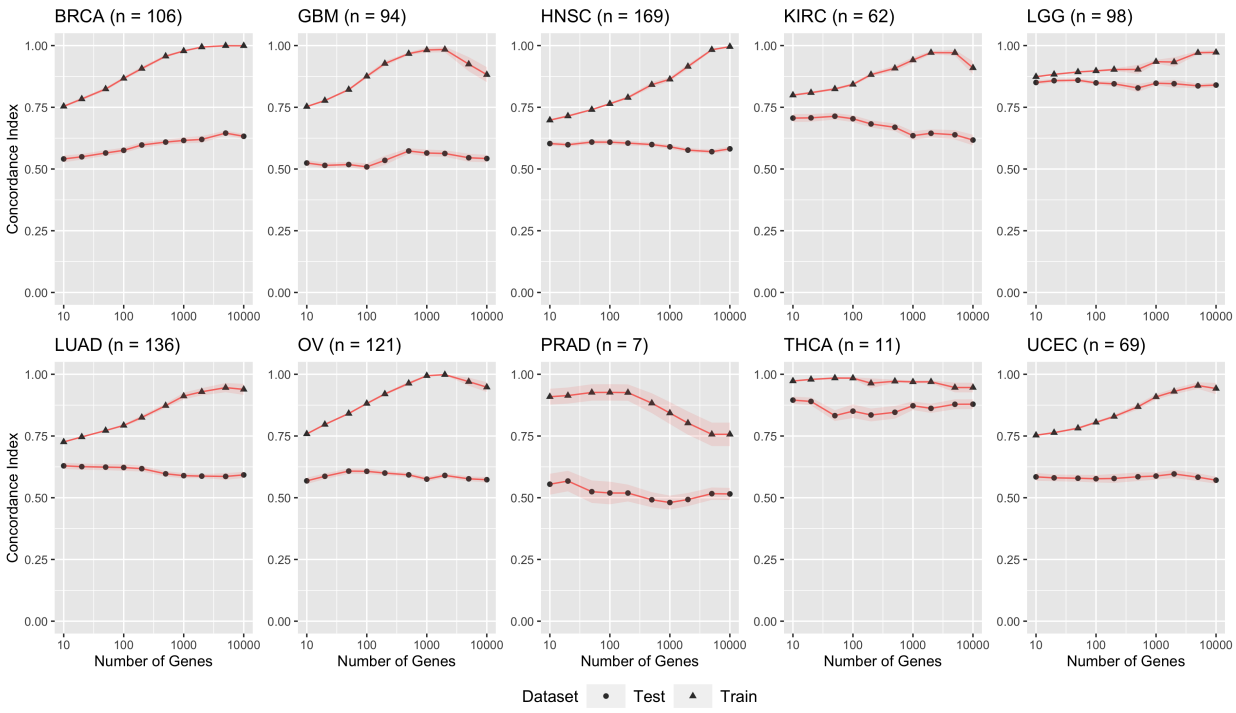


Figure 5.10: Results for Cox Lasso regression with separate feature selection, separate hyperparameter tuning, no regression parameter sharing, and separate objective functions.

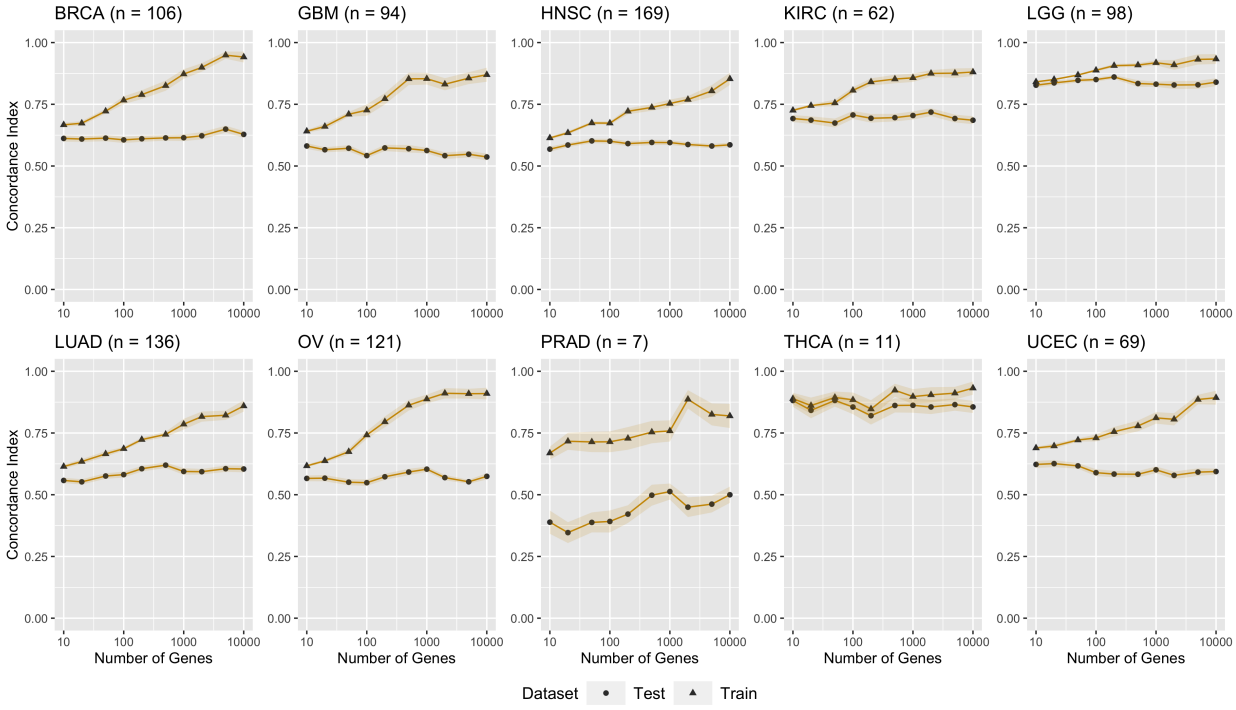


Figure 5.11: Results for Cox Lasso regression with joint feature selection, separate hyperparameter tuning, no regression parameter sharing, and separate objective functions.

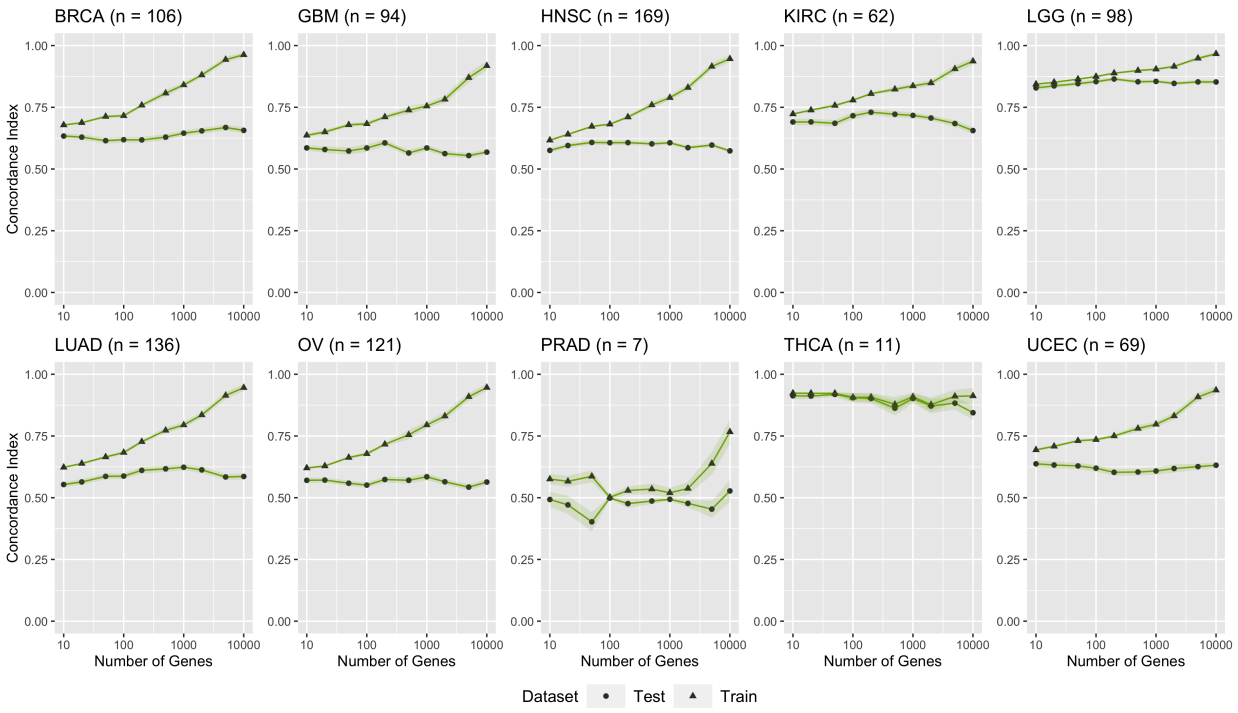


Figure 5.12: Results for Cox Lasso regression with joint feature selection, joint hyperparameter tuning, no regression parameter sharing, and separate objective functions.

