Detection of Low Rank Signals in Noise and Fast Correlation Mining with Applications to Large Biological Data

Andrey A. Shabalin

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research (Statistics).

Chapel Hill
2010

Approved by

Andrew B. Nobel, advisor

Amarjit Budhiraja, reader

Yufeng Liu, reader

J. S. Marron, reader

Charles M. Perou, reader

Ivan Rusyn, reader

ABSTRACT

ANDREY A. SHABALIN

Detection of Low Rank Signals in Noise and Fast Correlation Mining
with Applications to Large Biological Data

(Under the direction of Andrew Nobel)

Ongoing technological advances in high-throughput measurement have given biomedical researchers access to a wealth of genomic information. The increasing size and dimensionality of the resulting data sets requires new modes of analysis. In this thesis we propose, analyze and validate several new methods for the analysis of biomedical data. We seek methods that are at once biologically relevant, computationally efficient, and statistically sound.

The thesis is composed of two parts. The first concerns the problem of reconstructing a low-rank signal matrix observed in the presence of noise. In Chapter 1 we consider the general reconstruction problem, with no restrictions on the low-rank signal. We establish a connection with the singular value decomposition. This connection and recent results in random matrix theory are used to develop a new denoising scheme that outperforms existing methods on a wide range of simulated matrices.

Chapter 2 is devoted to a data mining tool that searches for low-rank signals equal to a sum of raised submatrices. The method, called LAS, searches for large average submatrices, also called biclusters, using an iterative search procedure that seeks to maximize a statistically motivated score function. We perform extensive validation of LAS and other biclustering methods on real datasets and assess the biological relevance of their findings

The second part of the thesis considers the joint analysis of two biological datasets. In Chapter 3 we address the problem of finding associations between single nucleotide polymorphisms (SNPs) and genes expression. The huge number of possible associations requires careful attention to issues of computational efficiency and multiple comparisons. We propose a new method, called FastMap, that exploits the discreteness of SNPs, and uses a permutation approach to account for multiple comparisons.

In Chapter 4 we describe a method for combining gene expression data produced from different measurement platforms. The method, called XPN, estimates and removes the systematic differences between datasets by fitting a simple block-linear model to the available data. The method is validated on real gene expression data.

The methods described in Chapters 2-4 have been implemented and are publicly available online.

ACKNOWLEDGEMENTS

I appreciate the enduring patience of my adviser Andrew Nobel, without him this dissertation would be similar to the paper of Upper (1974).

This work was supported, in part, by grants from Institutes of Health [grant numbers P42-ES005948 and R01-AA016258], National Science Foundation [grant numbers DMS-0406361 and DMS-0907177] National Cancer Institute Breast SPORE program to University of North Carolina at Chapel Hill [grant number P50-CA58223-09A1] National Cancer Institute [grant number RO1-CA-101227-01], and United States Environmental Protection Agency [grant numbers RD832720 and RD833825].

CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Rapid technological progress in the last decades has allowed biologists to produce increasingly large data sets of various types. The first and most popular type is gene expression microarrays, which became popular back in 1990's. Other technologies have developed to measure micro-RNA expression and copy number variation, to detect single nucleotide polymorphisms (SNPs), and even to perform full genome sequencing. The data sets produced by such technologies can often be represented as matrices of measurements, where each column corresponds to a sample, and each row corresponds to a measured variable. Currently, large data sets can have from tens of thousands (for gene expression arrays) to millions of variables (for SNP arrays). The number of samples in the data sets can range from tens to thousands, the latter when the efforts of multiple research centers is combined (see *TCGA: The Cancer Genome Atlas*). In most cased the resulting data sets are real-valued. Although next generation sequencing arrays generate integer values (counts), they can often be treated as real-valued. The clear exception is SNP arrays, which contain binary values for inbred homozygous populations, and ternary values for heterozygous populations like humans.

The analysis of biological data sets aims to reveal add information to our existing knowledge about human diseases such as cancer or cystic fibrosis. For instance, breast cancer is now known to be not one, but a family, of diseases that differ in speed of tumor growth, response to treatments, likelihood of metastases, and likelihood of relapse after surgical removal of a tumor. Gene expression technology enabled biologists to discover subtypes of breast and other types of cancer. SNP data can be used to determine how differences in genotype predispose people to different diseases and types of diseases. Genotypic differences between individuals can be associated with various phenotypes, including phenotypes derived from clinical variables and gene expression. Analysis of new and existing genetic data can lead to more effective treatments

targeting individuals genetic makeup.

Typical gene expression data set has tens of thousands of variables and hundreds of samples. SNP arrays have millions of variables. Even preliminary analysis of such large datasets is complicated by their size. While a simple visual inspection of gene expression data is now common, it is not practical for SNP data. Preliminary analysis by simply looking at the table of numbers is not possible even for moderately sized data sets, as they contain millions of measurements. The common approach to visualizing gene expression data is the following. The rows and columns of the data matrix are hierarchically clustered, and then reordered so that the clusters contiguous. Next, a heatmap is produced. In heatmap each measurement is represented by just one pixel, colored green for negative values, red for positive, with brightness proportional to the absolute value of the measurement. SNP data sets are usually not visualized in this way, as they have too many variables. Instead, scientists perform an analysis first, and then visualize the results of the analysis in some way. For example, one can evaluate association of each SNP with a response variable, like survival, and plot the association statistics for SNPs in the region near the SNP with the largest association.

Analysis of large biological datasets must be both computationally efficient and statistically principle. For instance, if a method is statistically motivated, but has complexity proportional to the cube of the number of variables, it would be impractical or even unfeasible for many data sets. Such methods may perform well when tested on data sets with 500 to 1000 variables, but they would not scale well to modern data sets with tens to hundreds of thousands of variables. On the other hand, some existing methods for the analysis of biological data sets are computationally efficient, but lack statistical justification. For instance, some data mining methods are designed to find all patterns in a given data set that satisfy a certain criterion, regardless of how likely it is to find such patterns in a matrix of pure noise. For this reason such methods may produce a large output with many spurious findings. Others may have some statistical motivation, but fail to account for multiple comparisons. Some data mining methods assess significance of their findings by calculating the probability of such exact pattern appearing in a random data matrix. However, there is usually a great number of patterns the method considers and assesses significance for, so the probability of finding a pattern with a low p-value in a pure noise matrix can actually be quite high.

Another, less widely recognized problem of some methods lies in the number of parameters they have. The method can be hard to apply if the parameters have to be individually hand-picked for each dataset. Moreover, if a generalization of an existing method is proposed, which adds more parameters, one can always choose the parameters for the new method to outperform the old method on any given test. However, this does not indicate that the new method would be better in practice, when the choice of all the parameters would become a problem, not an advantage.

Last but not least, methods that do not have the drawbacks listed above sometimes lack validation on real data. For example, methods for mining biological data sets, may search for particular features of the data in both computationally efficient and statistically sound fashion, but not be actually useful for a biologist.

In this dissertation we propose several new statistical methods for analysis of biological data sets, each computationally efficient, statistically principal, and validated on real data. In Section 2 we propose a new biclustering method, called LAS. The LAS section contain a revised version of the paper published in Annals of Applied Statistics (Shabalin, Weigman, Perou & Nobel 2009). Next, in Section 3 we present a new method for fast eQTL analysis, called FastMap. The FastMap section contain a revised and extended version of the paper published in Bioinformatics (Gatti, Shabalin, Lam, Wright, Rusyn & Nobel 2009). In Section 4 we present a new method for cross platform normalization of gene expression arrays, called XPN. The XPN section contain a revised version of the paper published in Bioinformatics (Shabalin, Tjelmeland, Fan, Perou & Nobel 2008). The dissertation begins with Chapter 1 which presents our most recent research and most theoretical research. It studies the problem of recovery of denoising of low-rank matrices. This research has not yet been submitted to any journal.

Outline

In Chapter 1 we present a new method for denoising of low-rank matrices with additive Gaussian noise. The denoising problem is usually solved by singular value decomposition of the observed data matrix followed by shrinkage or thresholding of its singular values. Although a wide family of reconstruction schemes is possible, we proof that under minor conditions an efficient

reconstruction scheme must indeed be based on the singular value of the observed matrix. Even more, it should only change the singular value of the matrix, leaving the singular vectors intact. However, as we determine in latter in the Chapter, it is not efficient to restrict the scheme to simple shrinkage and/or thresholding of singular values.

Next, applying random matrix theory we study the effect of noise on low rank matrices, namely on their singular values and singular vectors. Then we construct the proposed denoising scheme based on this knowledge.

Simulation study with a wide range of settings shows that the proposed reconstruction scheme strongly outperforms the conventional ones regardless of the choice of shrinking and thresholding parameters for them. The performanceof the proposed method nearly matches the performance of the general oracle denoising scheme.

As a side result we determine the minimum strength (singular value) the signal must have to be at least partially recoverable.

In Chapter 2 we present a new data mining method called LAS. It was inspired by the process of visual mining of heatmap data representations. Some biologists visually inspect heatmaps of gene expression data in search of solid red or green blocks. Such blocks represent sample-variable interactions, sets of gene that are simultaneously active or inactive for corresponding sets of samples. LAS approach a more general problem of finding submatrices with large positive or negative average. Such submatrices do not have to be contiguous in any heatmap representation of the data. To search for such submatrices, called biclusters, we first assign each submatrix a score, which is larger for larger and brighter submatrices. The score is defined as negative logarithm of p-value, calculated for the null model that data has only noise and Bonferroni corrected for the number of submatrices of given size. Biclusters with larger score are found first. Once a bicluster is found, it is removed by reducing its elements, and the search continues for the next ones.

The task of finding a bicluster with largest score is NP-complete, so we use a heuristic algorithm for the search. Our simulations have shown that LAS algorithm is most often successful in finding raised submatrices in simulated data.

In the chapter we also perform an extensive validation of LAS biclustering. We applied LAS

to breast and lung cancer data sets. We found at least one bicluster for each known cancer subtype those sample set closely matches the samples of the subtype. We have also tested the biclusters' sample sets for association with clinical variables and gene sets for overrepresentation of know gene categories. In all tests LAS outperforms other known biclustering methods.

The LAS problem can be seen as a special case of signal detection problem. LAS model assumes the signal in the observed data to be a sum of matrices, each equal to a fixed number on a submatrix and zero elsewhere. Note that the signal matrix with B biclusters has rank at most B, so the LAS model is a particular case of low-rank signal detection model considered in Chapter 1. However, as we show in Section 2.6.1, LAS algorithm can find submatrices that are not detectable with SVD of the data matrix.

Although the analysis of individual datasets has proven to be useful, more information can be discovered by joint analysis of two, or more datasets. A particular case of such analysis is gene expression quantitative trait loci (eQTL) mapping, which searches for associations between SNPs and genes. The number of associations to be calculated is equal to the product of the number of SNPs (>1m) and the number of genes measured (∼40k), which can be in the order of tens of billions. The computational burden is even greater if the researcher chooses to perform permutation analysis to assess the significance of their findings. To address the computational issues while keeping the analysis statistically correct, we propose a new computational method for eQTL analysis, called FastMap. The method exploits the discrete structure of SNP data (whether binary or ternary) to greatly improve the speed of the eQTL analysis. FastMap performs analysis on gene by gene basis. The significance of the strongest association of a given gene expression with the available SNPs is assessed using a permutation approach. By assessing the significance of the strongest association over all SNPs we avoid multiple comparison issue across SNPs. In order to address multiple comparisons across genes, FastMap assigns a q-value (Storey & Tibshirani 2003) assessing false discovery rate to each gene.

