Bioinformatics, 33, 2017, i350–i358
doi: 10.1093/bioinformatics/btx265
ISMB/ECCB 2017

Predicting phenotypes from microarrays using amplified, initially marginal, eigenvector regression

Lei Ding and Daniel J. McDonald*

Department of Statistics, Indiana University, Bloomington, IN 47405, USA

*To whom correspondence should be addressed.

Abstract

Motivation: The discovery of relationships between gene expression measurements and phenotypic responses is hampered by both computational and statistical impediments. Conventional statistical methods are less than ideal because they either fail to select relevant genes, predict poorly, ignore the unknown interaction structure between genes, or are computationally intractable. Thus, the creation of new methods which can handle many expression measurements on relatively small numbers of patients while also uncovering gene–gene relationships and predicting well is desirable.
Results: We develop a new technique for using the marginal relationship between gene expression measurements and patient survival outcomes to identify a small subset of genes which appear highly relevant for predicting survival, produce a low-dimensional embedding based on this small subset, and amplify this embedding with information from the remaining genes. We motivate our methodology by using gene expression measurements to predict survival time for patients with diffuse large B-cell lymphoma, illustrate the behavior of our methodology on carefully constructed synthetic examples, and test it on a number of other gene expression datasets. Our technique is computationally tractable, generally outperforms other methods, is extensible to other phenotypes, and also identifies different genes (relative to existing methods) for possible future study.
Availability and Implementation: All of the code and data are available at http://mypage.iu.edu/∼dajmcdon/research/.
Contact: dajmcdon@indiana.edu
Supplementary information: Supplementary material is available at *Bioinformatics* online.

1 Introduction

A typical scenario in genomics is to obtain expression measurements for thousands of genes from microarrays or RNA-Seq which may be relevant for predicting a particular phenotype. Such studies have been useful in relating specific genetic variations to a wide variety of outcomes such as disease specific indicators (Lesage and Brice, 2009; Barrett *et al.*, 2008; Burton *et al.*, 2007; Sladek *et al.*, 2007); drug or vaccine response (Saito *et al.*, 2016; Kennedy *et al.*, 2012); and individual traits like motion sickness (Hromatka *et al.*, 2015) or age at menarche (Elks *et al.*, 2010; Perry *et al.*, 2014). In these scenarios, researchers are interested in the accurate prediction of the phenotype and the identification of a handful of relevant genes with a reasonable computational expense. With these goals in mind, supervised linear regression techniques such as ridge regression (Hoerl and Kennard, 1970), the lasso (Tibshirani, 1996), the Dantzig selector (Candes and Tao, 2007) or other penalized methods are often employed.

However, because phenotypes tend to be the result of groups of genes, which perhaps together describe more complicated biomechanical processes, rather than individual polymorphisms, recent approaches have tried to account for this group structure. Techniques such as the group lasso (Yuan and Lin, 2006) can predict the response with sparse groupings of coefficients as long as the groups are partially understood ahead of time. In contrast, unsupervised methods such as principal components analysis (Hotelling, 1957; Jolliffe, 2002; Pearson, 1901) are often used directly on the genes when no phenotype is being examined (Alter *et al.*, 2000; Sladek *et al.*, 2007; Wall *et al.*, 2003). Finally, modern approaches developed specifically for the genomics context such as supervised gene shaving (Hastie *et al.*, 2000), tree harvesting (Hastie *et al.*, 2001), and supervised principal components (Bair and Tibshirani, 2004; Bair *et al.*, 2006) have sought to combine the presence of a response with the structure estimation properties of eigendecompositions from unsupervised techniques to obtain the best of both. It is this last set of techniques that most closely resemble the approach we present here. We give a more detailed discussion of supervised principal components next, before motivating our method with an example.

Notation: We will use bolded letters $\mathbf{M}$ to indicate matrices, capital letters to denote column vectors, such that M_j is the j^{th} column of the matrix $\mathbf{M}$, and lower case letters m_i to denote row vectors (a single subscript) or scalars (m_{ij} being the i, j element of $\mathbf{M}$). We will use the notation $\mathbf{M}_A$ to mean the columns of $\mathbf{M}$ whose indices are in the set A and $[k] = \{1, \ldots, k\}$. Finally, for a matrix $\mathbf{M}$, we write the singular value decomposition (SVD) of $\mathbf{M} = \mathbf{U}(\mathbf{M})\Lambda(\mathbf{M})\mathbf{V}(\mathbf{M})^\top$ and define $\mathbf{M}^\dagger$ to be the Moore-Penrose inverse of $\mathbf{M}$. In the case only of the design matrix $\mathbf{X}$ discussed below, we will use the more compact decomposition $\mathbf{X} = \mathbf{U}\Lambda\mathbf{V}^\top$.

1.1 Supervised eigenstructure techniques

The first technique for extending unsupervised principal components analysis to the case where a response is available is principal components regression (PCR, Hotelling, 1957; Kendall, 1965). Instead of regressing the response on all the available covariates as in ordinary least squares (OLS), PCR first performs an eigendecomposition of the empirical covariance matrix and then regresses the response on the subset of principal components corresponding to the largest variances. Defining $Y \in R^n$ to be the centered response vector, and $\mathbf{X}$ to be the $n \times p$ centered design matrix, write the (reduced) SVD of $\mathbf{X}$ as $\mathbf{X} = \mathbf{U}\Lambda\mathbf{V}^\top$. For some integer $d \le p$, the principal components regression estimator is given as the solution to

$$\widehat{\Gamma}_{PCR} = \underset{\Gamma}{\operatorname{argmin}} \, ||Y - \mathbf{U}_{[d]}\Lambda_{[d]}\Gamma||_2^2,$$

which has the closed form representation

$$\widehat{\Gamma}_{PCR} = ((\mathbf{U}_{[d]}\Lambda_{[d]})^\top \mathbf{U}_{[d]}\Lambda_{[d]})^{-1}(\mathbf{U}_{[d]}\Lambda_{[d]})^\top Y = \Lambda_{[d]}^{-1}\mathbf{U}_{[d]}^T Y.$$

Since this solution is in the space spanned by the principal components, it is easy to rotate the estimate back onto the span of $\mathbf{X}$: $\widehat{\beta}_{PCR} := \mathbf{V}_{[d]}\widehat{\Gamma}_{PCR} = \mathbf{V}_{[d]}\Lambda_{[d]}^{-1}\mathbf{U}_{[d]}^T Y$. Then any elements of $\widehat{\beta}_{PCR}$ which are identically zero imply the irrelevance of those genes for predicting the phenotype while the columns of $\mathbf{V}_{[d]}^\top$ can be interpreted as indicating groupings of individual genes.

Principal components regression performs well under certain conditions when we believe that there are natural groupings of covariates (linear combinations) which are useful for predicting the response. However, Lu (2002) and Johnstone and Lu (2009) show that the empirical singular vectors $\mathbf{U}_{[d]}$ are poor estimates of the associated population quantity (the left singular vectors of the expected value of $\mathbf{X}$) unless $p/n \to 0$ as $n \to \infty$. In particular, when $p \gg n$, as is common in genomics where the number of gene expression measurements is much larger than the number of patients, PCR will suffer.