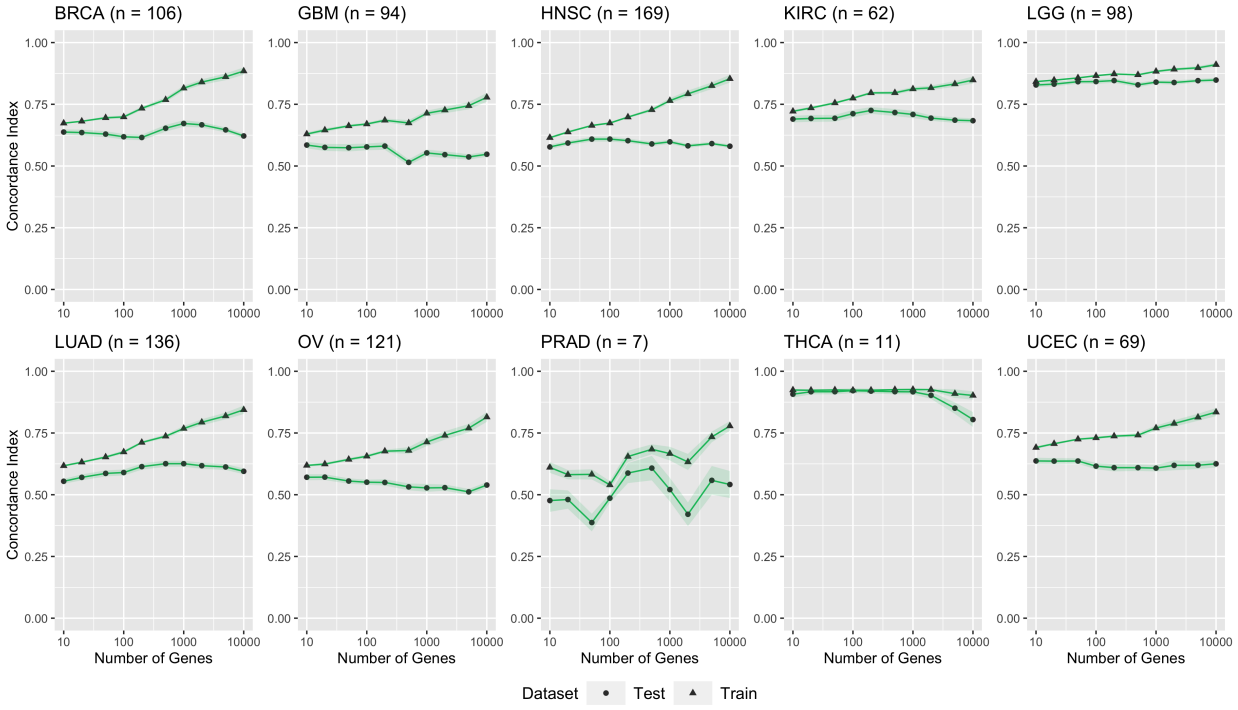


Figure 5.13: Results for Cox Lasso regression with joint feature selection, joint hyperparameter tuning, no regression parameter sharing, and joint objective functions.

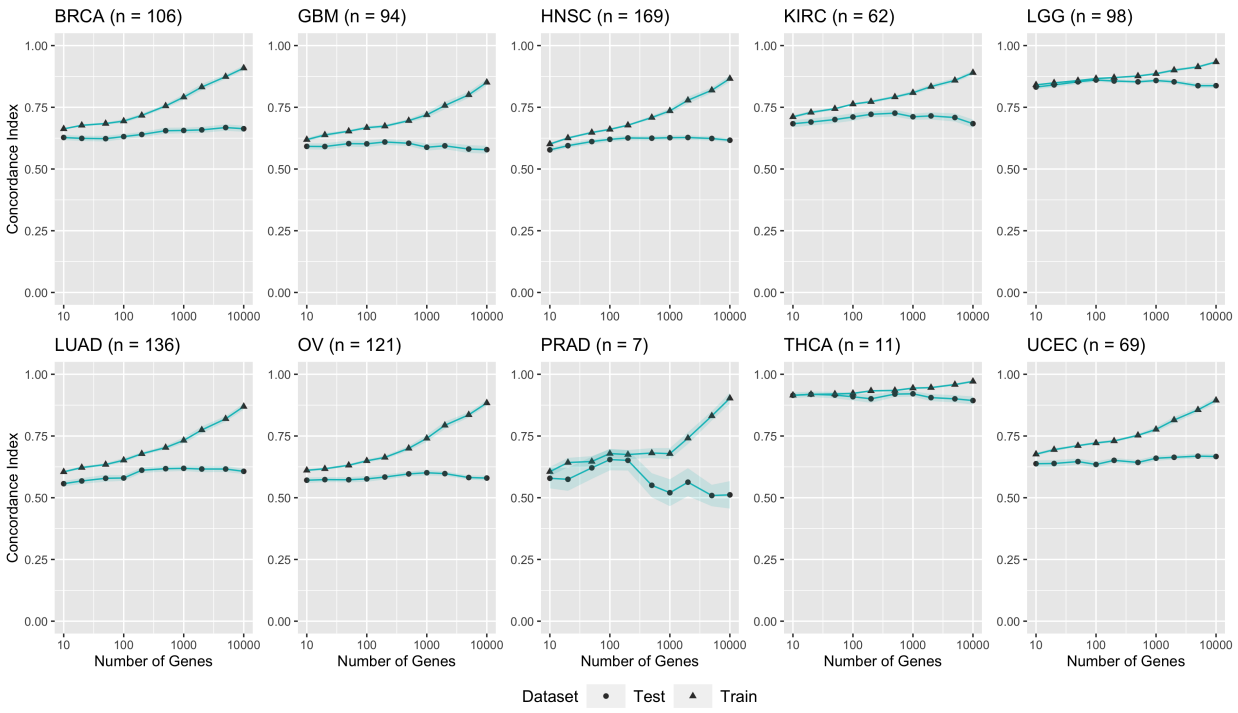


Figure 5.14: Results for Cox Lasso regression with joint feature selection, joint hyperparameter tuning, GFLasso regression parameter sharing, and separate objective functions.

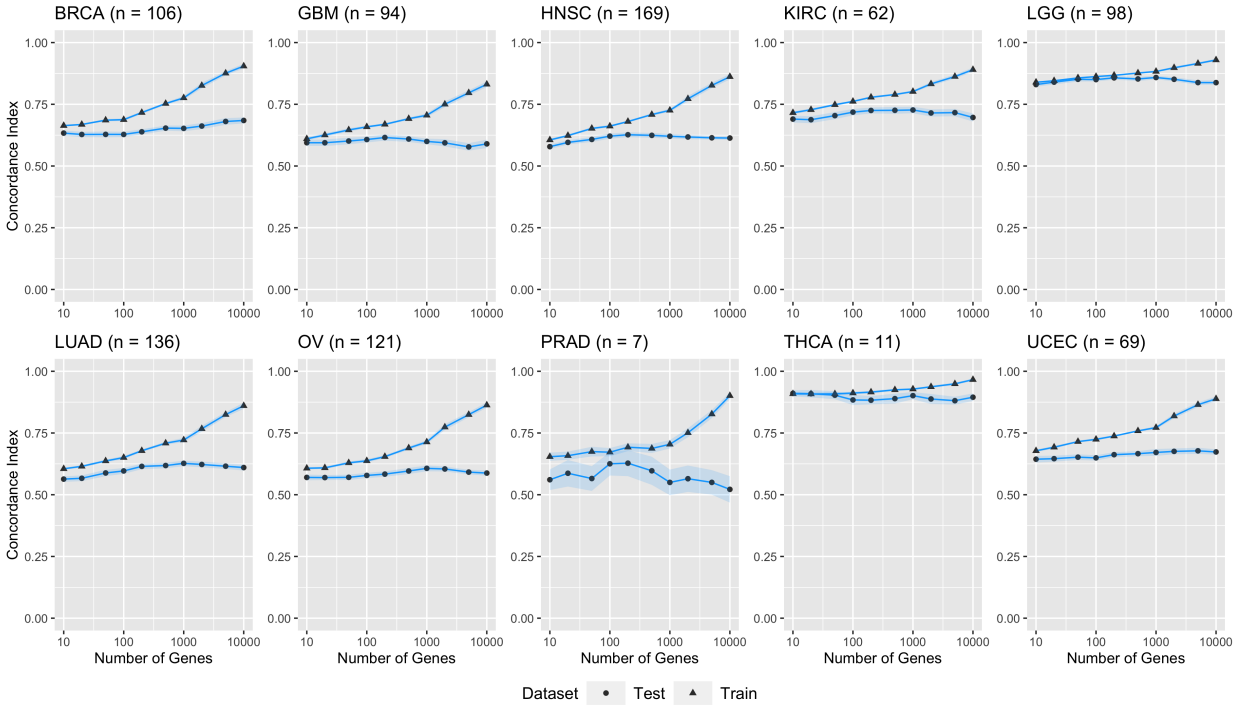


Figure 5.15: Results for Cox Lasso regression with joint feature selection, joint hyperparameter tuning, GFLasso regression parameter sharing, and joint objective functions.