The original FastMap program, as published in Gatti et al. (2009), was designed to work with homozygous SNP data only. Since then, the program has been improved, it now supports ternary SNP data, and works faster on datasets with many samples (>50).

Collaboration of different cancer centers allows biologist to combine data in order to gain better strength in the subsequent analysis. More and more datasets become publicly available with time. However one can not simply join datasets from different sources. The data produced by different research centers differs more than the data produced within one center. This difference can arise from a variety of reasons. First, different studies may use gene expression arrays from different manufacturers, or different versions of arrays from the same manufacturer. Second, even measurements from various batches of samples from the same array can differ more than the measurements within each batch. Differences across batches occur because of differences in measuring conditions, including different batches of arrays, batches of reagents, and different versions of processing software. Differences across platforms occur because the probes of different arrays may target different sequences from the the same genes, which are often located in different exons.

To remove batch and/or platform effects across gene expression datasets we propose a new method for cross platform normalization, called XPN. It is based on a block model of the data. XPN is distinguished from other platform normalization methods that are gene-wise linear. In Chapter 4 we describe the method and carefully validate it on several real data. The tests on real data show that XPN is more successful in removing platform effects while preserving important biological information.

Software

For the LAS, FastMap, and XPN methods presented in this dissertation we provide free implementations of the methods.

The LAS program is implemented in C# programming language with an intuitive graphics user interface (see Figure 1, left). An alternative implementation in Matlab is also available for those who may want to add modifications to the code and for cross-platform compatibility. The LAS software is available at https://genome.unc.edu/las/.

The FastMap method is implemented in Java. It also has an easy to navigate graphical user interface (see Figure 1, right). The FastMap program and source code are available at http://cebc.unc.edu/fastmap86.html.

The XPN method is implemented in Matlab and is available at https://genome.unc.edu/

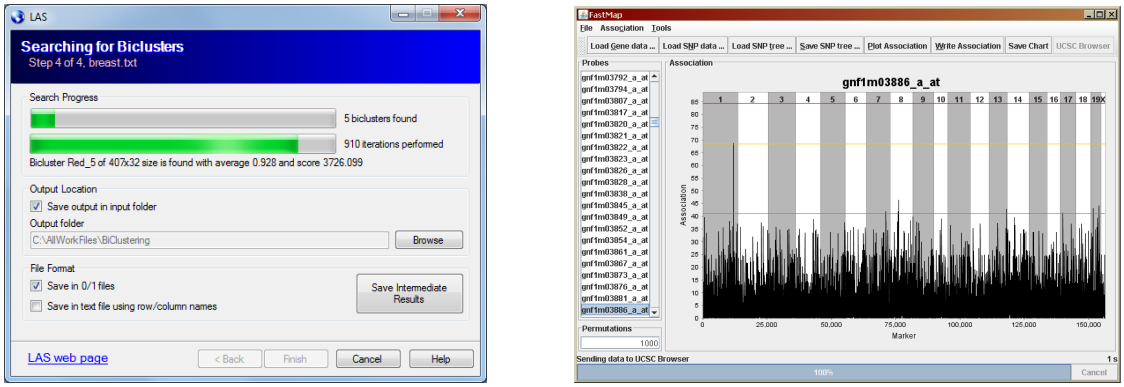

Figure 1: LAS program user interface on left and FastMap on right.

xpn.

CHAPTER 1

Reconstruction of a Low-rank Matrix in the Presence of Gaussian Noise

1.1 Introduction

This chapter addresses the problem of recovering a low rank matrix whose entries are observed in the presence of additive Gaussian noise.

Problems of this sort appear in multiple fields of study including compressed sensing and image denoising. In many of these cases the signal matrix is known to have low rank. For example, a matrix of squared distances between points in d-dimensional Euclidean space is know to have rank at most $d + 2$. A correlation matrix for a set of points in d-dimensional Euclidean space has rank at most d. In other cases the target matrix is often assumed to have low rank , or to have a good low-rank approximation. For example, Alter et al. (2000), Holter et al. (2000) and Raychaudhuri et al. (2000) assumed the signal component of gene expression matrices to have low rank.

The reconstruction problem considered here has a signal plus noise structure. Our goal is to recover an unknown $m \times n$ matrix A of low rank that is observed in the presence of i.i.d. Gaussian noise as matrix Y:

$$Y = A + \frac{\sigma}{\sqrt{n}} W, \qquad \text{where } W_{ij} \sim \text{ i.i.d. } N(0,1).$$

In what follows, we first consider the variance of the noise σ^2 to be known and assume it to be equal to one. Next, in Section 1.6.1 we propose an estimator for σ, which we use in the proposed reconstruction scheme.

The classical approach to denoising begins with the singular value decomposition of the

observed matrix Y. Then, the largest singular values are visually inspected on a scree plot. A sample scree plot is shown in Figure 1.1 below. The rank R of the signal is then estimated as the number of singular values to the left of the 'elbow' point. The scree plot on Figure 1.1 clearly indicates a rank-2 signal. The signal is then estimated as the sum of first R terms in the singular value decomposition of Y.

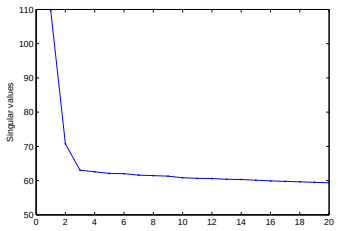

Figure 1.1: Scree plot for a 1000×1000 rank 2 signal matrix with noise.

A more formal version of this approach is known as hard thresholding. Hard thresholding estimates the signal matrix A by $\arg\min_{B:\ \mathrm{rank}(B)=R} \|Y - B\|_F^2$ for some data-driven choice of rank R. Hard thresholding preserves first R singular values of the matrix Y, and sets the rest to zero. Hard thresholding can be viewed equivalently as a minimization problem with rank-based penalty

$$\arg\min_B \left\{ \|Y - B\|_F^2 + \lambda^2 \mathrm{rank}(B) \right\},$$

for some parameter λ. Here and in what follows $\|\cdot\|_F$ denote Frobenius norm of a matrix.

Another approach is to shrink, not threshold the singular values of the observed matrix. It reduces all singular values by a constant λ and sets all singular values smaller that λ equal to zero. Such approach is called soft thresholding and it can also be formulated as minimization problem: $\arg\min_B \|Y - B\|_F^2 + 2\lambda\|B\|_*$, where λ is a parameter and $\|\cdot\|_*$ is matrix nuclear norm (sum of singular values).

Both hard and soft thresholding methods are studied in the literature and various rules for selection of the penalization parameters are proposed, some with performance guaranties under certain conditions. Both these approaches are based on SVD of the data and are popular in practical applications. For instance, Wall et al. (2001), Alter et al. (2000), and Holter et al. (2000) used SVD as a tool for mining gene expression data and Troyanskaya et al. (2001*a*)

applied SVD to impute missing values. SVD is also used for image denoising. However a better performance is achieved when SVD is applied to small blocks of pixels, not to the whole image. Denoising methods of Wongsawat et al. and Konstantinides et al. (1997) perform SVD on square subblocks and set to zero the singular values smaller than some threshold.

However, it is natural to ask whether SVD-based approach is optimal. In general, a reconstruction scheme is a map $g : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$. It does not have to be based on SVD of matrix Y and does not have to be formulated as a penalized minimization problem. It even does not have to produce matrices of low rank. Can we achieve better reconstruction if we do not restrict ourselves to scheme that just shrink or threshold singular values? Can we achieve a better reconstruction if we do not restrict ourselves to the method based on SVD and which produce low-rank matrices?

In the first part of this chapter we analyze the matrix reconstruction problem and determine several necessary properties of efficient reconstruction schemes. In Section 1.2 we prove that under mild conditions on the prior information about the signal (lack of information about its singular vectors) any effective denoising scheme must be based on the singular value decomposition (SVD) of the observed matrix. Moreover, it need only modify the singular values, not singular vectors. These facts alone reduce the space of efficient reconstruction schemes from $g : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ to just $g : \mathbb{R}^{m\wedge n} \to \mathbb{R}^{m\wedge n}$, where $m \wedge n$ denotes the minimum of m and n.

In the second part of the chapter we propose a new reconstruction scheme. Rather than adopting approaches of hard and soft thresholding, we start by determining the effect of additive noise on the singular values and singular vectors of low-rank matrices. We do it by first making a connection between the matrix reconstruction problem and spiked population models in random matrix theory. In Section 1.5 we translate relevant theorems from random matrix theory to the settings of the matrix reconstruction problem. The proposed reconstruction scheme is then derived from these results in Section 1.6. The proposed scheme is designed to reverse the effect of the noise on the singular values of the signal and corrects for the effect of the noise on the signal's singular vectors. We call the proposed method RMT for on its use of random matrix theory.

In Section 1.7 we compare the proposed method with oracle version of the soft and hard thresholding methods. The simulations show that RMT scheme strongly outperforms oracle

versions of the existing methods, and closely matches the performance of a general oracle scheme for generated matrices of various size and signal spectra.

1.2 The Matrix Reconstruction Problem

1.2.1 Statement of the Problem

The reconstruction problem considered here has a signal plus noise structure common to many estimation problems in statistics and signal processing. Our goal is to recover an unknown $m \times n$ matrix A of low rank that is observed in the presence of Gaussian noise. Specifically, we consider the matrix additive model

$$Y \;=\; A + \frac{\sigma}{\sqrt{n}} W \tag{1.1}$$

where Y denotes the observed matrix, $\sigma > 0$ is an unknown variance parameter, and W is a Gaussian random matrix with independent $N(0,1)$ entries. The matrices Y, A, and W are each of dimension $m \times n$. The factor $n^{-1/2}$ ensures that the signal and noise are comparable, and is essential for the for the asymptotic study of matrix reconstruction in Section 1.5, but it does not play a critical role in the characterization of orthoginally invariant schemes that follows.