To avoid this flaw in PCR, various approaches have been proposed. Hastie *et al.* (2000) proposed a method called 'gene shaving' that is applicable to both supervised (given a phenotype) and unsupervised (only gene expressions) settings. In the supervised setting, it works by computing the first principal component and ranking the genes using a combined measure that balances the principal component scores and the marginal relationship with the response. Those genes with lowest combined scores are removed and the process is repeated until only one gene remains, resulting in a nested sequence of clusters containing fewer and fewer genes. Then one chooses a cluster along this sequence, orthogonalizes the data with respect to the genes in that cluster, and repeats the entire process again, iterating until the desired number of clusters has been recovered. This procedure is somewhat computationally expensive as well as requiring both the cluster sizes and the number of clusters to be chosen.

An alternative with somewhat similar behavior is supervised principal components (SPC, Bair and Tibshirani, 2004; Bair *et al.*, 2006). SPC avoids the high-dimensional regression problem by first selecting a much smaller subset of useful genes which have high marginal correlation with the phenotype (in contrast to gene shaving, which uses the marginal correlation and the covariance between genes). By screening out most of the hopefully irrelevant genes, we can return to the scenario where $p < n$. In follow-up work, Paul *et al.* (2008) show that, if a small marginal correlation with the response implies irrelevance for prediction, then SPC will find any truly relevant genes and predict the phenotype accurately. They also suggest using lasso or forward stepwise selection after SPC to further reduce the number of genes. However, if some genes have small marginal relationship with the response but large conditional relationship, they will be erroneously ignored by SPC. It is this last property that our method attempts to correct. We now illustrate that the screening step of SPC is likely to remove important genes in typical applications before discussing how our procedure avoids suffering the same fate.

1.2 A motivating example

To motivate our methodology in relation to previous approaches, we examine a dataset consisting of 240 patients with diffuse large B-cell lymphoma (DLBCL, Rosenwald *et al.*, 2002) in some detail. Each patient is measured on 7399 genes, and her survival time is recorded. Previous approaches rely on the assumption that a small marginal correlation between the response variable, in this case patient survival time, and the vector of expression measurements for a particular gene is sufficient for guaranteeing the irrelevance of that particular gene for prediction. To make this assumption mathematically precise, suppose $y = x^\top\beta + \epsilon$, where y is the response, x is a vector of gene expression measurements, and ϵ is a mean-zero error. Then, the assumption can be stated mathematically as $\text{Cov}(x_j, y) = 0 \Rightarrow \beta_j = 0$. While reasonable under some conditions, this assumption is perhaps too strong for many gene expression datasets. Very often, individual gene expressions are only predictive of phenotype in the presence of other genes. We can rewrite this assumption using the population covariance matrix between genes, $\text{Cov}(x, x) = \Sigma_{xx}$, and the vector-valued covariance between gene expressions and phenotype, $\text{Cov}(x, y) = \Sigma_{xy}$. Then, using the population equation for β allows us to rewrite the assumption as

$$(\Sigma_{xy})_j = 0 \Rightarrow \beta_j = (\Sigma_{xx}^{-1}\Sigma_{xy})_j = 0. \tag{1}$$

In words, we are assuming that the dot product of the j^{th} row of the inverse covariance matrix with the covariance between x and y is zero whenever the j^{th} element of Σ_{xy} is zero.

To examine whether this assumption holds, we can estimate both Σ_{xx}^{-1} and Σ_{xy} using the DLBCL data and imagine that these estimates are the population quantities for illustration. To estimate Σ_{xy}, we use the standard covariance estimate, but set all but the largest 120 values equal to zero, corresponding to a sparse solution. For the case of Σ_{xx}^{-1}, estimating large inverse covariance matrices accurately is impossible when $p \gg n$ unless we assume some additional structure. If most of the entries are 0 [a necessary condition for (1) to hold], methods like the graphical lasso (glasso, Friedman *et al.*, 2008) or graph estimation (Meinshausen and Bühlmann, 2006) have been shown to work well. We use the graph estimation technique for all 7399 genes in the dataset at 10 different sparsity levels ranging from 100% to 99.2%. For visualization purposes, Figure 1 shows the first 250 genes for one estimate of the inverse covariance that is 97.5% sparse.

To assess the validity of (1), Table 1 shows the sparsity of the full inverse covariance matrix, the percentage of non-zero regression coefficients, and the percentage of non-zero regression coefficients which are incorrectly ignored by the assumption (the false negative rate). In all cases, Σ_{xy} is about 98% sparse. Even with an extremely sparse inverse covariance matrix, the false negative rate is at least 25% meaning that 25% of possibly relevant genes are ignored by the analysis. If the sparsity of Σ_{xx}^{-1} is allowed to increase only slightly, the false negative rate increases to over 95%.

1.3 Our contribution

For a similar computational budget, our method outperforms existing approaches by taking advantage of all the data. Our method does not require that the set of non-zero regression coefficients be a subset of the non-zero marginal correlations.

Suppose that $\mathbf{M} \in R^{p\times p}$ is a symmetric, non-negative definite matrix; that is, for all vectors $a \in R^p$, $a^\top \mathbf{M} a \geq 0$ and $\mathbf{M}^\top = \mathbf{M}$. To approximate the matrix $\mathbf{M}$, we fix an integer $\ell \ll p$ and form a sketching matrix $\mathbf{S} \in R^{p\times \ell}$. Then, we report the following approximation: $\mathbf{M} \approx (\mathbf{MS})(\mathbf{S}^\top\mathbf{MS})^\dagger(\mathbf{MS})^\top$. The details behind the formation of the matrix $\mathbf{S}$ control the type of approximation.

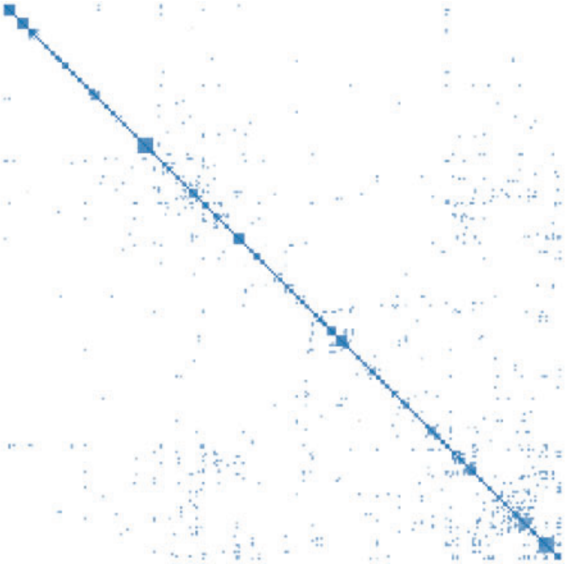

Fig. 1. A sparse estimate of the inverse covariance of gene expression measurements for the first 250 genes from the DLBCL dataset. The estimate has 97.5% of the off-diagonal elements equal to 0. Darker colors represent inverse co-variances of larger magnitude

In the simplest case, which we employ here, we take $\mathbf{S} = \pi\tau$, where $\pi \in R^{p\times p}$ is a permutation of the identity matrix and $\tau = [\mathbf{I}_\ell, 0]^\top \in R^{p\times \ell}$ is a truncation matrix. While many alternative sketching matrices, mostly based on random projections, have been proposed, this method is the only one necessary to develop our results. Without loss of generality, divide the matrix $\mathbf{M}$ into blocks