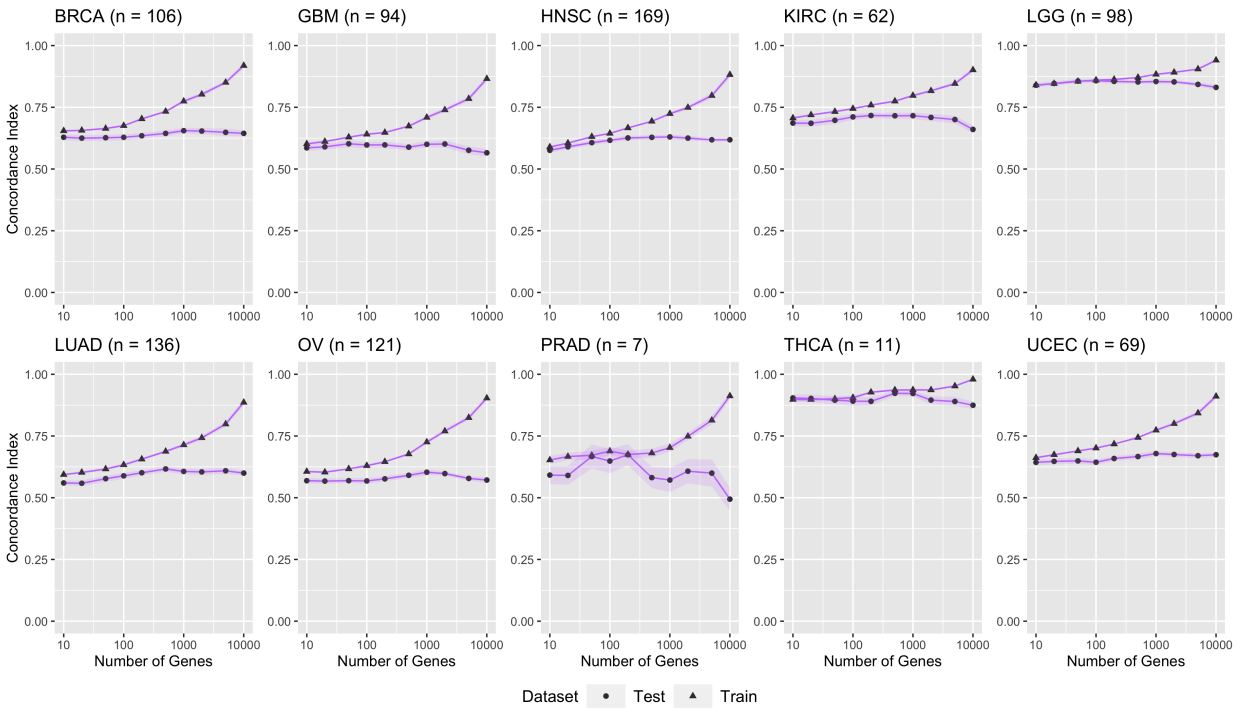


Figure 5.16: Results for Cox Lasso regression with joint feature selection, joint hyperparameter tuning, complete regression parameter sharing, and separate objective functions.

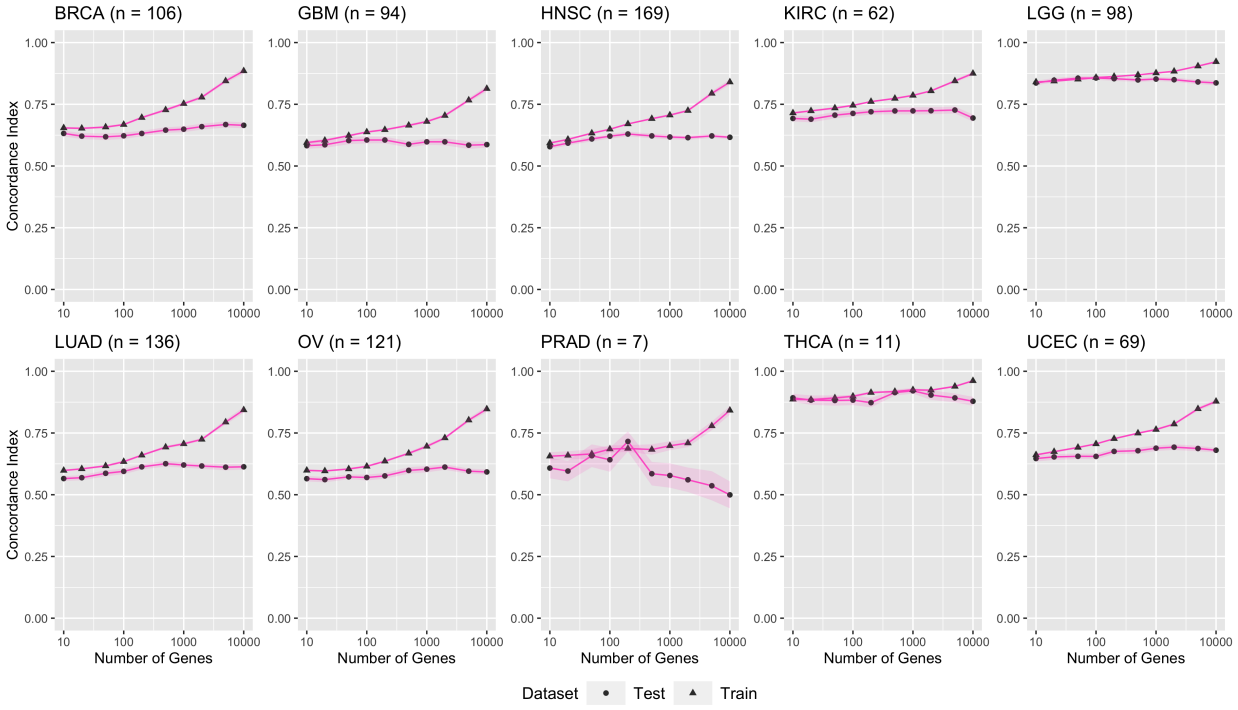


Figure 5.17: Results for Cox Lasso regression with joint feature selection, joint hyperparameter tuning, complete regression parameter sharing, and joint objective functions.

Table 5.3: Density of B Estimates

Cancer Type	Separate Lasso	GFLasso	Joint Lasso
Breast Invasive Carcinoma (BRCA)	0.017	0.061	0.082
Glioblastoma Multiforme (GBM)	0.005	0.040	0.082
Head and Neck Squamous Cell Carcinoma (HNSC)	0.064	0.028	0.082
Kidney Renal Clear Cell Carcinoma (KIRC)	0.008	0.055	0.082
Brain Lower Grade Glioma (LGG)	0.019	0.062	0.082
Lung Adenocarcinoma (LUAD)	0.022	0.071	0.082
Ovarian Serous Cystadenocarcinoma (OV)	0.010	0.060	0.082
Prostate Adenocarcinoma (PRAD)	0.000	0.027	0.082
Thyroid Carcinoma (THCA)	0.000	0.029	0.082
Uterine Corpus Endometrial Carcinoma (UCEC)	0.008	0.046	0.082

5.6 Qualitative Results

The quantitative results summarized in the previous section demonstrate that sharing information across cancer types leads to a significant improvement in the accuracy of cancer survival prediction. In particular, the two best-performing methods in nearly all situations were the ones that employed the largest amount of sharing, namely GFLasso and Joint Lasso. However, in biology, we are not solely interested in predictive power, but are also interested in understanding the biological mechanisms of disease. To that end, in this section, we perform a qualitative comparison of the results that incorporate different amounts of information sharing between the regression parameter estimates in order to see what insights each one reveals about the structure of the problem that we are studying.

For this analysis, we selected a single train/test split of the data in the $p = 1000$ setting and closely examined the regression parameter estimates from each of the three methods. We denote these as $\hat{B}_{\text{gflasso}}$, $\hat{B}_{\text{jlasso}}$, and $\hat{B}_{\text{slasso}}$ for GFLasso, Joint Lasso, and Separate Lasso, respectively. In order to reduce the total number of associations to analyze, we performed some additional post-processing on every $\hat{B}$ by thresholding the regression parameter values at $\epsilon = 0.01$ and setting any element with absolute value smaller than ϵ to 0. This allows us to retain only the strongest associations for closer analysis. The resulting densities of the three $\hat{B}$ estimates for each cancer type are shown in Table 5.3.