At the outset we will assume that the variance σ is known and equal to one. In this case the model (1.1) simplifies to

$$Y \;=\; A + \frac{1}{\sqrt{n}} W, \quad W_{ij} \text{ independent} \sim N(0,1). \tag{1.2}$$

Formally, a matrix recovery scheme is a map $g : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ from the space of $m \times n$ matrices to itself. Given a recovery scheme $g(\cdot)$ and an observed matrix Y from the model (1.2), we regard $\widehat{A} = g(Y)$ as an estimate of A. Recall that the squared Frobenius norm of an $m \times n$ matrix $B = \{b_{ij}\}$ is given by

$$\|B\|_F^2 \;=\; \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij}^2.$$

If the vector space $\mathbb{R}^{m \times n}$ is equipped with the inner product $\langle A, B \rangle = \mathrm{tr}(A'B)$, then it is easy to see that $\|B\|_F^2 = \langle B, B \rangle$. We measure the performance of an estimate $\widehat{A}$ by the squared

Frobenius norm

$$\text{Loss}(A, \widehat{A}) \;=\; \|\widehat{A} - A\|_F^2. \tag{1.3}$$

Remark: Our assumption that the entries of the noise matrix W are Gaussian arises from two conditions required in the analysis that follows. The results of Section 1.3 require that W has an orthogonally invariant distribution (see Definition 3). On the other hand, the results of Section 1.5 are based on theorems from random matrix theory, which require the elements of W to be i.i.d. with zero mean, unit variance, and finite forth moment. It is known that the only distribution satisfying both these assumptions is the Gaussian (Bartlett 1934). Nevertheless, our simulations (not presented) show that the Gaussian noise assumption is not required to ensure good performance of the RMT reconstruction method.

In the next section we will provide some insights to the structure of the reconstruction problem.

1.3 Invariant Reconstruction Schemes

The additive model (1.2) and Frobenius loss (1.3) have several elementary invariance properties, which lead naturally to the consideration of reconstruction methods with analogous forms of invariance. Recall that a square matrix U is said to be orthogonal if $UU' = U'U = I$, or equivalently, if the rows (or columns) of U are orthonormal. If we multiply each side of (1.2) from the left right by orthogonal matrices U and V' of appropriate dimensions, we obtain

$$UYV' \;=\; UAV' + \frac{1}{\sqrt{n}} UWV'. \tag{1.4}$$

Proposition 1. *Equation (1.4) is a reconstruction problem of the form (1.2) with signal UAV' and observed matrix UYV'. If $\widehat{A}$ is an estimate of A in model (1.2), then $U\widehat{A}V'$ is an estimate of UAV' in model (1.4) with the same loss.*

Proof. If A has rank r then UAV' also has rank r. To prove the first statement, it remains to show that UWV' in (1.4) has independent $N(0,1)$ entries, which follows from standard properties of the multivariate normal distribution. In order to establish the second statement,

let U and V be the orthogonal matrices in (1.4). For any $m \times n$ matrix B,

$$\|UB\|_F^2 \;=\; \mathrm{tr}\big[(UB)'(UB)\big] \;=\; \mathrm{tr}\big[B'B\big] \;=\; \|B\|_F^2,$$

and more generally $\|UBV'\|_F^2 = \|B\|_F^2$. Applying the last equality to $B = \widehat{A} - A$ yields

$$\mathrm{Loss}(UAV', U\widehat{A}V') \;=\; \|U(\widehat{A} - A)V'\|_F^2 \;=\; \|\widehat{A} - A\|_F^2 \;=\; \mathrm{Loss}(A, \widehat{A})$$

as desired. □

In light of Proposition 1 it is natural to consider reconstruction schemes that are invariant under orthogonal transformations of the observed matrix Y.

Definition 2. *A reconstruction scheme $g(\cdot)$ is orthogonally invariant if for any $m \times n$ matrix Y, and any orthogonal matrices U and V of appropriate size, $g(UYV') = Ug(Y)V'$.*

In general, a good reconstruction method need not be orthogonally invariant. For example, if the target matrix A is known to be diagonal, then for each Y the estimate $g(Y)$ should be diagonal as well, and in this case $g(\cdot)$ is not orthogonally invariant. However, as we show in the next theorem, if we have no information about the singular vectors of A (either prior or from the singular values of A), then it suffices to restrict our attention to orthogonally invariant reconstruction schemes.

Definition 3. *A random $m \times n$ matrix Z has an orthogonally invariant distribution if for any orthogonal matrices U and V of appropriate size the distribution of UZV' is the same as the distribution of Z.*

As noted above, a matrix with independent $N(0,1)$ entries has an orthogonally invariant distribution. If Z has an orthogonally invariant distibution, then its matrix of left (right) singular vectors is uniformly distributed on the space of $m \times m$ ($n \times n$) orthogonal matrices.

Theorem 4. *Let $Y = \mathbf{A} + W$, where $\mathbf{A}$ is a random target matrix. Assume that $\mathbf{A}$ and W are independent and have orthogonally invariant distributions. Then, for every reconstruction scheme $g(\cdot)$, there is an orthogonally invariant reconstruction scheme $\tilde{g}(\cdot)$ whose expected loss is the same, or smaller, than that of $g(\cdot)$.*

Proof. Let $\mathbf{U}$ be an $m \times m$ random matrix that is independent of $\mathbf{A}$ and W, and is distributed according to Haar measure on the compact group of $m \times m$ orthogonal matrices. Haar measure is (uniquely) defined by the requirement that, for every $m \times m$ orthogonal matrix C, both $C\mathbf{U}$ and $\mathbf{U}C$ have the same distribution as $\mathbf{U}$ (*c.f.* (Hofmann & Morris 2006)). Let $\mathbf{V}$ be an $n \times n$ random matrix distributed according to the Haar measure on the compact group of $n \times n$ orthogonal matrices that is independent of $\mathbf{A}$, W and $\mathbf{U}$. Given a reconstruction scheme $g(\cdot)$, define a new reconstruction scheme

$$\tilde{g}(Y) \;=\; \mathbb{E}[\mathbf{U}' g(\mathbf{U}Y\mathbf{V}')\mathbf{V} \,|\, Y].$$

It follows from the definition of $\mathbf{U}$ and $\mathbf{V}$ that $\tilde{g}(\cdot)$ is orthogonally invariant. The independence of $\{\mathbf{U}, \mathbf{V}\}$ and $\{\mathbf{A}, W\}$ ensures that conditioning on Y is equivalent to conditioning on $\{\mathbf{A}, W\}$, which yields the equivalent representation

$$\tilde{g}(Y) \;=\; \mathbb{E}[\mathbf{U}' g(\mathbf{U}Y\mathbf{V}')\mathbf{V} \,|\, \mathbf{A}, W].$$

Therefore,

$$\begin{aligned}
\mathbb{E}\,\mathrm{Loss}(\mathbf{A}, \tilde{g}(Y)) \;&=\; \mathbb{E}\left\|\mathbb{E}[\mathbf{U}' g(\mathbf{U}Y\mathbf{V}')\mathbf{V} - \mathbf{A} \,|\, \mathbf{A}, W]\right\|_F^2 \\
&\leq\; \mathbb{E}\|\mathbf{U}' g(\mathbf{U}Y\mathbf{V}')\mathbf{V} - \mathbf{A}\|_F^2 \\
&=\; \mathbb{E}\|g(\mathbf{U}Y\mathbf{V}') - \mathbf{U}\mathbf{A}\mathbf{V}'\|_F^2.
\end{aligned}$$

The inequality above follows from the conditional version of Jensen's inequality applied to each term in the sum defining the squared norm. The final equality follows from the orthogonality of $\mathbf{U}$ and $\mathbf{V}$. The last term in the previous display can be analyzed as follows:

$$\begin{aligned}
\mathbb{E}\left\|g(\mathbf{U}Y\mathbf{V}') - \mathbf{U}\mathbf{A}\mathbf{V}'\right\|_F^2 \;&=\; \mathbb{E}\Big[\mathbb{E}\big(\|g(\mathbf{U}\mathbf{A}\mathbf{V}' + n^{-1/2}\mathbf{U}W\mathbf{V}') - \mathbf{U}\mathbf{A}\mathbf{V}'\|_F^2 \,|\, \mathbf{U}, \mathbf{V}, \mathbf{A}\big)\Big] \\
&=\; \mathbb{E}\Big[\mathbb{E}\big(\|g(\mathbf{U}\mathbf{A}\mathbf{V}' + n^{-1/2}W) - \mathbf{U}\mathbf{A}\mathbf{V}'\|_F^2 \,|\, \mathbf{U}, \mathbf{V}, \mathbf{A}\big)\Big] \\
&=\; \mathbb{E}\left\|g(\mathbf{U}\mathbf{A}\mathbf{V}' + n^{-1/2}W) - \mathbf{U}\mathbf{A}\mathbf{V}'\right\|_F^2.
\end{aligned}$$

The first equality above follows from the definition of Y. The second equality follows from

the independence of W and $\mathbf{U}, \mathbf{A}, \mathbf{V}$, and the orthogonal invariance of $\mathcal{L}(W)$. By a similar argument, using the orthogonal invariance of $\mathcal{L}(\mathbf{A})$, we have

$$\begin{aligned}
\mathbb{E}\|g(\mathbf{UAV}' + n^{-1/2}W) - \mathbf{UAV}'\|_F^2 &= \mathbb{E}\Big[\mathbb{E}\big(\|g(\mathbf{UAV}' + n^{-1/2}W) - \mathbf{UAV}'\|_F^2 \,|\, \mathbf{U}, \mathbf{V}, W\big)\Big] \\
&= \mathbb{E}\Big[\mathbb{E}\big(\|g(\mathbf{A} + n^{-1/2}W) - \mathbf{A}\|_F^2 \,|\, \mathbf{U}, \mathbf{V}, W\big)\Big] \\
&= \mathbb{E}\|g(\mathbf{A} + n^{-1/2}W) - \mathbf{A}\|_F^2.
\end{aligned}$$

The final term above is $\mathbb{E}\,\mathrm{Loss}(\mathbf{A}, g(Y))$. This completes the proof. $\square$

In what follows we will restrict our attention to orthogonally invariant reconstruction schemes.

1.3.1 Singular Value Decomposition

A natural starting point for reconstruction of a target matrix A is the singular value decomposition (SVD) of the observed matrix Y. The SVD of Y is intimately connected with orthogonally invariant reconstruction methods. Recall that the singular value decomposition of an $m \times n$ matrix B is given by the factorization

$$B = UDV' = \sum_{j=1}^{m\wedge n} d_j u_j v_j'.$$

Here U is an $m \times m$ orthogonal matrix with columns u_j, V is an $n \times n$ orthogonal matrix with columns v_j, and D is an $m \times n$ matrix with diagonal entries $D_{jj} = d_j \geq 0$ for $j = 1, \ldots, m\wedge n$, and all other entries equal to zero. The numbers $d_1 \geq d_2 \geq ... \geq d_{m\wedge n} \geq 0$ are the singular values of B. The columns u_j (and v_j) are the left (and right) singular vectors of B. Although it is not necessarily square, we will refer to D as a diagonal matrix and write $D = \mathrm{diag}(d_1, \ldots, d_{m\wedge n})$.

An immediate consequence of the SVD is that $U'BV = D$, so we can diagonalize B by means of left and right orthogonal multiplications. The next proposition follows from our ability to diagonalize the target matrix A in the reconstruction problem.

Proposition 5. *Let $Y = A + n^{-1/2}W$, where W has an orthogonally invariant distribution. If $g(\cdot)$ is an orthogonally invariant reconstruction scheme, then for any fixed target matrix A, the distribution of Loss$(A, g(Y))$, and in particular $\mathbb{E}$Loss$(A, g(Y))$, depends only on the singular*

values of A.