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{21}^\top \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}$$

so that we can (implicitly) construct the matrix $\mathbf{F}(\mathbf{M}) \in R^{p\times \ell}$ as

$$\mathbf{F}(\mathbf{M}) := \mathbf{MS} = \begin{bmatrix} \mathbf{M}_{11} \\ \mathbf{M}_{21} \end{bmatrix}.$$

Because

$$\mathbf{M} \approx (\mathbf{MS})(\mathbf{S}^\top\mathbf{MS})^\dagger(\mathbf{MS})^\top = \mathbf{F}(\mathbf{M})(\mathbf{S}^\top\mathbf{MS})^\dagger\mathbf{F}(\mathbf{M})^\top,$$

we can approximate the eigendecomposition of $\mathbf{M}$ using the SVD of $\mathbf{F}(\mathbf{M})$. If we decompose $\mathbf{F} = \mathbf{U}(\mathbf{F})\Lambda(\mathbf{F})\mathbf{V}(\mathbf{F})^\top$, where we have suppressed the dependence of $\mathbf{F}$ on $\mathbf{M}$ when $\mathbf{F}$ is an argument for clarity, then the resulting approximation to the eigenvectors of $\mathbf{M}$ is $\mathbf{V}(\mathbf{M}) \approx \mathbf{F}\mathbf{V}(\mathbf{F})\Lambda(\mathbf{F})^\dagger = \mathbf{U}(\mathbf{F})$. Likewise, the approximate eigenvalues of $\mathbf{M}$ are given the singular values $\Lambda(\mathbf{F})$.

Homrighausen and McDonald (2016) show that this approximation is more accurate than the one based on $\mathbf{M}_{11}$ for performing a principal components analysis. As previous techniques for principal components regression (like SPC) are based on $\mathbf{M}_{11}$ rather than $\mathbf{F}$, it is possible that by using $\mathbf{F}$, we will have better results. As we will see, this intuition turns out to be true under some conditions which were suggested in Section 1.2. In particular, for essentially the same computational budget, our procedure outperforms previous procedures if some genes have small marginal correlations with the phenotype but are, nonetheless, important for predicting the phenotype conditional on the presence of other genes. Furthermore, even if the assumption in (1) is true, our procedure is not much worse than existing approaches.

In Section 2, we discuss exactly how to implement our methodology. We examine the behavior of our procedure in Section 3. In Section 3.1, we state an explicit model for the data-generating mechanism in order to be clear about the conditions under which our procedure works well. Section 3.2 uses a number of carefully constructed simulations to show when our technique works well, and when it doesn't. In Section 4, we examine our procedure on four genetics datasets, including the one discussed above. We find that our methods slightly outperform existing techniques on three of them, suggesting that the motivation is sound. Finally, in Section 5, we give conclusions and discuss some avenues for future work.

Table 1. This table shows properties of the coefficients of the linear model corresponding to 10 different estimates of the inverse covariance matrix, from complete sparsity on the left (a diagonal matrix) to still more than 99% sparsity on the right

Sparsity of Σ_{xx}^{-1}	1.0000	0.9999	0.9998	0.9995	0.9991	0.9984	0.9975	0.9963	0.9946	0.9922
% Non-zero β's	0.0162	0.0216	0.0287	0.0418	0.0618	0.0843	0.1193	0.1803	0.2645	0.3699
False Negative Rate	0.0000	0.2500	0.4340	0.6117	0.7374	0.8077	0.8641	0.9100	0.9387	0.9562

Note: The second row is the number of non-zero population regression coefficients corresponding to each inverse covariance matrix. The bottom row shows the percentage of non-zero regression coefficients which are incorrectly ignored under the assumption on the relationship between marginal correlations and regression coefficients.

2 Methods and computations

We now give the details of our methodology. For clarity, we assume that the design matrix $\mathbf{X}$ and the response Y are already centered. Let T be a p-dimensional vector denoting standardized regression coefficient estimates i.e. for any $j \in \{1, 2, \dots, p\}$, t_j is the coefficient estimate of standardized univariate regression between response Y and covariate X_j. We use standardized regression so that the coefficient estimates are comparable across disparate covariates. Note that t_j is also the marginal correlation between the response Y and covariate X_j.

For some threshold t_*, we separate $\mathbf{X}$ into two matrices $\mathbf{X}_A$ and $\mathbf{X}_{A^c}$, where $A = \{j : |t_j| > t_*\}$. We assume $|A| = \ell$. The hope is that $\mathbf{X}_A$ contains many of the genes that are most predictive of the phenotype under study. Ideally, high marginal correlations will suggest relevant predictors to be emphasized in the decomposition, but unlike other methods, we will also use those genes in the set A^c. We now focus on $\mathbf{X}_{\text{new}} = [\mathbf{X}_A, \mathbf{X}_{A^c}]$ and note that it has the same range as $\mathbf{X}$. Therefore, we will use the approximation technique discussed in Section 1.3 to try to estimate the eigendecomposition of $\boldsymbol{\Sigma}_{\text{xx}}$ using sample quantities. Because $\mathbf{X}_{\text{new}}^\top \mathbf{X}_{\text{new}}$ is symmetric and positive definite, write

$$\mathbf{F} = \mathbf{X}_{\text{new}}^\top \mathbf{X}_A = \begin{pmatrix} \mathbf{X}_A^\top \mathbf{X}_A \\ \mathbf{X}_{A^c}^\top \mathbf{X}_A \end{pmatrix},$$

and decompose $\mathbf{F} = \mathbf{U}(\mathbf{F})\Lambda(\mathbf{F})\mathbf{V}(\mathbf{F})$. For some integer $d \in \{1, \dots, \ell\}$, we define

$$\begin{aligned} \widehat{\mathbf{V}}_{[d]} &= \mathbf{U}_{[d]}(\mathbf{F}), \\ \widehat{\boldsymbol{\Lambda}}_{[d]} &= \Lambda_{[d]}(\mathbf{F})^{1/2}, \qquad \text{and} \\ \widehat{\mathbf{U}}_{[d]} &= \mathbf{X}_{new} \widehat{\mathbf{V}}_{[d]} \widehat{\boldsymbol{\Lambda}}_{[d]}^{-1}. \end{aligned}$$

Now, we have estimates for the principal components $\widehat{\mathbf{U}}_{[d]}\widehat{\Lambda}_{[d]}$. Therefore, just as with principal components regression, we can regress Y on the estimated principal components to produce estimated coefficients in principal component space:

$$\widehat{\Gamma}_{AIMER} = \underset{\Gamma}{\operatorname{argmin}} \, ||Y - \widehat{\mathbf{U}}_{[d]}\widehat{\Lambda}_{[d]}\Gamma||_2^2 = \widehat{\Lambda}_{[d]}^{-1} \widehat{\mathbf{U}}_{[d]}^T Y.$$

Then the coefficient estimates for linear regression in the space spanned by $\mathbf{X}_{new}$ are given by

$$\widehat{\beta}_{AIMER} = \widehat{\mathbf{V}}_{[d]} \widehat{\Gamma}_{AIMER} = \widehat{\mathbf{V}}_{[d]} \widehat{\Lambda}_{[d]}^{-1} \widehat{\mathbf{U}}_{[d]}^T Y. \tag{2}$$

Because our methodology uses marginal regression to select a small number of hopefully relevant predictors before 'amplifying' their eigenstructure information with the $\mathbf{F}$ matrix, we refer to our technique as 'Amplified, Initially Marginal, Eigenvector Regression' (AIMER).