In order to visualize the associations between genes and cancer types, we constructed a bipartite graph consisting of gene nodes and cancer type nodes, with edges in the graph representing associations (i.e. nonzero values in the thresholded $\hat{B}$). The graphs for the $\hat{B}$s estimated with all three methods are shown in Figure 5.18.

The top graph shows $\hat{B}_{\text{gflasso}}$. In this visualization, the gene nodes are colored according to their "gene type" as defined in Table 5.4, which categorizes the genes according to their approximate degree in the graph. We define four distinct gene types: non-cancer genes are not associated with any cancer types, and are shown in gray; single-cancer genes are associated with exactly one cancer type, and are shown in green; multi-cancer genes are associated with multiple but not all cancer types, and are shown in blue; pan-cancer genes are associated with all or nearly all cancer types, and are shown in pink. We use the R package ggnet2 to plot the graph, which uses the

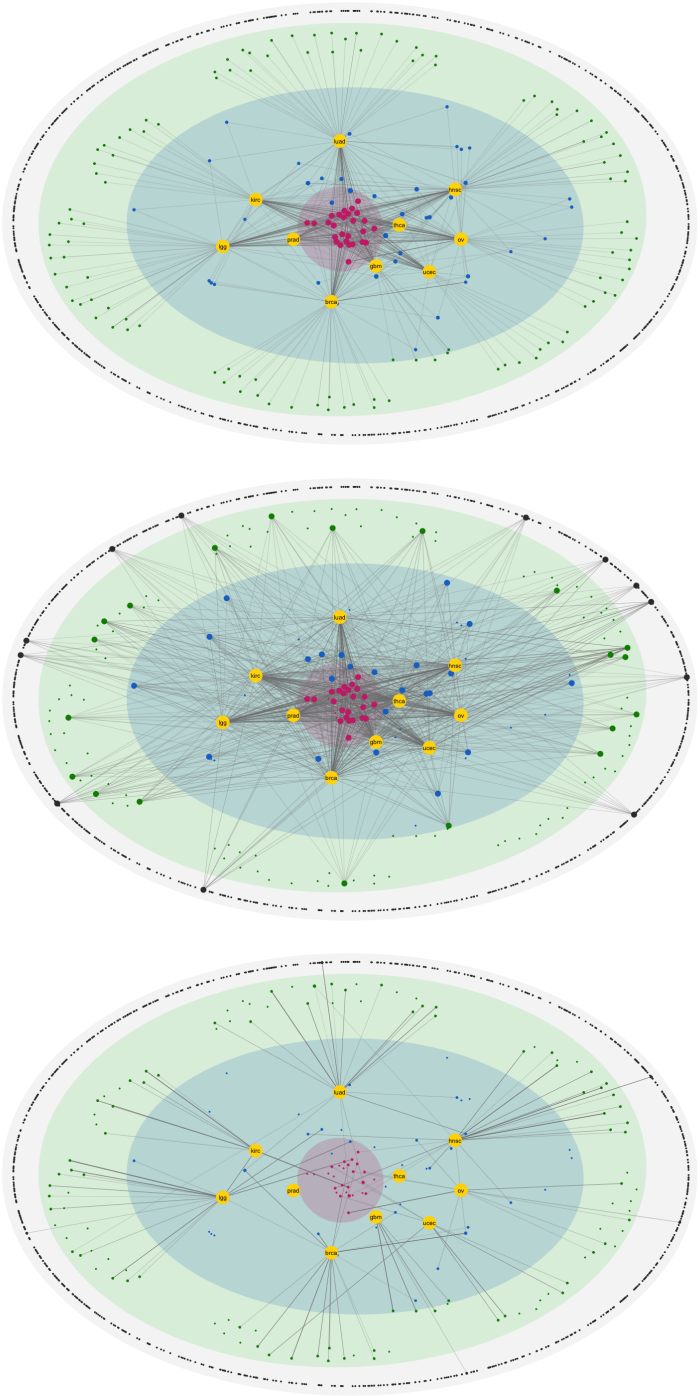


Figure 5.18: A graph representation of the associations between genes and cancer types estimated by GFLasso (top), Joint Lasso (middle), and Separate Lasso (bottom). The nodes corresponding to cancer types are shown in yellow and labeled with their cancer type. All other nodes correspond to individual genes, and the size of these gene nodes is proportional to their degree in the graph. An edge between gene j and cancer type t in the graph represents a nonzero association in $\hat{B}$, and the thickness of the edge represents the strength of the association. The gene nodes in all three graphs are colored according to their gene type in the GFLasso estimate (see Table 5.4 for definitions).

Table 5.4: Gene Type Definitions

Gene Type	Number of Associated Cancers	Node Color
Non-Cancer Gene	0	Gray
Single-Cancer Gene	1	Green
Multi-Cancer Gene	2-7	Blue
Pan-Cancer Gene	8-10	Pink

Table 5.5: Gene Type Counts in B Estimates

Gene Type	Separate Lasso	GFLasso	Joint Lasso
Number of Non-Cancer Genes	888	816	918
Number of Single-Cancer Genes	107	115	0
Number of Multi-Cancer Genes	5	42	0
Number of Pan-Cancer Genes	0	27	82

Fruchterman-Reingold algorithm [31] to determine a node placement that minimizes intersections among edges.

The middle and bottom graphs show $\hat{B}_{\text{jlasso}}$ and $\hat{B}_{\text{slasso}}$, respectively. For the visualizations in Figure 5.18, we kept the node placement unchanged in order to better highlight the contrasts across the three graphs. We also kept the node coloring unchanged, meaning that the node colors in all three graphs reflect the gene type that is determined by $\hat{B}_{\text{gflasso}}$, even though these same genes may have different node degrees in the other two graphs. However, the node sizes and the edges themselves are updated in each graph to reflect that particular estimate of $\hat{B}$.

Overall, these visualizations highlight stark differences between the three estimates. Table 5.5 shows a summary of the number of genes in each category that are identified by each method. The GFLasso estimate is able to capture a compact set of 27 pan-cancer genes, but also identifies a reasonably large set of 115 single-cancer genes, which are distributed across the cancer types. In contrast, Joint Lasso can only consider pan-cancer or non-cancer genes by design, which means that it ends up missing a lot of the single-cancer gene associations that are identified by the other two methods (it only picks up 20 out of the 115 single-cancer genes identified by GFLasso). Finally, Separate Lasso, although it does not have any constraints, seems unable to capture most of the pan-cancer gene associations that are identified by the other two methods, and virtually only identifies single-cancer associations (except for 5 multi-cancer genes, which are all associated with exactly two cancer types).

In order to illustrate the structural differences more clearly, we show the same graphs in Figure 5.19 but with the node placement and node coloring updated to reflect the gene node types determined by each individual $\hat{B}$ estimate. These three graphs have significantly different structure, which underscores the fact that GFLasso is the only one that can uncover nuanced information about which genes are truly pan-cancer, which are multi-cancer, and which are single-cancer. Table 5.6 shows how the four different categories of genes identified by GFLasso are treated in each of the other two estimates. Since GFLasso and Joint Lasso are the two methods that perform best in terms of concordance index, we analyze the differences between these two methods more closely