Proof. Let UD_AV' be the SVD of A. Then $D_A = U'AV$, and as the Frobenius norm is invariant under left and right orthogonal multiplications,

$$\begin{aligned}\text{Loss}(A, g(Y)) &= \| g(Y) - A \|_F^2 = \| U'(g(Y) - A)V \|_F^2 \\ &= \| U'g(Y)V - U'AV \|_F^2 = \| g(U'YV) - D_A \|_F^2 \\ &= \| g(D_A + n^{-1/2}U'WV) - D_A \|_F^2.\end{aligned}$$

The result now follows from the fact that UWV' has the same distribution as W. □

We now address the implications of our ability to diagonalize the observed matrix Y. Let $g(\cdot)$ be a orthogonally invariant reconstruction method, and let UDV' be the singular value decomposition of Y. It follows from the orthogonal invariance of $g(\cdot)$ that

$$g(Y) = g(UDV') = Ug(D)V' = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}u_iv_j' \tag{1.5}$$

where c_{ij} depend only on the singular values of Y. In particular, any orthogonally invariant $g(\cdot)$ reconstruction method is completely determined by how it acts on diagonal matrices. The following theorem allows us to substantially refine the representation (1.5).

Theorem 6. *Let $g(\cdot)$ be an orthogonally invariant reconstruction scheme. Then $g(Y)$ is diagonal whenever Y is diagonal.*

Proof. Assume without loss of generality that $m \geq n$. Let the observed matrix Y be diagonal, $Y = \text{diag}(d_1, d_2, ..., d_n)$, and let $\widehat{A} = g(Y)$ be the reconstructed matrix. Fix a row index $1 \leq k \leq m$. We will show that $\widehat{A}_{kj} = 0$ for all $j \neq k$. Let D_L be an $m \times m$ matrix derived from the identity matrix by flipping the sign of the k^{th} diagonal element. More formally, $D_L = I - 2e_ke_k'$, where e_k is the k^{th} standard basis vector in $\mathbb{R}^m$. The matrix D_L is known as a Householder reflection.

Let D_R be the top left $n \times n$ submatrix of D_L. Clearly $D_LD_L' = I$ and $D_RD_R' = I$, so both D_L and D_R are orthogonal. Moreover, all three matrices D_L, Y, and D_R are diagonal, and therefore we have the identity $Y = D_LYD_R$. It then follows from the orthogonal invariance of

$g(\cdot)$ that

$$\widehat{A} \;=\; g(Y) \;=\; g(D_L Y D_R) \;=\; D_L\, g(Y)\, D_R \;=\; D_L\, \widehat{A}\, D_R.$$

The $(i,j)^{th}$ element of the matrix $D_L \widehat{A}\, D_R$ is $\widehat{A}_{ij}(-1)^{\delta_{ik}}(-1)^{\delta_{jk}}$, and therefore $\widehat{A}_{kj} = -\widehat{A}_{kj}$ if $j \neq k$. As k was arbitrary, $\widehat{A}$ is diagonal. □

As an immediate corollary of Theorem 6 and equation (1.5) we obtain a compact, and useful, representation of any orthogonally invariant reconstruction scheme $g(\cdot)$.

Corollary 7. *Let $g(\cdot)$ be an orthogonally invariant reconstruction scheme. If the observed matrix Y has singular value decomposition $Y = \sum d_j u_j v_j'$ then the reconstructed matrix*

$$\widehat{A} \;=\; g(Y) \;=\; \sum_{j=1}^{m\wedge n} c_j u_j v_j', \tag{1.6}$$

where the coefficients c_j depend only on the singular values of Y.

The converse of Corollary 7 is true under a mild additional condition. Let $g(\cdot)$ be a reconstruction scheme such that $g(Y) = c_j u_j v_j'$, where $c_j = c_j(d_1, \ldots, d_{m\wedge n})$ are fixed functions of the singular values of Y. If the functions $\{c_j(\cdot)\}$ are such that $c_i = c_j$ whenever $d_i = d_j$, then $g(\cdot)$ is orthogonally invariant. This follows from the uniqueness of the singular value decomposition.

1.4 Hard and Soft Thresholding

Let Y be an observed $m \times n$ matrix with singular value decomposition $\sum_{j=1}^{m\wedge n} d_j u_j v_j'$. Many reconstruction schemes act by shrinking the singular values of the observed matrix towards zero. Shrinkage is typically accomplished by hard or soft thresholding. Hard thresholding schemes set every singular value of Y less than a positive threshold λ equal to zero, leaving other singular values unchanged. The family of hard thresholding schemes is defined by

$$g_\lambda^H(Y) \;=\; \sum_{j=1}^{m\wedge n} d_j I(d_j \geq \lambda)\, u_j v_j', \text{ where } \lambda > 0.$$

Soft thresholding schemes subtract a positive number ν from each singular value, setting values less than ν equal to zero. The family of soft thresholding schemes is defined by

$$g_\nu^S(Y) \;=\; \sum_{j=1}^{m\wedge n} (d_j - \nu)_+ \, u_j v_j', \;\text{ where } \nu > 0.$$

Hard and soft thresholding schemes can be defined equivalently in the respective penalized forms

$$g_\lambda^H(Y) \;=\; \arg\min_B \left\{ \|Y - B\|_F^2 + \lambda^2 \operatorname{rank}(B) \right\}$$

$$g_\nu^S(Y) \;=\; \arg\min_B \left\{ \|Y - B\|_F^2 + 2\nu \, \|B\|_* \right\}.$$

In the second display, $\|B\|_*$ denotes the nuclear norm of B, equal to the sum of its singular values.

In practice, hard and soft thresholding schemes require estimates of the noise variance, as well as the selection of appropriate cutoff or shrinkage parameters. There are numerous methods in the literature for choosing the hard threshold λ. Heuristic methods often make use of the scree plot, which displays the singular values of Y as a function of their rank: λ is typically chosen to be the y-coordinate of a well defined "elbow" in the plot. In recent work, Bunea et al. (2010) propose a specific choice of λ and provide performance guaranties for the resulting hard thresholding scheme using techniques from empirical process theory and complexity regularization. Selection of the soft thresholding shrinkage parameter ν may also be accomplished by a variety of methods. Negahban & Wainwright (2009) propose a specific choice of ν and provide performance guarantees for the resulting soft thresholding scheme. Hard and soft thresholding schemes are orthogonally invariant if the estimates of λ and ν, respectively, depend only on the singular values of Y.

1.5 Asymptotic Approach

The families of hard and soft thresholding methods described above include many existing reconstruction schemes. Both thresholding approaches seek low rank (sparse) estimates of the target matrix, both can be naturally formulated as optimization problems, and under mild conditions both yield orthogonally invariant reconstruction schemes. However, the family of

all orthogonally invariant reconstruction schemes encompasses a much broader class of possible reconstruction procedures, and it is natural to consider alternatives to thresholding that may offer better performance.

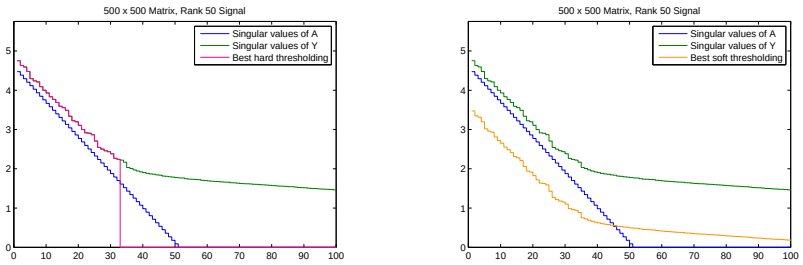

Figure 1.2: Singular values of hard and soft thresholding estimates.

Figure 1.2 illustrates the action of hard and soft thresholding on a 500×500 matrix with a rank 50 signal. The blue line marks the singular values of the signal A and the green line marks the those of the observed matrix Y. The plots show the singular values of the hard and soft thresholding estimates incorporating the best choice of parameters λ and ν, respectively. Clearly, neither thresholding scheme delivers an accurate estimate of the original singular values (or the original matrix). Moreover, the figures suggest that a hybrid scheme that combines soft and hard thresholding might offer better performance. We construct an improved reconstruction scheme in a principled fashion, by studying the effect of noise on low-rank signal matrices. The key tools in this analysis are several recent results from random matrix theory.

Random matrix theory is broadly concerned with the spectral properties of random matrices, and is an obvious starting point for an analysis of matrix reconstruction. The matrix reconstruction problem has several points of intersection with random matrix theory. Recently a number of authors have studied low rank deformations of Wigner matrices (Capitaine et al. 2009, Féral & Péché 2007, Maida 2007, Péché 2006). However, their results concern symmetric matrices, a constraint not present in the reconstruction model, and are not directly applicable to the reconstruction problem of interest here. (Indeed, our simulations of non-symmetric matrices exhibit behavior deviating from that predicted by the results of these papers.) A signal plus noise framework similar to matrix reconstruction is studied in Dozier & Silverstein (2007), Nadakuditi & Silverstein (2007), but these papers focus on the singular values of the observed

matrix (Dozier & Silverstein) or recovery of the singular values of the signal (Nadakuditi & Silverstein), and do not consider the more general problem of reconstruction.

Our proposed denoising scheme is based on the theory of spiked population models in random matrix theory. Using existing results on spiked population models, we establish asymptotic connections between the singular values and vectors of the target matrix A and those of the observed matrix Y. These asymptotic connections provide us with finite-sample estimates that can be applied in a non-asymptotic setting to matrices of small or moderate dimensions.

1.5.1 Asymptotic Matrix Reconstruction Model

The proposed reconstruction method is based on an asymptotic version of the matrix reconstruction problem (1.2). For $n \geq 1$ let integers $m = m(n)$ be defined in such a way that

$$\frac{m}{n} \to c > 0 \quad \text{as} \quad n \to \infty. \tag{1.7}$$

For each n let Y, A, and W be $m \times n$ matrices such that model (1.2) holds:

$$Y = A + \frac{1}{\sqrt{n}} W, \tag{1.8}$$

where the entries of W are independent $N(0,1)$ random variables. We assume that the target matrix A has fixed rank $r \geq 0$ and fixed non-zero singular values $\lambda_1(A), \ldots, \lambda_r(A)$ that are are independent of n. The constant c represents the limiting aspect ratio of the observed matrices Y. The scale factor $n^{-1/2}$ ensures that the singular values of the target matrix are comparable to those of the noise. Model (1.8) matches the asymptotic model used by Capitaine et al. (2009), Féral & Péché (2007) in their study of fixed rank perturbations of Wigner matrices.

In what follows $\lambda_j(B)$ will denote the j-th singular value of a matrix B, and $u_j(B)$ and $v_j(B)$ will denote, respectively, the left and right singular values corresponding to $\lambda_j(B)$. Our first proposition concerns the behavior of the singular values of Y when the target matrix A is equal to zero.