Unlike previous approaches, the solution given by (2) is not sparse: with probability 1, $(\widehat{\beta}_{AIMER})_j \neq 0, \forall j$. However, most of the coefficients will be small. We therefore threshold the estimates to produce our final estimator:

$$\widehat{\beta}_{AIMER}(b) := \widehat{\beta}_{AIMER} \mathbf{1}_{(b,\infty)}(|\widehat{\beta}_{AIMER}|), \tag{3}$$

where $b \geq 0$, and $\mathbf{1}_A(w)$ is the indicator function, which returns the value one for every element of $w \in A$ and zero otherwise. We summarize this procedure in Algorithm 1. As with SPC, the computational burden of our method is dominated by the SVD. We use an SVD of $\mathbf{F}$ while SPC uses the SVD of $\mathbf{X}_A$. However, since the SVD is cubic in the smaller dimension, in both cases the computation is $O(|A|^3)$. Thus, to leading order, both methods require the same amount of computation.

Algorithm 1: Amplified, Initially Marginal, Eigenvector Regression (AIMER)

```
Input: centered design matrix $\mathbf{X}$, centered response $Y$, thresholds $t_*, b_* \geq 0$, integer $d$
  Compute marginal correlation $t_j$ between $X_j$ and $Y$ for all $j$;
  Set $A = \{j : |t_j| > t_*\}$;
  Set $\mathbf{X}_{new} = [\mathbf{X}_A, \mathbf{X}_{A^c}]$;;
  Define $\mathbf{F} = \mathbf{X}_{new}^\top \mathbf{X}_A$;
  Decompose $\mathbf{F} = \mathbf{U}(\mathbf{F})\boldsymbol{\Lambda}(\mathbf{F})\mathbf{V}(\mathbf{F})^\top$;
  Set $\widehat{\mathbf{V}}_{[d]} = \mathbf{U}_{[d]}(\mathbf{F})$;
  Set $\widehat{\Lambda}_{[d]} = \Lambda_{[d]}(\mathbf{F})^{1/2}$;
  Set $\widehat{\mathbf{U}}_{[d]} = \mathbf{X}_{new}\widehat{\mathbf{V}}_{[d]}\widehat{\boldsymbol{\Lambda}}_{[d]}^{-1}$;
  Calculate $\widehat{\beta} = \widehat{\mathbf{V}}_{[d]}\widehat{\Lambda}_{[d]}^{-1}\widehat{\mathbf{U}}_{[d]}^T \mathbf{Y}$;
  Set $\widehat{\beta}(b_*) := \widehat{\beta}\mathbf{1}_{(b_*,\infty)}(|\widehat{\beta}|)$;
Output: coefficient estimates $\widehat{\beta}(b_*)$
```

To make predictions given a new observation x_*, we simply center it using the mean of the original data, reorder its entries to conform to $\mathbf{X}_{new}$, multiply by the coefficient vector in (3), and add the mean of the original response vector.

3 Experimental analysis

To examine the performance of our method, we set up a number of carefully constructed simulations under various conditions. We first discuss the generic data model we assume, a latent factor model, which is amenable to analysis via SPC or AIMER.

3.1 Data model

Consider the multivariate Gaussian linear regression model

$$y = x^\top \beta + \sigma_1 \epsilon \tag{4}$$

with y the response, $x \in R^p$ a column vector of gene expression measurements, $\beta = (\beta_1, \dots, \beta_p)^\top$ the coefficients, ϵ a random Gaussian distributed error with zero mean and variance 1, and $\sigma_1 > 0$. We further assume that $x \sim N_p(0, \Sigma_{xx})$ has a Gaussian distribution with mean vector 0 and covariance matrix Σ_{xx}. We will assume that β is sparse, in that most of its elements are exactly 0 indicating no linear relationship between the associated gene and the response. Finally, the design matrix $\mathbf{X}$ and the response vector Y include n independent observations of x and y, respectively.

Model for X.

As Σ_{xx} is symmetric and positive (semi-) definite, we can decompose it as

$$\begin{aligned} \Sigma_{xx} &= \mathbf{V}(\Sigma_{xx})\mathbf{L}(\Sigma_{xx})\mathbf{V}^T(\Sigma_{xx}) \\ &= \begin{pmatrix} V_1 & \cdots & V_p \end{pmatrix} \begin{pmatrix} l_1 & & 0 \\ & \ddots & \\ 0 & & l_p \end{pmatrix} \begin{pmatrix} V_1^\top \\ \vdots \\ V_p^\top \end{pmatrix}, \end{aligned}$$

where $V_1, \dots, V_p$ are orthonormal eigenvectors on R^p and $l_1 \geq \cdots \geq l_p \geq 0$ are eigenvalues. We assume that there is some $1 \leq G \leq p$ such that the eigenvalues can be separated into two groups, one of which includes relatively large eigenvalues and the other relatively small eigenvalues, that is, $l_k = \lambda_k + \sigma_0^2$ for $1 \leq k \leq G$ and $l_k = \sigma_0^2$ for $k > G$ where $\lambda_1 \geq \cdots \geq \lambda_G > 0$, and $\sigma_0^2 > 0$.

Then, because $\mathbf{X}$ is multivariate Gaussian, we can write $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{U}_G \mathbf{\Lambda}_G \mathbf{V}_G^T + \sigma_0 \mathbf{E}$$
$$= \begin{pmatrix} U_1 & \cdots & U_G \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & & 0 \\ & \ddots & \\ 0 & & \sqrt{\lambda_G} \end{pmatrix} \begin{pmatrix} V_1^\top \\ \vdots \\ V_G^\top \end{pmatrix} + \sigma_0 \mathbf{E}$$

where latent factors $U_1, \ldots, U_G$ are independent and identically distributed (i.i.d.) $N_n(0, \mathbf{I})$ vectors, and the noise matrix $\mathbf{E}$ is $n \times p$ with i.i.d. $N(0, 1)$ entries independent of $U_1, \ldots, U_G$.

Model for Y.

We assume that Y is a linear function of the first $K \leq G$ latent factors in $\mathbf{U}_G$ plus additive Gaussian noise: $Y = \mathbf{U}_K \Theta + \sigma_1 Z$, where Θ is the coefficient vector, $\sigma_1 > 0$ is a constant, and Z is distributed $N_n(0, \mathbf{I})$, independent of $\mathbf{X}$. Note that the expectation of Y is zero and that this is a specific form of (4).

Implication of the model.