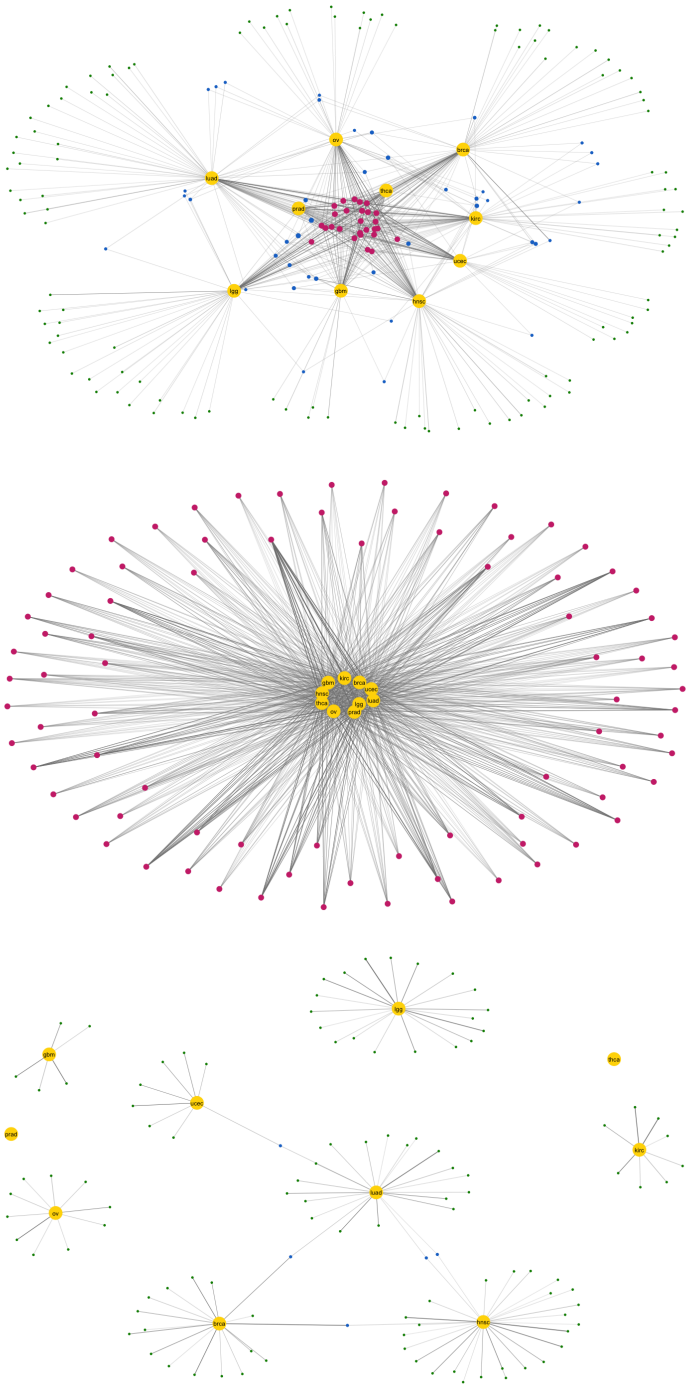


Figure 5.19: A graph representation of the associations between genes and cancer types estimated by GFLasso (top), Joint Lasso (middle), and Separate Lasso (bottom). The nodes corresponding to cancer types are shown in yellow and labeled with their cancer type. All other nodes correspond to individual genes, and the size of these gene nodes is proportional to their degree in the graph. An edge between gene j and cancer type t in the graph represents a nonzero association in $\hat{B}$, and the thickness of the edge represents the strength of the association. The gene nodes are colored according to their gene type in their own respective estimates (see Table 5.4 for definitions).

Table 5.6: Gene Type Overlaps in B Estimates

		Separate Lasso		*Joint Lasso*	
GFLasso Type	Count	Not Selected	Selected	Not Selected	Selected
Non-Cancer	816	806	10	804	12
Single-Cancer	115	49	66	95	20
Multi-Cancer	42	22	20	16	26
Pan-Cancer	27	11	16	3	24
Total	1,000	888	112	918	82

by examining a few genes in detail.

We first look at the 3 genes that were identified by GFLasso as pan-cancer but were not selected by Joint Lasso. Among these, we find FOXD1, which is known to play a role in tumor growth across a wide range of cancer types [47, 67, 34, 37]. It's possible that Joint Lasso missed this pan-cancer gene because it may have different magnitudes of association with different cancer types, which is not something that the Joint Lasso estimate is expressive enough to capture.

Next we examine the set of genes that GFLasso identifies as multi-cancer and that Joint Lasso still selects but (by necessity) identifies as pan-cancer. Among these, we find CAMSAP3, which encodes a protein that suppresses the epithelial to mesenchymal transition of epithelial cells, a process that leads to cancer metastasis [72]. Because of this, increased expression of CAMSAP3 is known to be associated with less metastasis and therefore better cancer prognoses. This is also what we find in both the GFLasso and Joint Lasso estimates; namely, both models identify a negative association between CAMSAP3 expression and survival hazard. However, GFLasso only associates it with 5 cancers out of the full set of 10. Although we don't know for sure whether this multi-cancer association is more accurate than the pan-cancer association identified by Joint Lasso, this again highlights the fact that Joint Lasso does not have as much expressive power as GFLasso to identify these nuanced patterns. There may be some genes that are only associated with carcinomas, or only associated with metastatic cancers, that the Joint Lasso model simply does not have the ability to capture.

Finally we examine the set of genes that GFLasso identifies as single-cancer and that Joint Lasso does not select at all. As shown in Table 5.6, there are 95 genes in this set, which is quite a large fraction of the GFLasso single-cancer genes (83%). This illustrates the greatest weakness of the Joint Lasso method relative to GFLasso, because it leaves behind a number of genes that are known in the literature to be associated with cancer survival. A notable example is SHOX2, which has been independently established as a strong biomarker for predicting LGG survival [107]. In both the GFLasso and Separate Lasso estimates, this gene was in the top 3 genes associated with LGG, but Joint Lasso is not able to pick up the association at all because it is not correlated with survival in any other cancer types.

5.7 Conclusion

In this study, we experimented with sharing information across related cancer types in order to boost the signal to noise ratio of cancer survival prediction. Our results lead to two significant conclusions.

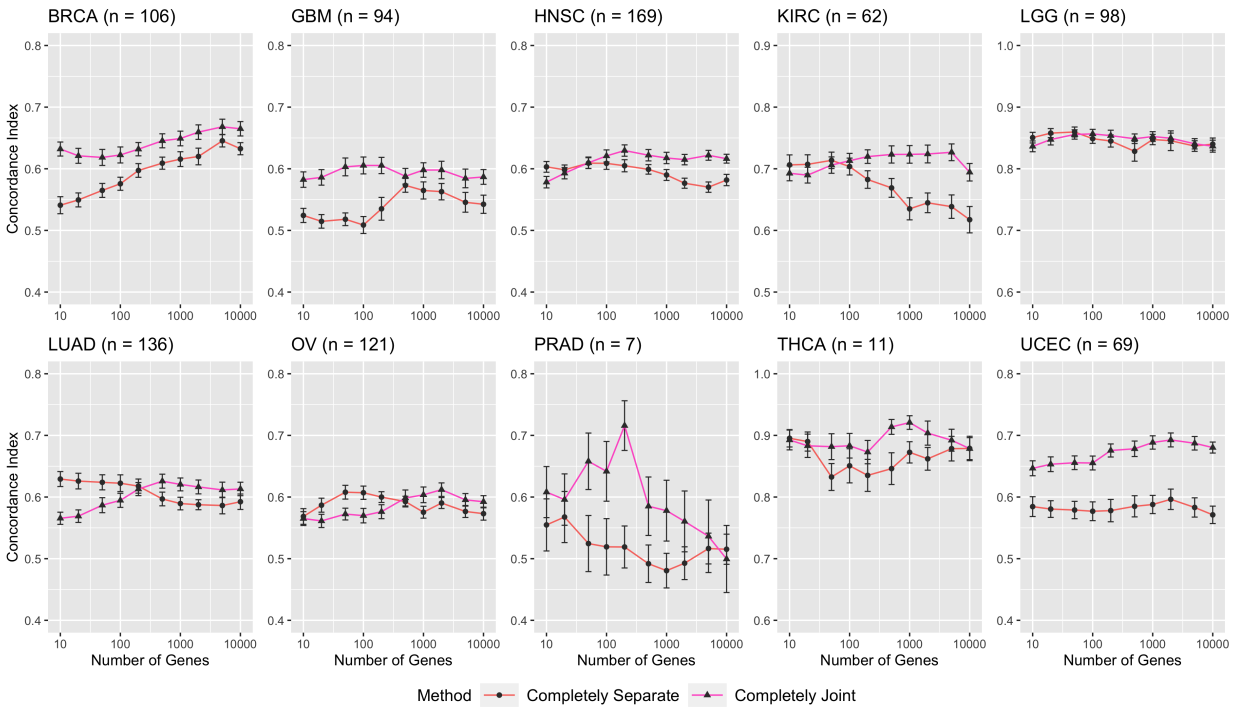


Figure 5.20: Comparison of separate feature selection, separate hyperparameter tuning, separate objective, and no parameter sharing (red) to joint feature selection, joint hyperparameter tuning, joint objective, and full sharing (pink). The fully joint method significantly outperforms the fully separate method on nearly all cancer types in nearly all settings.

First, incorporating sharing leads to a significant improvement on the task of survival prediction relative to the fully separate baseline. In fact, the fully joint model is arguably the method that performs the best on the prediction task. Figure 5.20 drives this point home by comparing the results of the fully separate model side by side with those of the fully joint model. The joint method outperforms the separate method by a significant margin on nearly every cancer type in nearly every setting. The other methods that incorporate significant amounts of sharing, such as GFLasso with the separate objective formulation, are nearly tied for performance with the fully joint model.

Second, we observe that GFLasso has significant advantages over the fully joint model because it is able to capture the rich structure of the underlying problem. In particular, it is able to distinguish between pan-cancer, multi-cancer, and single-cancer genes and identify meaningful associations in all three categories. Ultimately, based on these results, we conclude GFLasso has both the statistical power (due to sharing information between cancer types) and the expressivity (due to not having any hard constraints on B) to identify rich patterns of association between gene expression and cancer survival.