Proposition 8. *Under the asymptotic reconstruction model with $A = 0$ the empirical distribution of the singular values $\lambda_1(Y) \geq \cdots \geq \lambda_{m \wedge n}(Y)$ converges weakly to a (non-random)*

distribution with density

$$f_Y(s) = \frac{s^{-1}}{\pi(c \wedge 1)}\sqrt{(a-s^2)(s^2-b)}, \qquad s \in [\sqrt{a}, \sqrt{b}], \tag{1.9}$$

where $a = (1-\sqrt{c})^2$ *and* $b = (1+\sqrt{c})^2$. *Moreover,* $\lambda_1(Y) \xrightarrow{P} 1+\sqrt{c}$ *and* $\lambda_{m\wedge n}(Y) \xrightarrow{P} 1-\sqrt{c}$ *as* n *tends to infinity.*

The existence and form of the density $f_Y(\cdot)$ are a consequence of the classical Marčenko-Pastur theorem (Marčenko & Pastur 1967, Wachter 1978). The in-probability limits of $\lambda_1(Y)$ and $\lambda_{m\wedge n}(Y)$ follow from later work of Geman (1980) and Wachter (1978), respectively. If $c = 1$, the density function $f_Y(s)$ simplifies to the quarter-circle law $f_Y(s) = \pi^{-1}\sqrt{4-s^2}$ for $s \in [0, 2]$.

The next two results concern the limiting eigenvalues and eigenvectors of Y when A is non-zero. Proposition 9 relates the limiting eigenvalues of Y to the (fixed) eigenvalues of A, while Proposition 10 relates the limiting singular vectors of Y to the singular vectors of A. Proposition 9 is based on recent work of Baik & Silverstein (2006), while Proposition 10 is based on recent work of Paul (2007), Nadler (2008), and Lee et al. (2010). The proofs of both results are given in Section 1.8.2.

Proposition 9. *Let* Y *follow the asymptotic matrix reconstruction model (1.8) with target singular values* $\lambda_1(A) \geq ... \geq \lambda_r(A) > 0$. *For* $1 \leq j \leq r$, *as* n *tends to infinity,*

$$\lambda_j(Y) \xrightarrow{P} \begin{cases} \left(1 + \lambda_j^2(A) + c + \frac{c}{\lambda_j^2(A)}\right)^{1/2} & \text{if} \quad \lambda_j(A) > \sqrt[4]{c} \\ 1+\sqrt{c} & \text{if} \quad 0 < \lambda_j(A) \leq \sqrt[4]{c} \end{cases}$$

The remaining singular values $\lambda_{r+1}(Y), \ldots, \lambda_{m\wedge n}(Y)$ *of* Y *are associated with the zero singular values of* A: *their empirical distribution converges weakly to the limiting distribution in Proposition 8.*

Proposition 10. *Let* Y *follow the asymptotic matrix reconstruction model (1.8) with distinct target singular values* $\lambda_1(A) > \lambda_2(A) > ... > \lambda_r(A) > 0$. *Fix* j *such that* $\lambda_j(A) > \sqrt[4]{c}$. *Then as*

n tends to infinity,

$$\langle u_j(Y), u_j(A)\rangle^2 \xrightarrow{P} \left(1 - \frac{c}{\lambda_j^4(A)}\right) / \left(1 + \frac{c}{\lambda_j^2(A)}\right)$$

and

$$\langle v_j(Y), v_j(A)\rangle^2 \xrightarrow{P} \left(1 - \frac{c}{\lambda_j^4(A)}\right) / \left(1 + \frac{1}{\lambda_j^2(A)}\right)$$

Moreover, if $k \neq j$, $1 \leq k \leq r$ *then* $\langle u_i(Y), u_k(A)\rangle \xrightarrow{P} 0$ *and* $\langle v_i(Y), v_k(A)\rangle \xrightarrow{P} 0$ *as n tends to infinity.*

The limits established in Proposition 9 indicate a phase transition. If the singular value $\lambda_j(A)$ is less than or equal to $\sqrt[4]{c}$ then, asymptotically, the singular value $\lambda_j(Y)$ lies within the support of the Marčenko-Pastur distribution and is not distinguishable from the noise singular values. On the other hand, if $\lambda_j(A)$ exceeds $\sqrt[4]{c}$ then, asymptotically, $\lambda_j(Y)$ lies outside the support of the Marčenko-Pastur distribution, and the corresponding left and right singular vectors of Y are associated with those of A (Proposition 10).

1.6 Proposed Reconstruction Scheme

Let Y be an observed $m \times n$ matrix generated from the additive model $Y = A + n^{-1/2}\sigma W$. Assume for the moment that the variance σ^2 of the noise is known, and equal to one. Estimation of σ is discussed in the next subsection. Let

$$Y = \sum_{j=1}^{m\wedge n} \lambda_j(Y)\, u_j(Y) v_j'(Y)$$

be the SVD of Y. Following the discussion in Section 1.3, we seek an estimate of A of the form

$$\widehat{A} = \sum_{j=1}^{m\wedge n} c_j\, u_j(Y) v_j'(Y),$$

where each coefficient c_j depends only on the singular values $\lambda_1(Y), \ldots, \lambda_{m\wedge n}(Y)$ of Y.

Our proposed reconstruction scheme is derived from the limiting relations in Propositions 9 and 10. By way of approximation, we treat these relations as exact in the fixed dimension setting under study, using the symbols $\overset{l}{=}$, $\overset{l}{\leq}$ and $\overset{l}{>}$ to denote limiting equality and inequal-

ity relations. Suppose that the singular values and vectors of A are known. Then we seek coefficients $\{c_j\}$ minimizing

$$\mathrm{Loss}(A, \widehat{A}) \;=\; \Big\| \sum_{j=1}^{m\wedge n} c_j \, u_j(Y) v_j'(Y) \;-\; \sum_{j=1}^{m\wedge n} \lambda_j(A) \, u_j(A) v_j'(A) \Big\|_F^2.$$

Phase transition phenomenon (Proposition 9) indicated that we can restrict the first sum to the first $r_0 = \#\{j : \lambda_j(A) > \sqrt[4]{c}\}$ elements. By definition of A only first r elements of the second sum are non-zero, so

$$\mathrm{Loss}(A, \widehat{A}) \;=\; \Big\| \sum_{j=1}^{r_0} c_j \, u_j(Y) v_j'(Y) \;-\; \sum_{j=1}^{r} \lambda_j(A) \, u_j(A) v_j'(A) \Big\|_F^2$$

Proposition 10 ensures that the left singular vectors $u_i(Y)$ and $u_k(A)$ are asymptotically orthogonal if $i \neq k, i \leq r_0, k \leq r$, and therefore

$$\mathrm{Loss}(A, \widehat{A}) \;\overset{l}{=}\; \sum_{j=1}^{r_0} \big\| c_j \, u_j(Y) v_j'(Y) \;-\; \lambda_j(A) \, u_j(A) v_j'(A) \big\|_F^2.$$

Fix $1 \leq j \leq r_0$. Expanding the j-th term in the above sum gives

$$\begin{aligned}
&\big\| \lambda_j(A) \, u_j(A) v_j'(A) \;-\; c_j \, u_j(Y) v_j'(Y) \big\|_F^2 \\
&= c_j^2 \, \big\| u_j(Y) v_j'(Y) \big\|_F^2 \;+\; \lambda_j^2(A) \, \big\| u_j(A) v_j'(A) \big\|_F^2 \;-\; 2 c_j \lambda_j(A) \, \big\langle u_j(A) v_j'(A), u_j(Y) v_j'(Y) \big\rangle \\
&= \lambda_j^2(A) \;+\; c_j^2 \;-\; 2 c_j \lambda_j(A) \, \big\langle u_j(A), u_j(Y) \big\rangle \, \big\langle v_j(A), v_j(Y) \big\rangle.
\end{aligned}$$

Differentiating the last expression with respect to c_j yields the optimal value

$$c_j^* \;=\; \lambda_j(A) \, \big\langle u_j(A), u_j(Y) \big\rangle \, \big\langle v_j(A), v_j(Y) \big\rangle. \tag{1.10}$$

In order to estimate the coefficient c_j^* we consider separately singular values of Y that are less than or greater than $1 + \sqrt{c}$, where $c = m/n$ is the aspect ratio of Y. By Proposition 9, the relation $\lambda_j(Y) \overset{l}{\leq} 1 + \sqrt{c}$ implies $\lambda_j(A) \leq \sqrt[4]{c}$ and so the j-th component is not recoverable. Thus if $\lambda_j(Y) \leq 1 + \sqrt{c}$ we set $c_j^* = 0$.

On the other hand, $\lambda_j(Y) \overset{l}{>} 1 + \sqrt{c}$ implies $\lambda_j(A) > \sqrt[4]{c}$ and each of the inner products in

(1.10) are asymptotically positive. Moreover, the displayed equations in Propositions 9 and 10 can be used to obtain estimates of each term in (1.10) based only on the (observed) singular values of Y and its aspect ratio c. In particular,

$$\widehat{\lambda}_j^2(A) \;=\; \frac{1}{2}\left[\lambda_j^2(Y) - (1+c) + \sqrt{[\lambda_j^2(Y) - (1+c)]^2 - 4c}\right] \;\text{ estimates }\; \lambda_j^2(A),$$

$$\hat{\theta}_j^2 \;=\; \left(1 - \frac{c}{\widehat{\lambda}_j^4(A)}\right) \Big/ \left(1 + \frac{c}{\widehat{\lambda}_j^2(A)}\right) \;\text{ estimates }\; \langle u_j(A), u_j(Y)\rangle^2,$$

$$\hat{\phi}_j^2 \;=\; \left(1 - \frac{c}{\widehat{\lambda}_j^4(A)}\right) \Big/ \left(1 + \frac{1}{\widehat{\lambda}_j^2(A)}\right) \;\text{ estimates }\; \langle v_j(A), v_j(Y)\rangle^2.$$

With these estimates in hand, the proposed reconstruction scheme is defined via the equation

$$G_o^{RMT}(Y) \;=\; \sum_{\lambda_j(Y) > 1+\sqrt{c}} \widehat{\lambda}_j(A)\, \hat{\theta}_j\, \hat{\phi}_j\, u_j(Y) v_j'(Y), \tag{1.11}$$

where $\widehat{\lambda}_j(A)$, $\hat{\theta}_j$, and $\hat{\phi}_j$ are the positive square roots of the estimates defined above.

In general, the variance σ^2 of the noise is not known, but we have access to an estimate $\widehat{\sigma}^2$ of σ^2. In this case, we define

$$G^{RMT}(Y) \;=\; \widehat{\sigma}\; G_o^{RMT}\left(\frac{Y}{\widehat{\sigma}}\right), \tag{1.12}$$

where $G_o^{RMT}(\cdot)$ is the estimate defined in (1.11). An estimate $\widehat{\sigma}^2$ of the noise variance is discussed in the next subsection.

The RMT method shares features with both hard and soft thresholding. The RMT method sets to zero singular values of Y smaller than the threshold $(1+\sqrt{c})$, and it shrinks the remaining singular values. However, unlike soft thresholding the amount of shrinkage depends on the singular values, the larger singular values are shrunk less than the smaller ones. This latter feature is similar to that of LASSO type estimators based on an L_q penalty (also known as bridge estimators, Fu (1998)) with $0 < q < 1$. It is important to note that, unlike hard and soft thresholding schemes, the proposed RMT method has no tuning parameters. The only

unknown, the noise variance, is estimated within the procedure.