Under this model for $\mathbf{X}$ and Y, the population marginal covariance between each gene X_j and the response Y can be written as

$$\Sigma_{xy} = \begin{pmatrix} \mathrm{Cov}(X_1, Y) \\ \vdots \\ \mathrm{Cov}(X_p, Y) \end{pmatrix} = \mathbf{V}_K \mathbf{\Lambda}_K \mathbf{\Theta}.$$

Therefore, the population ordinary least squares coefficients of regressing Y on $\mathbf{X}$ ($\boldsymbol{\beta}$ in (4)) can be written as

$$\boldsymbol{\beta} = \Sigma_{xx}^{-1} \Sigma_{xy} = \mathbf{V}_K \mathbf{L}_K^{-1} \mathbf{\Lambda}_K \Theta \qquad (6)$$

We will define the set $\mathrm{B} := \{j : (\Sigma_{xy})_j \neq 0\}$ and the set $\mathrm{A} := \{j : \beta_j \neq 0\}$. We note that for $K = 1$, it is always the case that $\mathrm{A} = \mathrm{B}$. By manipulating the parameters in $\mathbf{\Theta}$, $\mathbf{L}$, and $\mathbf{\Lambda}$, we can create a number of scenarios for testing AIMER against alternative methods.

3.2 Experiments

We present results under five different experiments. For each of the simulations which follow, we generate datasets with $n = 200$ and $p = 1000$. We use half ($n = 100$) to estimate the model and test our predictions on the other half. We repeat this process 100 times for each combination of parameters. Throughout, we use $\sigma_0 = \sqrt{.1} \approx .3$ and $\sigma_1 = .1$. The matrix $\mathbf{U}$ is generated with i.i.d. standard Gaussian entries, while the matrix $\mathbf{V}$ is constructed by hand to have the correct number of orthogonal components.

The first experiment is designed to be favorable to AIMER. The second is designed to be favorable to SPC. The third examines the extent to which the assumption that $\mathcal{A} = \mathcal{B}$ is beneficial to SPC over AIMER. The fourth examines the impact of using incorrect numbers of components, while the fifth uses cross validation on all the tuning parameters.

Simulation 1: Favorable conditions for AIMER.

In this simulation, we create data which is amenable to AIMER at the expense of the conditions for SPC, that is we use $\mathrm{B} \subset \mathrm{A}$. We set parameters in the data model as $K = G = 3$ and choose $\lambda_1 = 10$, $\lambda_2 = 5$, and $\lambda_3 = 1$. In order to achieve $\mathrm{B} \subset \mathrm{A}$, we set $\theta_1 = \theta_2 = 1$ and solve (5) for θ_3 so that some corresponding elements of Σ_{xy} will be zero. We make the first 15 elements of $\boldsymbol{\beta}$ non-zero, five corresponding to each of the three principal components. Thus, the first 10 genes have non-zero population marginal correlation and the remaining 990 have zero marginal correlation. In this scenario, SPC should find the first 10 important genes, but AIMER will find the remaining five important genes as well.

In order to focus on the relationship between performance and the condition $\mathrm{B} \subset \mathrm{A}$, we examine the methods for a fixed computational budget and choose t_* to select the same 50 most predictive genes. We examine SPC, SPC with lasso, AIMER($b = 0$), and AIMER. We use the first three principal components for regression in all the methods. For SPC with lasso and AIMER, we choose the remaining tuning parameters via 10-fold cross-validation. We also give results for OLS on the first 15 genes. This is the oracle estimator, the best one could hope to do with foreknowledge of the predictive genes.

Figure 2 shows the classification performance using a receiver operating characteristic (ROC) curve for SPC with lasso and AIMER in the left panel (the remaining panels are for the next two simulations). Examining the figure, it is easy to see that SPC + lasso identifies the first 10 genes easily, but AIMER is able to capture all 15 predictive genes at a low cost of false positive identifications. A more detailed analysis is given in the first row of Figure 3. Panel 1a shows the ability of each method to estimate the β coefficients of three different factors. Coefficient estimates for the five genes in

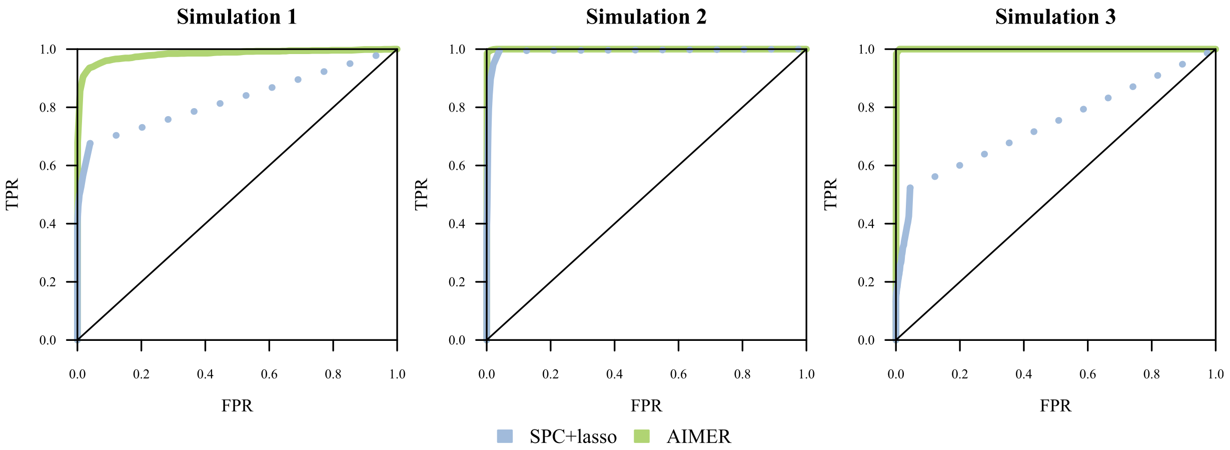

Fig. 2. Receiver operating characteristic (ROC) Curve for Simulations 1–3. The x-axis is the false positive rate while the y-axis is the true positive rate. The curves present averages across 100 replications. SPC is limited to only 50 selected genes, and so its false positive rate is bounded. The dashed line indicates its best case theoretical performance were it allowed to continue to select further genes