Chapter 6

Conclusion

6.1 Thesis Summary

The central goal of this thesis was to develop and apply machine learning methods that can boost the signal-to-noise ratio when learning from inherently noisy genomic data. To that end, we developed the following novel approaches:

- Time-varying group SpAM boosts the signal of GWAS by leveraging dynamic trait data and incorporating biological context into a time-varying nonparametric regression model [56].
- Inverse covariance fused lasso boosts the signal of eQTL mapping and gene network estimation by using transfer learning to share information between the two highly related tasks [58].
- Hybrid subspace learning estimates an informative latent representation of high-dimensional data by combining the ideas of feature combination and feature selection and enforcing mutual exclusion between low-rank and high-dimensional features [57].
- Multi-task Cox regression with GFLasso improves the performance of cancer survival prediction by using structured sparsity to share information across cancer types while retaining the flexibility to estimate separate regression parameters for each cancer type.

All of these methods follow the overarching framework laid out at the beginning of this thesis in Equation 1.1, which uses regularization penalties to impose structural priors on the problem. In addition to demonstrating that structured sparsity leads to improved performance on prediction tasks, throughout this thesis, we also investigated and highlighted the ability of this class of methods to reveal interesting patterns in the data that provide scientific or medical insights into the biological mechanisms at play. In particular:

- We used time-varying group SpAM to identify specific SNPs with time-varying patterns of influence on childhood asthma. These SNPs may have been missed by previous analyses that only considered static patterns of SNP influence, and studying them further could provide new insights into how the functional mechanisms of asthma change as children grow older.
- We used inverse covariance fused lasso to study Alzheimer's disease and identified several candidate eQTLs that may play a role in Alzheimer's. In particular, we identified a previously discovered connection between Acute Myeloid Leukemia and Alzheime'rs disease. Further investigating the additional eQTLs identified by our method may provide new insights and new connections with other diseases that were not previously known.

- We used hybrid subspace learning to identify aberrant genes whose expression does not fit into any coherent gene modules in cancer patients. This could help reveal driver mutations in cancer, which is a critical step in understanding how individual cancers evolve, grow, and spread in the body.

- We used penalized multi-task Cox regression to identify a set of single-cancer, multi-cancer, and pan-cancer genes. This helps provide greater insight into the overall structure of cancer by revealing which genes play a role in all cancers vs. affect only a subset of cancer types.

Overall, throughout this thesis, we sought to use our models both to generate accurate predictions and to facilitate scientific exploration. We did not develop new statistical methods purely in order to optimize a metric on a dataset, but instead also demonstrated their capability to shed insight into some of the extraordinarily complex functions of biological systems.

6.2 Future Directions

There are many promising areas of future work that can be explored using the ideas presented in this thesis as a starting point. Below we outline a few key directions in broad strokes.

6.2.1 Personalized Learning and Zero-Shot Learning

One of the promises of high-throughput genomic data is that it will facilitate the development of personalized medicine. To that end, one important area of future work is to extend the ideas of transfer learning explored in several sections of this thesis to the personalized setting. In particular, given the Cox GFLasso formulation of Equation 5.3 and using the joint objective formulation of Equation 5.6, we can completely drop the information about which sample belongs to which cancer type and simply learn an individualized regression parameter estimate $\beta^{(i)}$ for each patient. Specifically, we can formulate the optimization problem as follows:

$$\min_{\{\beta^{(i)}\}_{i=1}^{n}} \sum_{i:E_i=1} \left(\log \sum_{j:Y_j \geq Y_i} \exp\{X_j^T \beta^{(j)}\} - X_i^T \beta^{(i)} \right) + \lambda \sum_{i=1}^{n} \|\beta^{(i)}\|_1 + \gamma \sum_{i,j=1}^{n} \|\beta^{(i)} - \beta^{(j)}\|_1 \qquad (6.1)$$

In the above model, we learn an individual set of regression parameters for each sample i, and we use a GFLasso penalty to encourage similarity between all pairs of β's. This can be done without having any prior knowledge about the classes or types of the samples. An open question is how the regression parameter estimates would be used to make predictions for unseen samples at test time. However, even without addressing that problem, analyzing the relationships among the β's that are estimated from the training set would likely still reveal some very interesting patterns and could potentially be used to discover new cancer subtypes.

Closely related to the idea of personalized learning is the idea of learning a model that can make predictions for a task that was never observed at training time, a problem that is known as zero-shot learning. Given that the results from our multi-task survival prediction analysis in Chapter 5 revealed that the best-performing model was the fully joint model, which did not distinguish between cancer types but simply aggregated all of the data, a very promising future direction would be to use this model to make predictions for cancer types that were not even included in the training set. This would be particularly useful for rare cancer types, which suffer from an extreme data shortage. This idea could then be extended to the problem of learning models for rare diseases in general, which have historically been under-studied, due in part to the lack of data [22].

6.2.2 Inferring Latent Structure

The methods developed in this thesis all use structured regularization penalties to impose structural priors on a problem. We then solve an optimization problem and analyze the resulting parameter estimates to understand the structure that they encode. However, the majority of these methods do not explicitly model or estimate the latent structure of the problem, with the possible exception of ICLasso. One potential future direction would be to extend these methods to directly estimate latent structure while simultaneously using the structure to constrain the parameter estimates.

To achieve this, one or more of the models presented this thesis could be recast in a Bayesian framework. As a specific example, the inverse covariance induced fused lasso could be reformulated by assuming the eQTL map $B \in \mathbb{R}^{p \times q}$ is drawn from a Matrix Normal distribution with latent variables for the row-wise and column-wise covariances. Another possible approach would be to design a Mixture of Experts [61] formulation for the Cox Lasso model that jointly estimates a latent cancer type or sub-type along with the regression parameters $\beta^{(t)}$ for that sub-type.

Finally, another potential direction would be to extend the ICLasso multi-task regression formulation described in Chapter 3 and applied to eQTL mapping to the problem of multi-task cancer survival prediction. One key difference between the eQTL setting and the pan-cancer setting is that in the former, we have matched input features $x_i \in \mathbb{R}^p$ and output targets $y_i \in \mathbb{R}^q$ for every sample, and we observe the values of all q prediction tasks for all samples. In contrast, in the pan-cancer setting, each sample belongs to a separate prediction task, or cancer type, which means that we only observe the value of a single prediction task for each sample. This makes it impossible to directly apply ICLasso for cancer survival prediction. However, an interesting direction for future work would be to explore one or more extensions of ICLasso that would allow us to directly estimate a latent graph structure G over the cancer types and then use that similarity graph as weights in the GFLasso penalty for the Cox regression formulation given in Equation 5.3.

6.2.3 Multi-View Learning

In this thesis, we developed models that shared information across related input variables, related output tasks, and related samples. However, one other major area of information sharing is the problem of sharing information across related "views" or data modalities. One example of this is in the cancer genomics setting. Table 6.1 shows the full TCGA dataset that we collected and curated for the pan-cancer study, not all of which was used in the analysis presented in Chapter 5. In addition to collecting gene expression data, we also collected somatic mutations and copy number variations for each patient. These different data modalities can be said to provide different views of the disease being studied. Since the views provide both overlapping and complementary information, we can further boost the signal-to-noise ratio by designing methods that explicitly model the relationships between these data views. This problem is known as multi-view learning.

One exciting direction for future work would be to make use of the full dataset shown in Table 6.1 and design methods for structured multi-view learning. In particular, the hybrid subspace learning method described in Chapter 4 could be extended to the multi-view setting using ideas from canonical correlation analysis [83].

6.2.4 Biological Validation

Finally, another important future direction would be to conduct wet-lab experiments to biologically validate some of the associations inferred by the various methods introduced in this work. This would provide true evidence for the quality of the models, but more importantly, could lead to meaningful new discoveries about the functional mechanisms of particular diseases.

Table 6.1: Full 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)

Since our hypothesis is that structured sparsity increases the power of regression models to detect true associations, it would be particularly interesting to validate some of the associations identified by the structured models that are missed by other models that do not explicitly incorporate structure but still jointly reason about the same set of covariates.

6.3 Closing Thoughts

Although this thesis focused solely on modeling and reasoning about structure in the context of genomics and computational biology, the reality of our world is that data from all domains exhibits rich underlying structure. If this were not the case, we would not be able to use machine learning to uncover meaningful patterns in the first place. This suggests that all machine learning problems are in fact structured prediction problems, even if they are not always formulated that way.

Even with the recent explosion of deep learning and the popularity of connectionism, structured prediction and inductive biases still play an important role in many models and algorithms. The evidence from both this thesis and from the broader machine learning literature indicate that when we design methods that can either uncover this structure or make use of this structure to learn efficiently, we are able to learn meaningful patterns from even a very small number of examples and generalizable patterns that transfer to new settings. Understanding and leveraging this structure is critically important for the future of applied machine learning in all domains.