1.6.1 Estimation of the Noise Variance

Suppose that $Y = A + \sigma n^{-1/2} W$ is derived from the asymptotic reconstruction model with σ unknown. One may approach the estimation of σ in a fashion analogous to the estimation of A. In particular, any orthogonally invariant estimate of σ will depend only on the singular values of Y.

Proposition 9 shows that the empirical distribution of the $(m - r)$ singular values $S = \{\lambda_j(Y/\sigma) : \lambda_j(A) = 0\}$ converges weakly to a distribution with density (1.9) supported on the interval $[|1 - \sqrt{c}|, 1 + \sqrt{c}]$. Following the general approach outlined in (Györfi et al. 1996), we estimate σ by minimizing the Kolmogorov-Smirnov distance between the empirical and the theoretical limiting sample distributions of singular values. Let F be the CDF of the density (1.9). For each $\sigma > 0$ let $\widehat{S}_\sigma$ be the set of singular values $\lambda_j(Y)$ that fall in the interval $[\sigma|1 - \sqrt{c}|, \sigma(1 + \sqrt{c})]$, and let $\widehat{F}_\sigma$ be the empirical CDF of $\widehat{S}_\sigma$. Then

$$K(\sigma) = \sup_s |F(s/\sigma) - \widehat{F}_\sigma(s)|$$

is the Kolmogorov-Smirnov distance between the empirical and theoretical singular value distribution functions, and define our estimate

$$\hat{\sigma}(Y) = \arg\min_{\sigma > 0} K(\sigma) \tag{1.13}$$

to be the value of σ minimizing $K(\sigma)$. A routine argument shows that the estimator $\hat{\sigma}$ is scale invariant, in the sense that $\hat{\sigma}(\beta Y) = \beta \hat{\sigma}(Y)$ for each $\beta > 0$. By considering the jump points of the empirical CDF, the supremum in $K(\sigma)$ simplifies to

$$K(\sigma) = \max_{s_i \in \widehat{S}_\sigma} \left| F(s_i/\sigma) - \frac{i - 1/2}{|\widehat{S}_\sigma|} \right| + \frac{1}{2|\widehat{S}_\sigma|},$$

where $\{s_i\}$ are the ordered elements of $\widehat{S}_\sigma$. The objective function $K(\sigma)$ is discontinuous at points where the $\widehat{S}_\sigma$ changes, so we minimize it over a fine grid of points in the range where $|\widehat{S}_\sigma| > (m \wedge n)/2$ and $\sigma(1 + \sqrt{c}) < 2\lambda_1(Y)$. The closed form of the cumulative distribution

function $F(\cdot)$ is presented is Section 1.8.1.

1.7 Simulations

We carried out a simulation study to evaluate the performance of the RMT reconstruction scheme $G^{RMT}(\cdot)$ defined in (1.12) using the variance estimate $\widehat{\sigma}$ in (1.13). The study compared the performance of $G^{RMT}(\cdot)$ to three alternatives: the best hard thresholding reconstruction scheme, the best soft thresholding reconstruction scheme, and the best orthogonally invariant reconstruction scheme. Each of the three competing alternatives is an oracle-type procedure that is based on information about the target matrix A that is not available to $G^{RMT}(\cdot)$.

1.7.1 Hard and Soft Thresholding Oracle Procedures

Hard and soft thresholding schemes require specification of a threshold parameter that can depend on the observed matrix Y. Estimation of the noise variance can be incorporated into the choice of the threshold parameter. In order to compare the performance of $G^{RMT}(\cdot)$ against every possible hard and soft thresholding scheme, we define oracle procedures

$$G^H(Y) \;=\; g^H_{\lambda^*}(Y) \quad \text{where} \quad \lambda^* \;=\; \underset{\lambda>0}{\arg\min} \left\| A - g^H_\lambda(Y) \right\|_F^2 \tag{1.14}$$

$$G^S(Y) \;=\; g^S_{\nu^*}(Y) \quad \text{where} \quad \nu^* \;=\; \underset{\nu>0}{\arg\min} \left\| A - g^S_\nu(Y) \right\|_F^2 \tag{1.15}$$

using knowledge of the target A. By definition, the loss $\|A - G^H(Y)\|_F^2$ of $G^H(Y)$ is less than that of any hard thresholding scheme, and similarly the loss of $G^S(Y)$ is less than that of any soft thresholding procedure. In effect, the oracle procedures have access to both the unknown target matrix A and the unknown variance σ. They are constrained only by the form of their respective thresholding families. The oracle procedures are not realizable in practice.

1.7.2 Orthogonally Invariant Oracle Procedure

As shown in Theorem 6, every orthogonally invariant reconstruction scheme $g(\cdot)$ has the form

$$g(Y) \;=\; \sum_{j=1}^{m\wedge n} c_j \, u_j(Y) v_j(Y)',$$

where the coefficients c_j are functions of the singular values of Y. The orthogonally invariant oracle scheme has coefficients c_j^* minimizing the loss

$$\left\|A - \sum_{j=1}^{m\wedge n} c_j\, u_j(Y)v_j(Y)'\right\|_F^2$$

over all choices c_j. As is the case with the hard and soft thresholding oracle schemes, the coefficients c_j^* depend on the target matrix A, which in practice is unknown.

The (rank one) matrices $\{u_j(Y)v_j(Y)'\}$ form an orthonormal basis of an $m \wedge n$-dimensional subspace of the mn-dimensional space of all $m \times n$ matrices. The optimal coefficient c_j^* is simply the matrix inner product $\langle A, u_j(Y)v_j(Y)'\rangle$, and the orthogonally invariant oracle scheme has the form of a projection

$$G^*(Y) \;=\; \sum_{j=1}^{m\wedge n} \left\langle A, u_j(Y)v_j(Y)'\right\rangle u_j(Y)v_j(Y)' \tag{1.16}$$

By definition, for any orthogonally invariant reconstruction scheme $g(\cdot)$ and observed matrix Y, we have $\|A - G^*(Y)\|_F^2 \le \|A - g(Y)\|_F^2$.

1.7.3 Simulations

We compared the reconstruction schemes $G^H(Y)$, $G^S(Y)$ and $G^{RMT}(Y)$ to $G^*(Y)$ on a wide variety of target matrices generated according to the model (1.2). As shown in Proposition 5, the distribution of the loss $\|A - G(Y)\|_F^2$ depends only on the singular values of A, so we considered only diagonal target matrices. As the variance estimate used in $G^{RMT}(\cdot)$ is scale invariant, all simulations were run with noise of unit variance. (Estimation of noise variance is not necessary for the oracle reconstruction schemes.)

Square Matrices

Our initial simulations considered 1000×1000 square matrices. Target matrices A were generated using three parameters: the rank r; the largest singular value ($\lambda_1(A)$); and the decay profile of the remaining singular values. We considered ranks $r \in \{1, 3, 10, 32, 100\}$ corresponding to successive powers of $\sqrt{10}$ up to $(m \wedge n)/10$, and maximum singular values $\lambda_1(A) \in \{0.9, 1, 1.1, ..., 10\}\sqrt[4]{c}$ falling below and above the critical threshold of $\sqrt[4]{c} = 1$. We

considered several coefficient decay profiles: (i) all coefficients equal; (ii) linear decay to zero; (iii) linear decay to $\lambda_1(A)/2$; and exponential decay as powers of 0.5, 0.7, 0.9, 0.95, or 0.99. Independent noise matrices W were generated for each target matrix A. All reconstruction schemes were then applied to the resulting matrix $Y = A + n^{-1/2}W$. The total number of generated target matrices was 3,680.

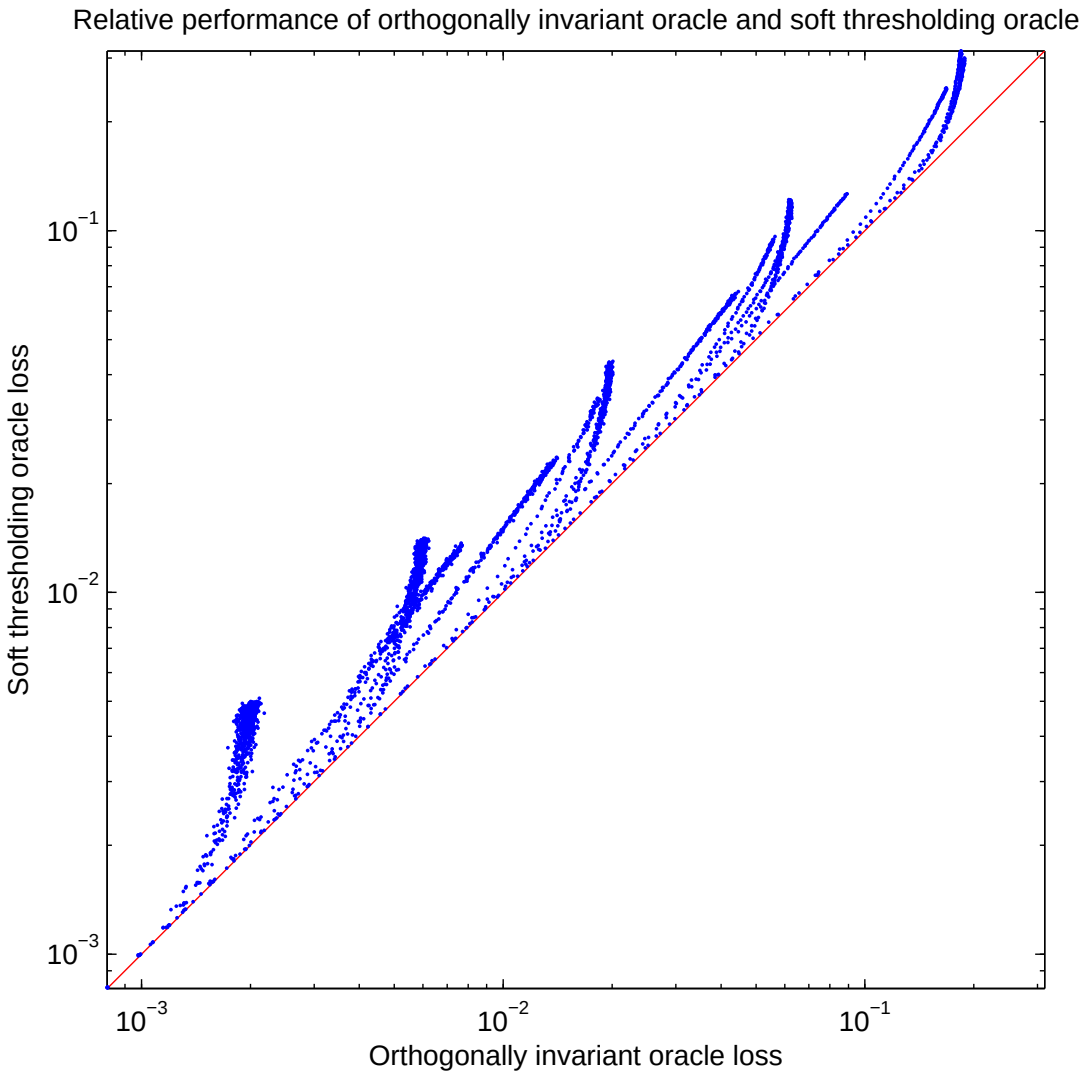

Figure 1.3: Relative performance of soft thresholding and orthogonally invariant oracle methods for 1000×1000 matrices.