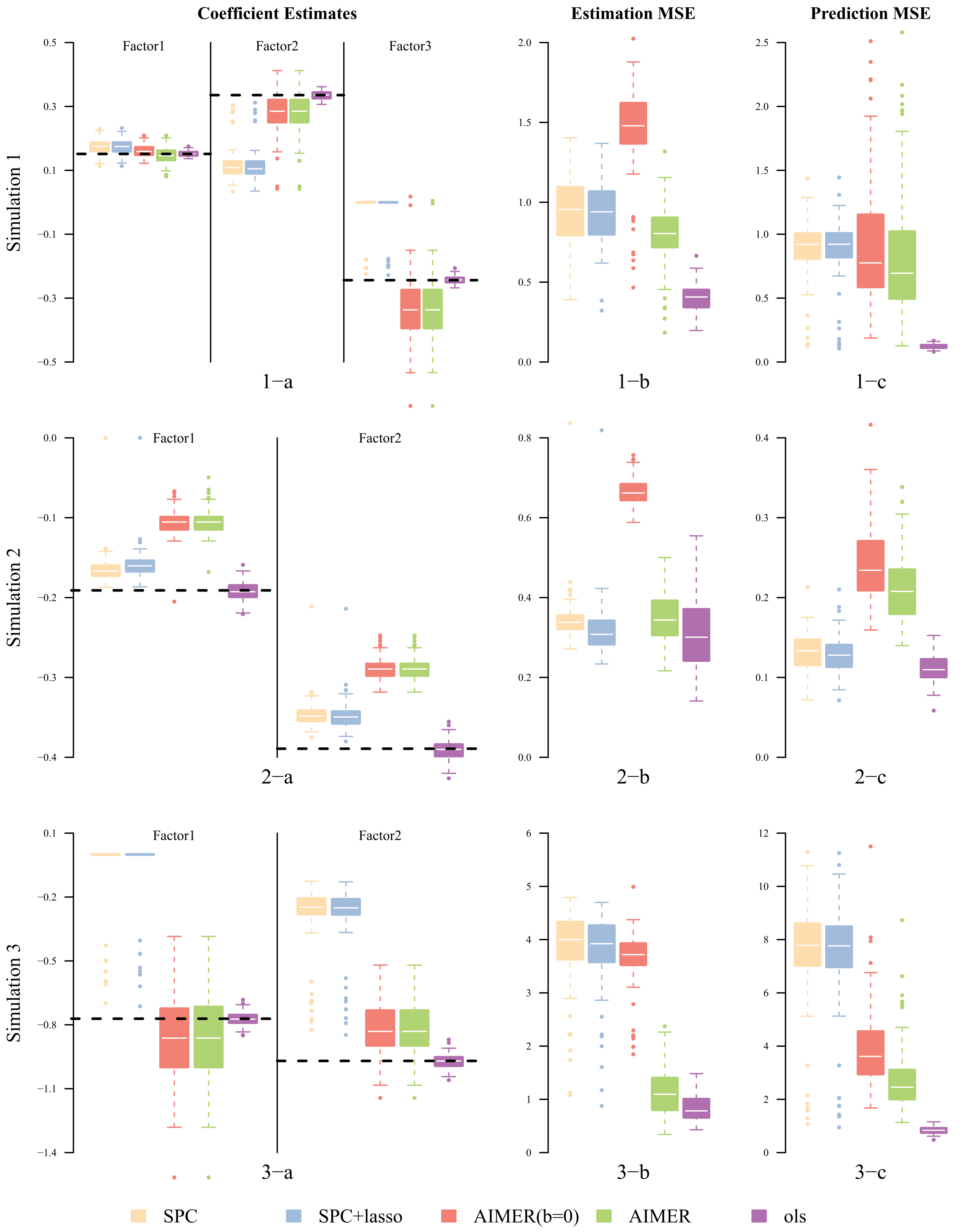

Fig. 3. Estimation and prediction performance of SPC and AIMER in the first three simulations. The left panel shows the estimates of the regression coefficients, the middle panel shows the mean squared error (MSE) of estimation for all 1000 genes, and the right panel shows prediction MSE on the held-out data. The boxes indicate variability across 100 replications. The dashed black horizontal lines indicate the true values of β

factor 1 by AIMER are slightly more accurate, and no more variable, than SPC + lasso. Furthermore, AIMER is better at estimating those β's associated with factor 2, and much better at those associated with factor 3 (these are assumed zero in SPC). Panel 1b examines the mean square error (MSE) of estimation as the average squared difference between the true coefficients and their estimates for all 1000 genes. The overall estimation accuracy of AIMER ($b = 0$) is worse because of the inclusion of so many useless genes (it estimates all 1000), however, by thresholding with AIMER, accuracy is improved and exceeds that of SPC with and without lasso. In panel 1c, we show the MSE for prediction, the average squared difference between predicted values and the actual observations, for a test set. This MSE is smaller for AIMER than for SPC much of the time, but the variance across simulations is large.

Simulation 2: Favorable conditions for SPC.

This simulation compares the performance of SPC and AIMER under conditions which are more favorable to SPC. In particular, we choose parameters such that A = B. While AIMER is likely to perform worse because it will tend to include irrelevant genes, it is not too much worse. Most of the parameters are the same as in Simulation 1, except that $K = G = 2$, $\lambda_1 = 10$, $\lambda_2 = 1$, $\theta_1 = \theta_2 = 1$, and we use the first two principal components to do regression. Therefore, 10 out of 1000 genes are truly predictive of the response, and all 10 have non-zero marginal correlation with the response (the rest have $\Sigma_{xy} = 0$). Looking again at Figure 2, both SPC + lasso and AIMER can identify all 10 predictive genes at a small price of false positives. Examining Figure 3, we see that the estimation accuracy of SPC/SPC + lasso is better than that of AIMER as expected, and the MSE of prediction for AIMER is about twice that of SPC/SPC + lasso. The estimation MSE (panel 2b) of AIMER is comparable to that of SPC.

Simulation 3: Slight perturbations.

In this simulation, we adjust only $\theta_2 = 3$, rather than 1 as in simulation 2, thereby maintaining the condition that A = B. However, in this case AIMER works much better than SPC/SPC + lasso. Figures 2 and 3 show that AIMER can easily identify all the predictive genes, has more precise coefficient estimates, and has much smaller MSE for prediction. The reason is that, even though A = B, the marginal correlations for some predictive genes are very small. Therefore, those genes are more difficult for SPC to identify, but AIMER can compensate.

Table 2. Average final number of predictive genes in Simulations 1, 2 and 3

Simulation	1	2	3
True #	15	10	10
SPC	50 (0)	50 (0)	50 (0)
SPC+lasso	31 (9.011)	39 (3.636)	46 (2.665)
AIMER ($b = 0$)	1000 (0)	1000 (0)	1000 (0)
AIMER	39 (9.225)	21 (12.750)	16 (7.558)

Note: The standard deviation is shown in parentheses.

For one further comparison, Table 2 shows the average (standard deviation in parentheses) number of predictive genes selected in each of the first three simulations. AIMER selects the smallest number of coefficients in most cases.

Simulation 4: Choosing the number of components.

In the previous simulations, we used the correct number of principal components, though such a choice is unlikely to be possible given real data. In this simulation, we examine the impact choosing the number of components has on estimation accuracy. We use similar parameter settings as Simulation 1 except with $K = G = 2$ rather than 3 (we maintain the condition that $\mathcal{B} \subset \mathcal{A}$). We then use all the methods with 1, 2 and 3 components. We also adjust the values of λ_1 in a range from 5 to 50. As we can see in Figure 4, using two components reduces MSE for AIMER($b = 0$) and AIMER across all values of λ_1 relative to using only one component, while using more than two components has little impact. With only one component, SPC performs better than AIMER, likely due to smaller variance for a similar bias, but using two or three components leads to large gains for AIMER. In practice, it is worthwhile to try several numbers of components and use cross-validation to decide which works best.

Simulation 5: The screening threshold.

In previous simulations, we choose t_* so that variable screening by the marginal correlation would always select exactly 50 genes. Thus, we could compare methods based on their ability to use the same amount of information. In reality, it may be better to choose the threshold t_* using cross validation. In this simulation, we use the same conditions as in the previous simulation with $\lambda_1 = 10$. It is still not appropriate to have more genes than patients, so we allow the number of selected genes to be anything less than the number of patients (100). We further use 10-fold cross-validation to choose the best threshold.