Bibliography

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

[2] Jordan Anaya, Brian Reon, Wei-Min Chen, Stefan Bekiranov, and Anindya Dutta. A pan-cancer analysis of prognostic genes. *PeerJ*, 3:e1499, 2016.

[3] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[4] Albert-Laszlo Barabasi and Zoltan N Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

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

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

[7] Gabriel F Berriz, Oliver D King, Barbara Bryant, Chris Sander, and Frederick P Roth. Characterizing gene sets with FuncAssociate. *Bioinformatics*, 19(18):2502–2504, 2003.

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

[9] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

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

[11] Rachel B Brem and Leonid 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.

[12] Jeffrey S Breunig, Sean R Hackett, Joshua D Rabinowitz, and Leonid Kruglyak. Genetic basis of metabolome variation in yeast. *PLoS Genetics*, 10(3):e1004142, 2014.

[13] Kenneth P Burnham and David R Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer Science & Business Media, 2003.

[14] Rob A Cairns, Isaac S Harris, and Tak W Mak. Regulation of cancer cell metabolism. *Nature Reviews Cancer*, 11(2):85–95, 2011.

[15] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[16] Safiye Celik, Benjamin A Logsdon, Stephanie Battle, Charles W Drescher, Mara Rendi, R David Hawkins, and Su-In Lee. 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.

[17] Anika Cheerla and Olivier Gevaert. Deep learning with multimodal representation for pancancer prognosis prediction. *Bioinformatics*, 35(14):i446–i454, 2019.

[18] Xi Chen, Seyoung Kim, Qihang Lin, Jaime G Carbonell, and Eric P Xing. Graph-structured multi-task regression and an efficient optimization method for general fused lasso. *arXiv preprint*, arXiv:1005.3579, 2010.

[19] Wei-Yi Cheng, Tai-Hsien Ou Yang, and Dimitris Anastassiou. Biomolecular events in cancer revealed by attractor metagenes. *PLoS Computational Biology*, 9(2):e1002920, 2013.

[20] Wei-Yi Cheng, Tai-Hsien Ou Yang, and Dimitris Anastassiou. Development of a prognostic model for breast cancer survival in an open challenge environment. *Science Translational Medicine*, 5(181):181ra50–181ra50, 2013.

[21] Geraldine M Clarke, Carl A Anderson, Fredrik H Pettersson, Lon R Cardon, Andrew P Morris, and Krina T Zondervan. Basic statistical analysis in genetic case-control studies. *Nature Protocols*, 6(2):121–133, 2011.

[22] Sandra Courbier, Rebecca Dimond, and Virginie Bros-Facer. Share and protect our health data: an evidence based approach to rare disease patients' perspectives on data sharing and data protection-quantitative survey and recommendations. *Orphanet Journal of Rare Diseases*, 14(1):175, 2019.

[23] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

[24] Patrick Danaher, Pei Wang, and Daniela M 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.

[25] Kiranmoy Das, Jiahan Li, Zhong Wang, Chunfa Tong, Guifang Fu, Yao Li, Meng Xu, Kwangmi Ahn, David Mauger, Runze Li, and Rongling Wu. A dynamic model for genome-wide association studies. *Human Genetics*, 129(6):629–639, 2011.

[26] Kiranmoy Das, Jiahan Li, Guifang Fu, Zhong Wang, Runze Li, and Rongling Wu. Dynamic semiparametric Bayesian models for genetic mapping of complex trait with irregular longitudinal data. *Statistics in Medicine*, 32(3):509–523, 2013.

[27] Robert T Dorsam and J Silvio Gutkind. G-protein-coupled receptors and cancer. *Nature Reviews Cancer*, 7(2):79–94, 2007.

[28] Eran Eden, Roy Navon, Israel Steinfeld, Doron Lipson, and Zohar Yakhini. GOrilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists. *BMC Bioinformatics*, 10(1):48, 2009.

[29] Manuel AR Ferreira, Louise O'Gorman, Peter Le Souëf, Paul R Burton, Brett G Toelle, Colin F Robertson, Peter M Visscher, Nicholas G Martin, and David L Duffy. 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.

[30] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[31] Thomas MJ Fruchterman and Edward M Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

[32] Nicholas A Furlotte, Eleazar Eskin, and Susana Eyheramendy. Genome-wide association mapping with longitudinal data. *Genetic Epidemiology*, 36(5):463–471, 2012.

[33] Peisong Gao, Hisako Kawada, T Kasamatsu, Xiao Quang Mao, Mark H Roberts, Yoshihiro Miyamoto, Michihiro Yoshimura, Y Saitoh, Hirofumi Yasue, Kazuwa Nakao, et al. Variants of NOS1, NOS2, and NOS3 genes in asthmatics. *Biochemical and Biophysical Research Communications*, 267(3):761–763, 2000.

[34] Yuan-Feng Gao, Tao Zhu, Xiao-Yuan Mao, Chen-Xue Mao, Ling Li, Ji-Ye Yin, Hong-Hao Zhou, and Zhao-Qian Liu. Silencing of forkhead box D1 inhibits proliferation and migration in glioma cells. *Oncology Reports*, 37(2):1196–1202, 2017.

[35] Timothy S Gardner and Jeremiah J Faith. Reverse-engineering transcription control networks. *Physics of Life Reviews*, 2(1):65–88, 2005.

[36] Justin Guinney, Rodrigo Dienstmann, Xin Wang, Aurélien De Reyniès, Andreas Schlicker, Charlotte Soneson, Laetitia Marisa, Paul Roepman, Gift Nyamundanda, Paolo Angelino, et al. The consensus molecular subtypes of colorectal cancer. *Nature Medicine*, 21(11): 1350–1356, 2015.

[37] Tao Han, Jie Lin, Yannan Wang, Qihao Fan, Haojie Sun, Youmao Tao, and Caixia Sun. Forkhead box D1 promotes proliferation and suppresses apoptosis via regulating polo-like kinase 2 in colorectal cancer. *Biomedicine & Pharmacotherapy*, 103:1369–1375, 2018.

[38] Frank E Harrell, Robert M Califf, David B Pryor, Kerry L Lee, and Robert A Rosati. Evaluating the yield of medical tests. *JAMA*, 247(18):2543–2546, 1982.

[39] Trevor Hastie and Robert Tibshirani. *Generalized Additive Models*. Chapman & Hall, 1990.

[40] Lucia A Hindorff, Jacqueline MacArthur, Joannella Morales, Heather A Junkins, Peggy N Hall, Alan K Klemm, and Teri A Manolio. A catalog of published genome-wide association studies, 2015. URL www.genome.gov/gwastudies.

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

[42] Ian Jolliffe. *Principal Component Analysis*. Wiley Online Library, 2002.

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

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

[45] Seyoung Kim, Kyung-Ah Sohn, and Eric P Xing. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, 25(12):204–212, 2009.

[46] John P Klein and Melvin L Moeschberger. *Survival analysis: techniques for censored and truncated data*. Springer Science & Business Media, 2006.

[47] Dan Li, Suyun Fan, Fei Yu, Xuchao Zhu, Yingchun Song, Meng Ye, Lihong Fan, and Zhongwei Lv. FOXD1 promotes cell growth and metastasis by activation of vimentin in NSCLC. *Cellular Physiology and Biochemistry*, 51(6):2716–2731, 2018.

[48] Jiahan Li, Kiranmoy Das, Guifang Fu, Runze Li, and Rongling Wu. The Bayesian lasso for genome-wide association studies. *Bioinformatics*, 27(4):516–523, 2011.

[49] Jiahan Li, Zhong Wang, Runze Li, and Rongling 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.

[50] Zitong Li and Mikko J Sillanpää. A Bayesian nonparametric approach for mapping dynamic quantitative traits. *Genetics*, 194(4):997–1016, 2013.

[51] Ying-Ju Lin, Jeng-Sheng Chang, Xiang Liu, Hsinyi 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.

[52] Jun Liu, Shuiwang Ji, and Jieping Ye. SLEP, 2009. URL http://www.public.asu.edu/~jye02/Software/SLEP.

[53] Tao Liu, Tanya M Laidlaw, Chunli Feng, Wei Xing, Shiliang Shen, Ginger L Milne, and Joshua A Boyce. 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.

[54] Manasi Malik, Joe Chiles III, Hualin S Xi, Christopher Medway, James Simpson, Shobha Potluri, Dianna Howard, Ying Liang, Christian M Paumi, Shubhabrata Mukherjee, et al. Genetics of CD33 in Alzheimer's disease and acute myeloid leukemia. *Human Molecular Genetics*, 24(12):3557–3570, 2015.

[55] Michelle L Manni, Keven M Robinson, and John F 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.

[56] Micol Marchetti-Bowick, Junming Yin, Judie A Howrylak, and Eric P Xing. A time-varying group sparse additive model for genome-wide association studies of dynamic complex traits. *Bioinformatics*, 32(19):2903–2910, 2016.