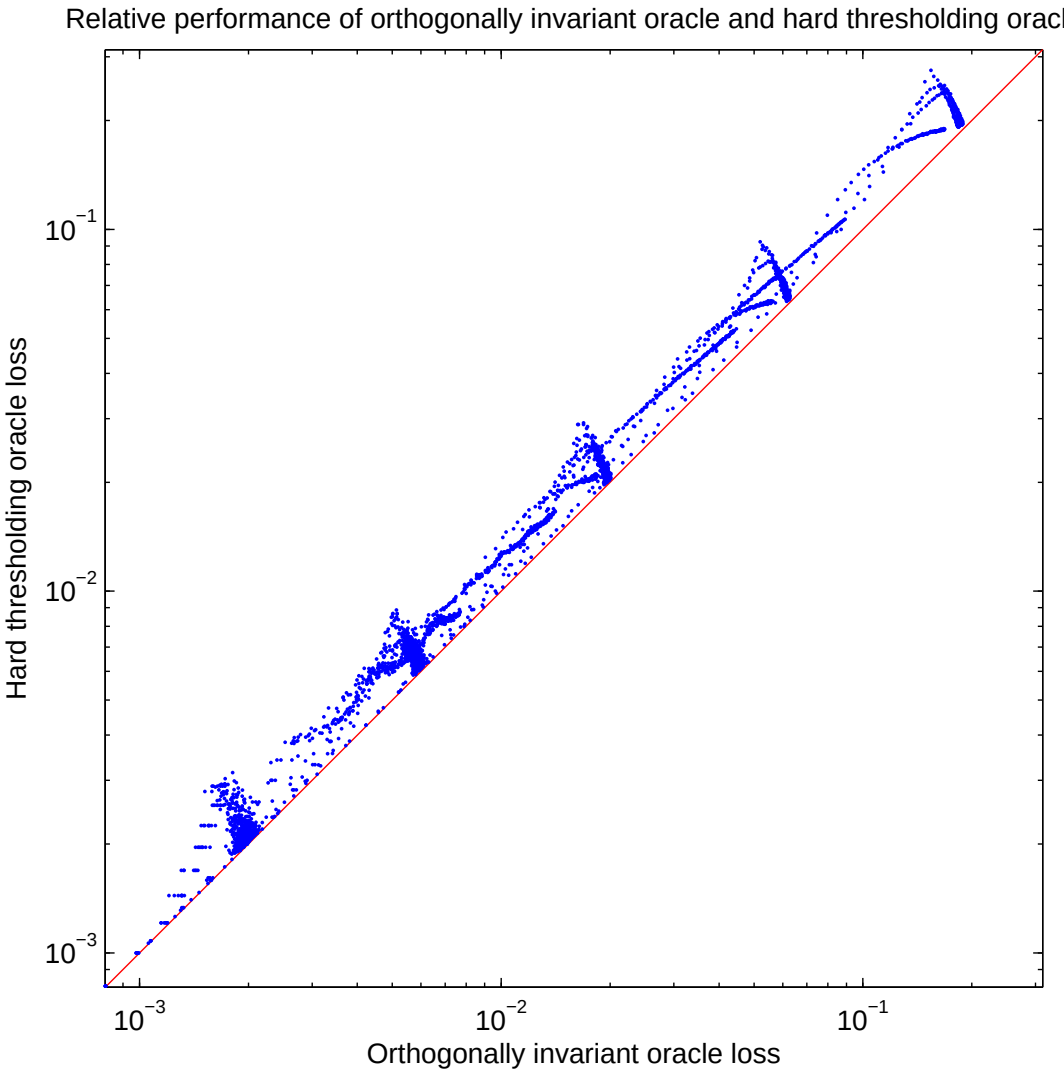

Figure 1.4: Relative performance of hard thresholding and orthogonally invariant oracle methods for 1000×1000 matrices.

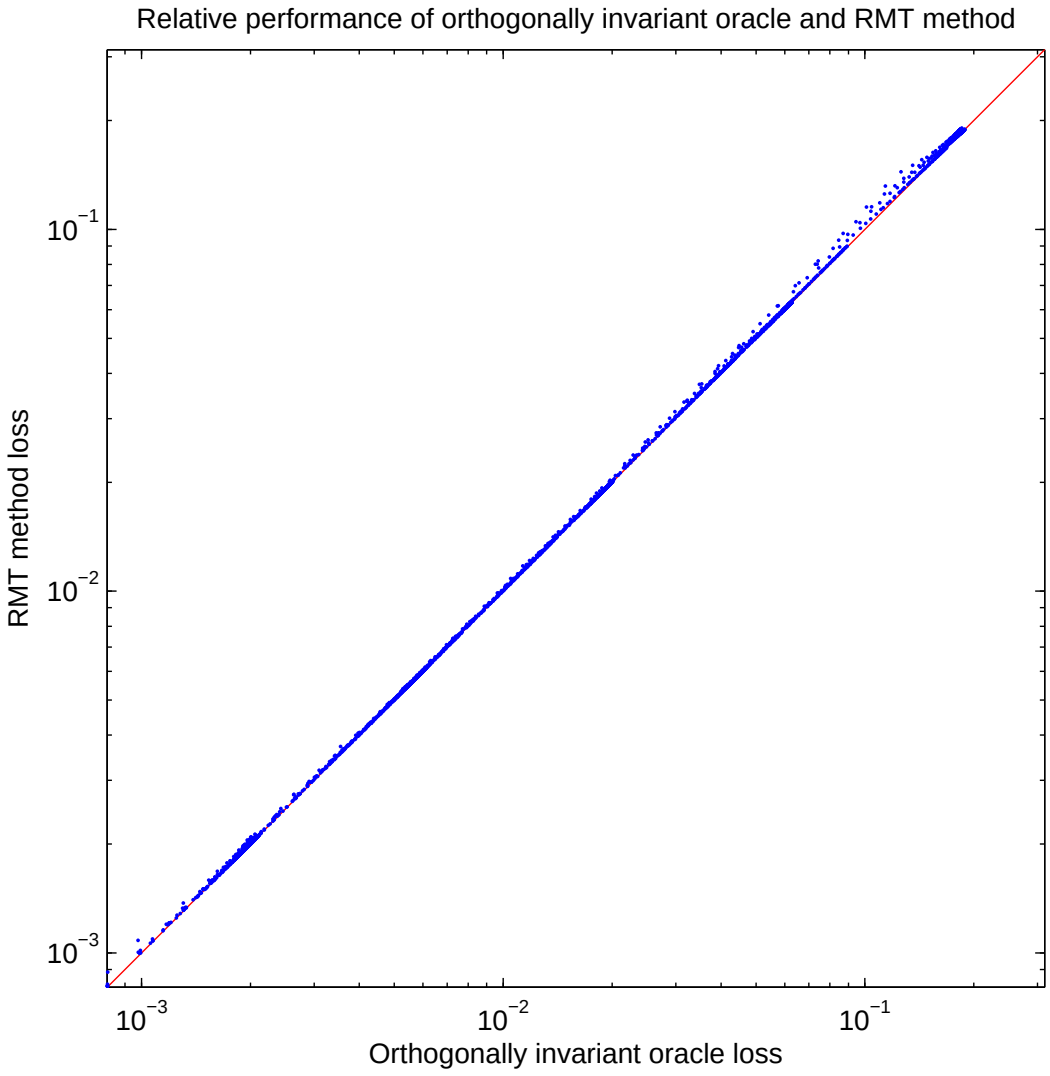

Figure 1.5: Relative performance of RMT method and orthogonally invariant oracle method for 1000×1000 matrices.

Figures 1.3, 1.4, and 1.5 illustrate, respectively, the loss of the best soft thresholding, best hard thresholding and RMT reconstruction methods (y axis) relative to the best orthogonally invariant scheme (x axis). In each case the diagonal represents the performance of the orthogonally invariant oracle: points farther from the diagonal represent worse performance. The plots show clearly that $G^{RMT}(\cdot)$ outperforms the oracle schemes $G^H()$ and $G^S(\cdot)$, and has performance comparable to the orthogonally invariant oracle. In particular, $G^{RMT}(\cdot)$ outperforms any hard or soft thresholding scheme, even if the latter schemes have access to the unknown variance σ and the target matrix A.

In order to summarize the results of our simulations, for each scheme $G(\cdot)$ and for each matrix Y generated from a target matrix A we calculated the relative excess loss of $G()$ with respect to $G^*()$:

$$\mathrm{REL}(A, G(Y)) = \frac{\mathrm{Loss}(A, G(Y))}{\mathrm{Loss}(A, G^*(Y))} - 1 \tag{1.17}$$

The definition of $G^*()$ ensures that relative excess loss is non-negative. The average RELs of $G^S(\cdot)$, $G^H()$, and $G^{RMT}(\cdot)$ across the 3680 simulated 1000×1000 matrices were 68.3%, 18.3%, and 0.61% respectively. Table 1.1 summarizes these results, and the results of analogous simulations carried out on square matrices of different dimensions. The table clearly shows the strong performance of RMT method for matrices with at least 50 rows or columns. Even for $m = n = 50$, the average relative excess loss of the RMT method is almost twice smaller then those of the oracle soft and hard thresholding methods.

Matrix size (square)		2000	1000	500	100	50
Scheme	$G^S(\cdot)$	0.740	0.683	0.694	0.611	0.640
	$G^H(\cdot)$	0.182	0.183	0.178	0.179	0.176
	$G^{RMT}(\cdot)$	0.003	0.006	0.008	0.029	0.071

Table 1.1: Average relative excess losses of oracle soft thresholding, oracle hard thresholding and the proposed RMT reconstruction method for square matrices of different dimensions.

Rectangular Matrices

We performed simulations for rectangular matrices of different dimensions m, n and different aspect ratios $c = m/n$. For each choice of dimensions m, n we simulated target matrices using the same rules as in the square case; rank $r \in \{1, 3, 10, 32, \ldots\}$ not exceeding $(m \wedge n)/10$,

maximum singular values $\lambda_1(A) \in \{0.9, 1, 1.1, ..., 10\}\sqrt[4]{c}$, and coefficients decay profiles like those in the square case. A summary of the results is given in Table 1.2. It shows the average REL for matrices with 2000 rows and 10 to 2000 columns. Although random matrix theory used to construct the RMT scheme requires both m and n to tend to infinity, the numbers in Table 1.2 clearly show that the performance of the RMT scheme is excellent even for small n with average REL between 0.3% and 0.54%. On the contrary, the other two schemes did not reach average REL below 18%.