As shown in Figure 5, allowing t_* to be chosen rather than fixed leads to improved results for AIMER relative to SPC/SPC + lasso. The prediction MSE decreases and fewer genes are selected.

4 Performance on real data

We now illustrate our methods on four empirical datasets in genomics that record the censored survival time and gene expression measurements from DNA microarrays of patients with four different

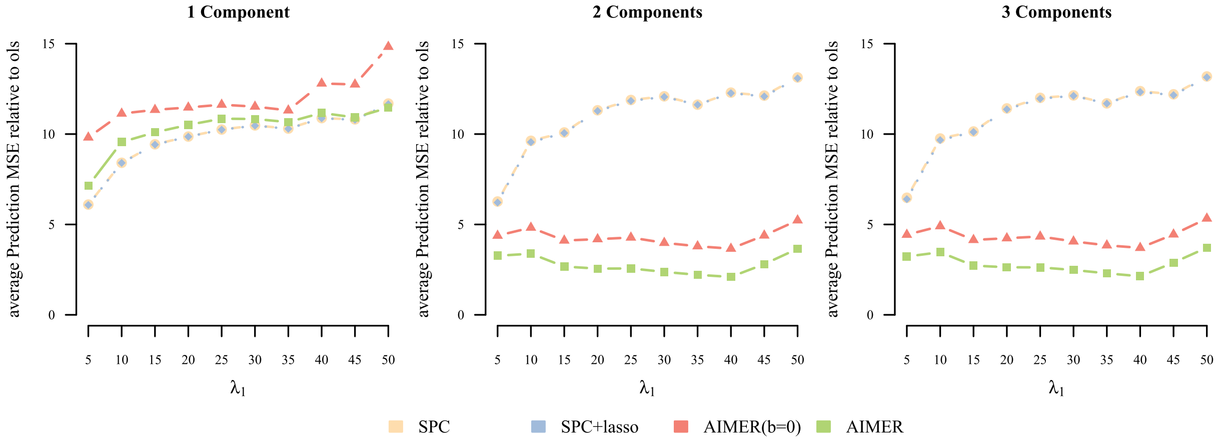

Fig. 4. Prediction MSE averaged across 100 replications for each method for different numbers of components (Simulation 4). We also allow λ_1 to vary between 5 and 50

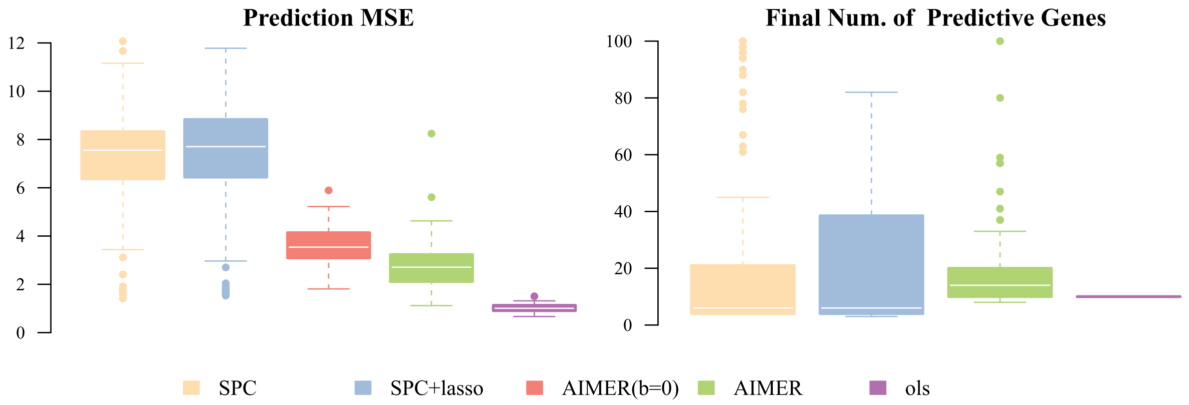

Fig. 5. Performance of each method when we allow t_* to be chosen by cross validation rather than fixed to choose 50 genes (Simulation 5)

Table 3. The MSE on the test set, the number of selected genes and the number of principal components used (*d* if relevant), each averaged across the 10 random training-testing splits

	DLBCL			Breast cancer			Lung cancer			AML		
Methods	MSE	# genes	*d*	MSE	# genes	*d*	MSE	# genes	*d*	MSE	# genes	*d*
lasso	0.6805	20		**0.6285**	9		0.8159	22		1.9564	6	
ridge	**0.6485**	7399		0.6407	4751		**0.7713**	7129		**1.9234**	6283	
SPC	0.6828	41	3	0.6066	16	2	**0.8344**	19	3	2.4214	24	2
SPC+lasso	0.6780	31	3	0.6029	14	2	0.8436	9	4	2.3980	22	2
AIMER($b = 0$)	1.1896	7399	2	2.6531	4751	1	0.9444	7129	1	12.4014	6283	1
AIMER	**0.6518**	28	4	**0.6004**	31	3	1.0203	13	1	**1.8746**	36	4

Note: Bolded values indicate the best predictive performance for each type of method (with and without structure learning) for each dataset.

types of cancer. The first dataset comes from Rosenwald *et al.* (2002) and contains 240 patients with diffuse large B-cell lymphoma (DLBCL) and 7399 genes. The second dataset has 4751 gene expression measurements of 78 breast cancer patients (Van't Veer *et al.*, 2002). The third consists of 86 lung cancer patients measured on 7129 genes (Beer *et al.*, 2002), and finally, we analyze a dataset consisting of 116 patients with acute myeloid leukemia (AML, Bullinger *et al.*, 2004) and 6283 genes.

Since the survival times for some patients are censored and right-skewed, we use log (survival time + 1) as the response. A Cox model would be more appropriate, but this transformation is enough to illustrate our methodology. In order to assess our method using limited data, we randomly select half of the data as the training set and let the rest be in the testing set, then estimate each model using the training half and predict the held out data. We repeat this procedure for 10 random splits and report the average error. We use 10-fold cross-validation on the training set to choose all tuning parameters (t_*, b_*, d and λ where appropriate), mimicking the procedure of a real data analysis.

We apply seven methods on each dataset: (i) PCR; (ii) lasso; (iii) ridge regression; (iv) SPC; (v) SPC + lasso; (vi) AIMER($b = 0$) and (vii) AIMER. We use the R packages pls (Mevik and Wehrens, 2007) to perform PCR and glmnet (Friedman *et al.*, 2010) to perform lasso and ridge. For PCR, SPC, SPC + lasso, AIMER($b = 0$) and AIMER, we allow the number of components d to be chosen between 1 and 5.

Our results are shown in Table 3. For each dataset, we show the MSE on the testing set, the number of selected genes, and the number of principal components used (if relevant), averaged across the 10 random training-testing splits. We do not show results for PCR because it is uniformly awful. The results in Table 3 are largely consistent with the conclusions we derive from simulations. AIMER and SPC + lasso tend to select a similar number of genes, though AIMER has better prediction error on three of the four datasets. Interestingly, the genes selected by SPC + lasso, lasso and AIMER rarely overlap, suggesting that to identify genes for further study, one should try all three methods. The online Supplementary Material lists the genes identified by AIMER for each dataset. In the case of DLBCL, we also list any previous research relating the selected genes to lymphoma.