[57] Micol Marchetti-Bowick, Benjamin J Lengerich, Ankur P Parikh, and Eric P Xing. Hybrid subspace learning for high-dimensional data. *arXiv preprint arXiv:1808.01687*, 2018.

[58] Micol Marchetti-Bowick, Yaoliang Yu, Wei Wu, and Eric P Xing. A penalized regression model for the joint estimation of eQTL associations and gene network structure. *The Annals of Applied Statistics*, 13(1):248–270, 2019.

[59] Andriy Marusyk and Kornelia Polyak. Tumor heterogeneity: causes and consequences. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 1805(1):105–117, 2010.

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

[61] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293, 2014.

[62] Joel A Mathews, Jill Ford, Sarah Norton, D Kang, Anthony Dellinger, David R Gibb, Andrew Q Ford, Hugh Massay, Christopher Lynn Kepley, Peggy Scherle, 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.

[63] Rahul Mazumder, Jerome H Friedman, and Trevor Hastie. SparseNet: coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495):1125–1138, 2011.

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

[65] Shigenori Nagai and Masakazu Toi. Interleukin-4 and breast cancer. *Breast Cancer*, 7(3): 181–186, 2000.

[66] Kamalpreet Nagpal, Shilpy Sharma, B-Rao Chandrika, Sanober Nahid, Pramod V Niphadkar, Surendra K Sharma, and Balaram Ghosh. TGFβ1 haplotypes and asthma in Indian populations. *Journal of Allergy and Clinical Immunology*, 115(3):527–533, 2005.

[67] Sohei Nakayama, Kenzo Soejima, Hiroyuki Yasuda, Satoshi Yoda, Ryosuke Satomi, Shinnosuke Ikemura, Hideki Terai, Takashi Sato, Norihiro Yamaguchi, Junko Hamamoto, et al. FOXD1 expression is associated with poor prognosis in non-small cell lung cancer. *Anticancer Research*, 35(1):261–268, 2015.

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

[69] Carole Ober and Tsung-Chieh Yao. The genetics of asthma and allergic disease: a 21st century perspective. *Immunological Reviews*, 242(1):10–30, 2011.

[70] Sun-Hee Oh, Yong-Hoon Kim, Se-Min Park, et al. Association analysis of thromboxane a synthase 1 gene polymorphisms with aspirin intolerance in asthmatic patients. *Pharmacogenomics*, 12(3):351–363, 2011.

[71] Jie Peng, Pei Wang, Nengfeng Zhou, and Ji Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746, 2009.

[72] Varisa Pongrakhananon, Onsurang Wattanathamsan, Masatoshi Takeichi, Paninee Chetprayoon, and Pithi Chanvorachote. Loss of CAMSAP3 promotes EMT via the modification of microtubule–Akt machinery. *Journal of Cell Science*, 131(21):jcs216168, 2018.

[73] Srinivasa R Prasad, Peter A Humphrey, Jay R Catena, Vamsi R Narra, John R Srigley, Arthur D Cortez, Neal C Dalrymple, and Kedar N Chintapalli. Common and uncommon histologic subtypes of renal cell carcinoma: imaging spectrum with pathologic correlation. *RadioGraphics*, 26(6):1795–1806, 2006.

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

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

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

[77] Matthew V Rockman and Leonid Kruglyak. Genetics of global gene expression. *Nature Reviews Genetics*, 7(11):862–872, 2006.

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

[79] M Siedlinski, Cleo C van Diemen, Dirkje S Postma, Judith M Vonk, and H Marike Boezen. Superoxide dismutases, lung function and bronchial responsiveness in a general population. *European Respiratory Journal*, 33(5):986–992, 2009.

[80] Kyung-Ah Sohn and Seyoung 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)*, pages 1081–1089, 2012.

[81] Zachary D Stephens, Skylar Y Lee, Faraz Faghri, Roy H Campbell, Chengxiang Zhai, Miles J Efron, Ravishankar Iyer, Michael C Schatz, Saurabh Sinha, and Gene E Robinson. Big data: astronomical or genomical? *PLoS Biology*, 13(7):e1002195, 2015.

[82] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, 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.

[83] Bruce Thompson. Canonical correlation analysis. *Encyclopedia of statistics in behavioral science*, 2005.

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

[85] Paul Van Eerdewegh, Randall D Little, Josée Dupuis, Richard G Del Mastro, Kathy Falls, Jason Simon, Dana Torrey, Sunil Pandit, et al. Association of the ADAM33 gene with asthma and bronchial hyperresponsiveness. *Nature*, 418(6896):426–430, 2002.

[86] Chandrasekar Venkataraman, Kathleen Justen, Jingyong Zhao, Elizabeth Galbreath, and Songqing 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.

[87] Roel GW Verhaak, Katherine A Hoadley, Elizabeth Purdom, Victoria Wang, Yuan Qi, Matthew D Wilkerson, C Ryan Miller, Li Ding, Todd Golub, Jill P Mesirov, 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.

[88] K David Voduc, Maggie CU Cheang, Scott Tyldesley, Karen Gelmon, Torsten O Nielsen, and Hagen Kennecke. Breast cancer subtypes and the risk of local and regional relapse. *Journal of Clinical Oncology*, 28(10):1684–1691, 2010.

[89] Zhong Wang and Jiahan Li. fGWAS2, 2012. URL http://statgen.psu.edu/software/fgwas-r2.html.

[90] Lisandra West, Smruti J. Vidwans, Nicholas P. Campbell, Jeff Shrager, George R. Simon, Raphael Bueno, Phillip A. Dennis, Gregory A. Otterson, and Ravi Salgia. A novel classification of lung cancer into molecular subtypes. *PLoS ONE*, 7(2):1–11, 02 2012.

[91] Tong Tong Wu, Yi Fang Chen, Trevor Hastie, Eric Sobel, and Kenneth Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, 2009.

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

[93] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust PCA via outlier pursuit. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2496–2504, 2010.

[94] Xinsen Xu, Lei Huang, Chun Hei Chan, Tao Yu, Runchen Miao, and Chang Liu. Assessing the clinical utility of genomic expression data across human cancers. *Oncotarget*, 7(29):45926, 2016.

[95] Jie Yang, Rongling Wu, and George Casella. Nonparametric functional mapping of quantitative trait loci. *Biometrics*, 65(1):30–39, 2009.

[96] Mi Young Yang, Mary Beth Hilton, Steven Seaman, et al. Essential regulation of lung surfactant homeostasis by the orphan G protein-coupled receptor GPR116. *Cell Reports*, 3 (5):1457–1464, 2013.

[97] Junming Yin, Xi Chen, and Eric P Xing. Group sparse additive models. In *International Conference on Machine Learning (ICML)*, pages 871–878, 2012.

[98] Jie You, Wei Peng, Xu Lin, Qing-Ling Huang, and Jian-Yin 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.

[99] Safoora Yousefi, Fateme Amrollahi, Mohamed Amgad, Coco Dong, Joshua E Lewis, Congzheng Song, David A Gutman, Sameer H Halani, Jose Enrique Velazquez Vega, Daniel J Brat, et al. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Scientific Reports*, 7(1), 2017.

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

[101] Ming Yuan and Yi 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.

[102] Xiao-Tong Yuan and Tong Zhang. Partial Gaussian graphical model estimation. *IEEE Transactions on Information Theory*, 60(3):1673–1687, 2014.

[103] Yuan Yuan, Eliezer M Van Allen, Larsson Omberg, Nikhil Wagle, Ali Amin-Mansour, Artem Sokolov, Lauren A Byers, Yanxun Xu, Kenneth R Hess, Lixia Diao, et al. Assessing the clinical utility of cancer genomic and proteomic data across tumor types. *Nature Biotechnology*, 32 (7):644–652, 2014.

[104] Bin Zhang, Chris Gaiteri, Liviu-Gabriel Bodea, Zhi Wang, Joshua McElwee, Alexei A Podtelezhnikov, Chunsheng Zhang, Tao Xie, Linh Tran, Radu Dobrin, et al. Integrated systems approach identifies genetic nodes and networks in late-onset Alzheimer's disease. *Cell*, 153(3):707–720, 2013.

[105] M-Q Zhang and Henk Timmerman. Mast cell tryptase and asthma. *Mediators of Inflammation*, 6(5-6):311–317, 1997.

[106] Xiaoke Zhang, Byeong U Park, and Jane-Ling Wang. Time-varying additive models for longitudinal data. *Journal of the American Statistical Association*, 108(503):983–998, 2013.

[107] Yu-An Zhang, Yunyun Zhou, Xin Luo, Kai Song, Xiaotu Ma, Adwait Sathe, Luc Girard, Guanghua Xiao, and Adi F Gazdar. SHOX2 is a potent independent biomarker to predict survival of WHO grade II–III diffuse gliomas. *EBioMedicine*, 13:80–89, 2016.

[108] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.