The Lung Cancer data is rather odd in that AIMER($b = 0$) has better performance than AIMER. This anomaly is likely because, in contrast with the other datasets, the lung cancer expression measurements have not been scaled relative to a control group. We tried two transformations using only the treatment group to approximate such a scaling, but, while the performance of our method becomes comparable to SPC following transformations, it remains slightly worse. Without a control group, it is difficult to explain this outcome with any certainty. A comparison of these alternative transformations with our results in Table 3 is contained in the online Supplementary Material.

As seen in the table, ridge regression is sometimes the best of all the methods. Previous experience suggests that ridge regression is dominant if the genes are highly correlated or when there is not a particularly predictive set of genes. However, the fact that ridge does not screen out unimportant genes is a barrier to its applications in genomics. On the other hand, AIMER approaches or exceeds the small prediction error of ridge regression while also selecting a small number of predictive genes, making it a better candidate for solving these types of problems.

5 Discussion

High-dimensional regression methods help in predicting future survival time and identifying possibly predictive genes for diseases. However, the large number of genes, the limited access to patients, and the complex covariance structure between genes make the problem both computationally and statistically difficult. In both simulations and analysis of actual gene expression datasets, AIMER has comparable or slightly improved prediction accuracy relative to existing methods and finds small numbers of actually predictive genes, all while having a similar computational burden. On the other hand, there are some issues which warrant further exploration.

A major benefit of SPC is that it comes with theoretical guarantees under certain assumptions. While our methodology is intended to work when these assumptions don't hold, we do not yet have comparable guarantees. However, the simulated experiments in this paper have suggested how we might derive such results in a more general setting.

For the real data examples in this paper, we applied a simple monotonic transformation to the response variable, however, extending our methods to Cox models, which are more appropriate, and other generalized linear models for predicting discrete traits is highly desirable. It may also be useful to examine other eigenstructure techniques such as Locally Linear Embeddings or Laplacian Eigenmaps to produce non-linear predictors. Finally, using other matrix approximation techniques may yield improved performance or be more amenable to theoretical analysis.

Funding

This work is supported by the National Science Foundation [grant number DMS–14-07439 to D.J.M.].

Conflict of Interest: none declared.

References

Alter,O. *et al.* (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA*, **97**, 10101–10106.

Bair,E. *et al.* (2006) Prediction by supervised principal components. *J. Am. Stat. Assoc.*, **101**, 119–137.

Bair,E., and Tibshirani,R. (2004) Semi-supervised methods to predict patient survival from gene expression data. *PLoS Biol.*, **2**, e108.

Barrett,J.C. *et al.* (2008) Genome-wide association defines more than 30 distinct susceptibility loci for crohn's disease. *Nat. Genet.*, **40**, 955–962.

Beer,D.G. *et al.* (2002) Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nat. Med.*, **8**, 816–824.

Bullinger,L. *et al.* (2004) Gene expression profiling identifies new subclasses and improves outcome prediction in adult myeloid leukemia. *New Engl. J. Med.*, **350**, 1605–1616.

Burton,P.R. *et al.* (2007) Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, **447**, 661–678.

Candes,E.J., and Tao,T. (2007) The Dantzig selector: statistical estimation when p is much larger than n. *Ann. Stat.*, **35**, 2313–2351.

Elks,C.E. *et al.* (2010) Thirty new loci for age at menarche identified by a meta-analysis of genome-wide association studies. *Nat. Genet.*, **42**, 1077–1085.

Friedman,J. *et al.* (2008) Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**, 432–441.

Friedman,J. *et al.* (2010) Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.*, **33**, 1.

Hastie,T. *et al.* (2001) Supervised harvesting of expression trees. *Genome Biol.*, **2**, research0003–research0001.

Hastie,T. *et al.* (2000) Identifying distinct sets of genes with similar expression patterns via "gene shaving". *Genome Biol.*, **1**, 1–21.

Hoerl,A.E., and Kennard,R.W. (1970) Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, **12**, 55–67.

Homrighausen,D., and McDonald,D.J. (2016) On the Nyström and column-sampling methods for the approximate principal components analysis of large data sets. *J. Comput. Graph. Stat.*, **25**, 344–362.

Hotelling,H. (1957) The relations of the newer multivariate statistical methods to factor analysis. *Br. J. Stat. Psychol.*, **10**, 69–79.

Hromatka,B.S. *et al.* (2015) Genetic variants associated with motion sickness point to roles for inner ear development, neurological processes and glucose homeostasis. *Hum. Mol. Genet.*, **24**, 2700–2708.

Johnstone,I.M., and Lu,A.Y. (2009) On consistency and sparsity for principal components analysis in high dimensions. *J. Am. Stat. Assoc.*, **104**, 682–693.

Jolliffe,I.T. (2002) *Principal Component Analysis*. Springer, New York.

Kendall,M.G. (1965) *A Course in Multivariate Analysis*. Charles Griffin & Co., London.

Kennedy,R.B. *et al.* (2012) Genome-wide analysis of polymorphisms associated with cytokine responses in smallpox vaccine recipients. *Hum. Genet.*, **131**, 1403–1421.

Lesage,S., and Brice,A. (2009) Parkinson's disease: from monogenic forms to genetic susceptibility factors. *Hum. Mol. Genet.*, **18**, R48–R59.

Lu,A.Y. (2002). Sparse principal component analysis for functional data. PhD Thesis, Stanford University, Stanford, CA.

Meinshausen,N., and Bühlmann,P. (2006) High-dimensional graphs and variable selection with the lasso. *Ann. Stat.*, **34**, 1436–1462.

Mevik,B.H., and Wehrens,R. (2007) The pls package: principal component and partial least squares regression in r. *J. Stat. Softw.*, **18**, 1–23.

Paul,D. *et al.* (2008) 'Preconditioning' for feature selection and regression in high-dimensional problems. *Ann. Stat.*, **36**, 1595–1618.

Pearson,K. (1901) Principal components analysis. *Lond. Edinb. Dublin Philos. Mag. J.*, **6**, 566.

Perry,J.R.B. *et al.* (2014) Parent-of-origin-specific allelic associations among 106 genomic loci for age at menarche. *Nature*, **514**, 92–97.

Rosenwald,A. *et al.* (2002) The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *New Engl. J. Med.*, **346**, 1937–1947.

Saito,T. *et al.* (2016) Pharmacogenomic study of clozapine-induced agranulocytosis/granulocytopenia in a Japanese population. *Biol. Psychiatry*, **80**, 636–642.

Sladek,R. *et al.* (2007) A genome-wide association study identifies novel risk loci for type 2 diabetes. *Nature*, **445**, 881–885.

Tibshirani,R. (1996) Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc. B*, **58**, 267–288.

Van't Veer,L.J. *et al.* (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

Wall,M.E. *et al.* (2003) Singular value decomposition and principal component analysis. In: *A Practical Approach to Microarray Data Analysis*. Springer, New York, p.91–109.

Yuan,M., and Lin,Y. (2006) Model selection and estimation in regression with grouped variables. *J. Roy. Stat. Soc. B*, **68**, 49–67.