Matrix size	m	2000	2000	2000	2000	2000	2000
	n	2000	1000	500	100	50	10
Scheme	$G^S(\cdot)$	0.740	0.686	0.653	0.442	0.391	0.243
	$G^H(\cdot)$	0.182	0.188	0.198	0.263	0.292	0.379
	$G^{RMT}(\cdot)$	0.003	0.004	0.004	0.004	0.004	0.005

Table 1.2: Average relative excess loss of oracle soft thresholding, oracle hard thresholding, and RMT reconstruction schemes for matrices with different dimensions and aspect ratios.

1.7.4 Simulation Study of Spiked Population Model and Matrix Reconstruction

In Section 1.8.2 we have built a connection between matrix reconstruction model and spiked population model. The most complicated part is connection between the non-random signal matrix A from matrix reconstruction model with a random matrix $n^{-1/2}X_1$, a part of X from spiked population model. We used this connection to translate several theorems from random matrix theory to determine how the singular values of the unobserved matrix A translate into the singular values of the observed matrix $Y = A + n^{-1/2}W$.

One may question whether this prediction works well and if it does, whether the prediction is better or worse for the matrix reconstruction model compared to the spiked population model. To address this question we have performed additional simulations.

For square matrices of size $m = n = 1000$ we considered rank one signal matrices with singular value $\alpha = 1, 2, \ldots, 1000$. For each signal matrix A, an independent copy of the noise matrix W was generated along with the observed matrix $Y = A + W$. For each α the matrix X from the matching spiked population model was generated as $X = T^{1/2}W$, where $T = \text{diag}(1 + \alpha^2/n, 1, \ldots, 1)$. The largest singular values are then calculated for both Y and X and compared

to the prediction based on Theorem A and Proposition 9.

Figure 1.6 illustrated the findings for matrix reconstruction model on the left two plots and for spiked population models on the right two. The top plots show the largest singular value (of Y on left plot and X on right) against α as blue dots and the predicted values as a red line. The bottom plots show the difference between the realized first singular values and the prediction.

In clear from Figure 1.6 that the prediction for matrix reconstruction model does not just work well, it actually works better than the original prediction for the spiked population model. This result can be explain by the fact that the signal A is non-random in matrix reconstruction model while X_1 in the spiked population model is random. Note that even though $n^{-1/2}X_1$ is random, under spiked population model, its non-zero singular values converge almost surely to non-random limits as $n \to \infty$. In the matrix reconstruction model we remove the randomness of X_1 by replacing it by its non-random asymptotic version A. This explains the better performance of the prediction for matrix reconstruction model illustrated on Figure 1.6.

1.8 Appendix

1.8.1 Cumulative Distribution Function for Variance Estimation

The cumulative density function $F(\cdot)$ is calculated as the integral of $f_{n^{-1/2}W}(s)$. For $c = 1$ it is a common integral $(a = 0, b = 4)$

$$F(x) = \int_{\sqrt{a}}^{x} f(s)ds = \frac{1}{\pi}\int_{0}^{x}\sqrt{b - s^2}ds = \frac{1}{2\pi}\left(x\sqrt{4 - x^2} + 4\arcsin\frac{x}{2}\right)$$

For $c \neq 1$ the calculations are more complicated. First we change variables $t = s^2$

$$F(x) = \int_{\sqrt{a}}^{x} f(s)ds = C\int_{\sqrt{a}}^{x} s^{-2}\sqrt{(b - s^2)(s^2 - a)}ds^2 = C\int_{a}^{x^2} t^{-1}\sqrt{(b - t)(t - a)}dt,$$

where $C = 1/(2\pi(c \wedge 1))$. Next we perform a change of variables to make the expression in the square root look like $h^2 - x^2$. The change of variables is $y = t - [a + b]/2$.

$$F(x) = C\int_{-[b-a]/2}^{x^2-[a+b]/2} \frac{\sqrt{([b - a]/2 - y)(y + [b - a]/2)}}{y + [a + b]/2}dy = C\int_{-2\sqrt{c}}^{x^2-(1+c)} \frac{\sqrt{4c - y^2}}{y + 1 + c}dy,$$

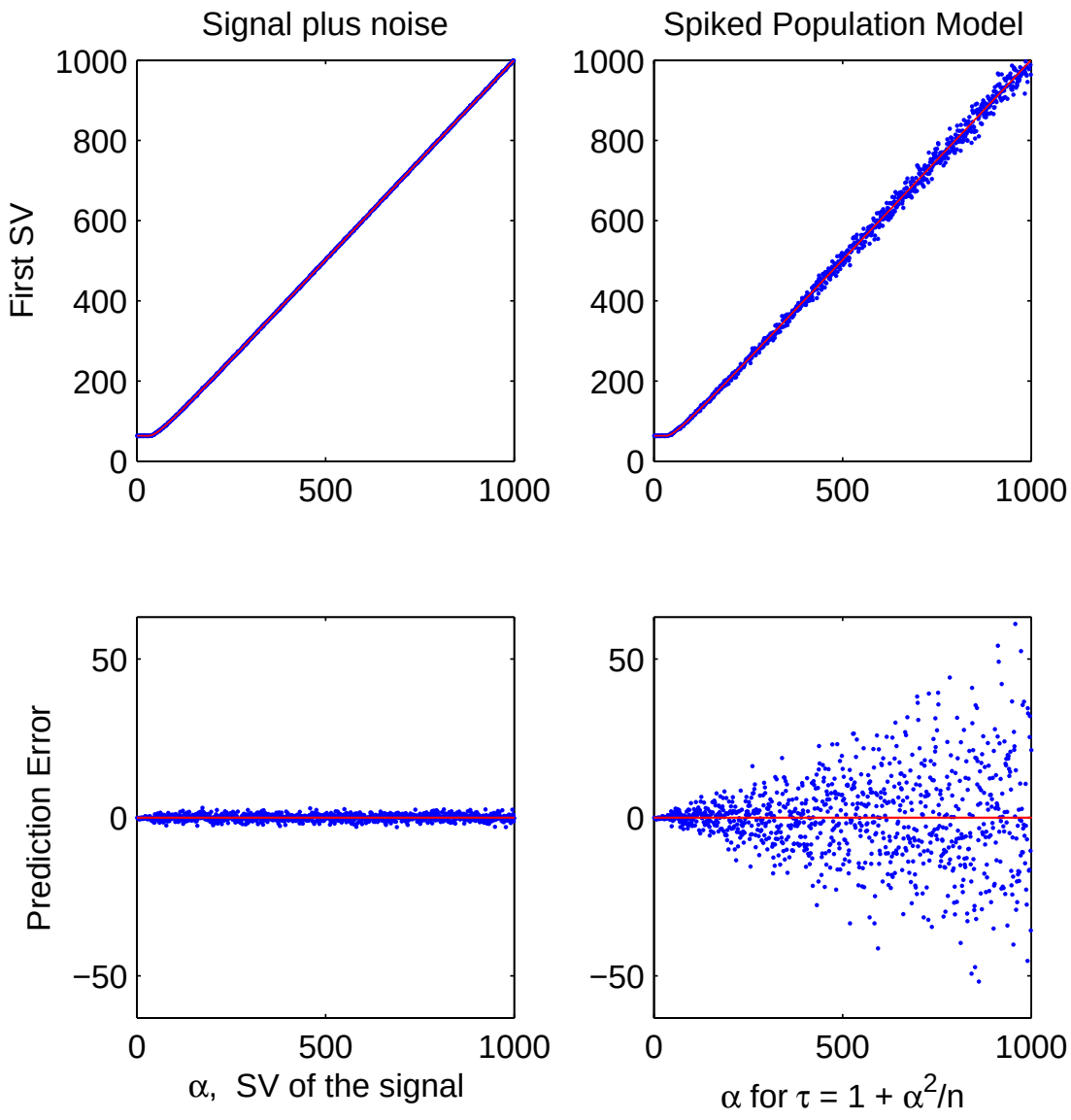

Figure 1.6: Largest singular values of the matched rank one matrices from reconstruction and spiked population models against the signal strength α. See complete description and discussion in Section 1.7.4.

The second equality uses the fact that $a+b=2(1+c)$ and $b-a=4\sqrt{c}$. The simple change of variables $y=2\sqrt{c}z$ is performed next to make the numerator $\sqrt{1-z^2}$

$$F(x) \;=\; \frac{\sqrt{c}}{\pi(c\wedge 1)}\int_{-1}^{[x^2-(1+c)]/2\sqrt{c}} \frac{\sqrt{1-z^2}}{z+(1+c)/2\sqrt{c}}dz$$

Next, the following formula is applied to find the closed form of $F(x)$ by substituting $z=[x^2-(1+c)]/2\sqrt{c}$ and $q=(1+c)/2\sqrt{c}$

$$\int \frac{\sqrt{1-z^2}}{z+q}dw \;=\; \sqrt{1-z^2}-\sqrt{q^2-1}\arctan\left(\frac{qz+1}{\sqrt{(q^2-1)(1-z^2)}}\right)+q\arcsin(z)$$

The final expression can be simplified using $\sqrt{q^2-1}=\sqrt{[(1+c)/2\sqrt{c}]^2-1}=|1-c|/2\sqrt{c}$.

1.8.2 Limit theorems for asymptotic matrix reconstruction problem

Propositions 9 and 10 in Section 1.5 provide an asymptotic connection between the eigenvalues and eigenvectors of the target matrix A and those of the observed matrix Y. Each proposition is derived from recent work in random matrix theory on spiked population models. Spiked population models were introduced by Johnstone (2001).

The Spiked Population Model

The spiked population model is formally defined as follows. Let $r\geq 1$ and constants $\tau_1\geq\cdots\geq\tau_r>1$ be given, and for $n\geq 1$ let integers $m=m(n)$ be defined in such a way that

$$\frac{m}{n}\;\to\; c\;>\;0\quad\text{as}\quad n\to\infty. \tag{1.18}$$

For each n let

$$T=\mathrm{diag}(\tau_1,\ldots,\tau_r,1,\ldots,1)$$

be an $m\times m$ diagonal matrix (with $m=m(n)$), and let X be an $m\times n$ matrix with independent $N_m(0,T)$ columns. Let $\widehat{T}=n^{-1}XX'$ be the sample covariance matrix of X.

The matrix X appearing in the spiked population model may be decomposed as a sum of matrices that parallel those in the matrix reconstruction problem. In particular, X can be

represented as a sum

$$X \;=\; X_1 + Z, \tag{1.19}$$

where X_1 has independent $N_m(0, T - I)$ columns, Z has independent $N(0, 1)$ entries, and X_1 and Z are independent. It follows from the definition of T that

$$(T - I) \;=\; \mathrm{diag}(\tau_1 - 1, \ldots, \tau_r - 1, 0, ..., 0),$$

and therefore the entries in rows $r + 1, \ldots, m$ of X_1 are equal to zero. Thus, the sample covariance matrix $\widehat{T}_1 = n^{-1} X_1 X_1'$ of X_1 has the simple block form

$$\widehat{T}_1 \;=\; \left[\begin{array}{c|c} \widehat{T}_{11} & 0 \\ \hline 0 & 0 \end{array}\right]$$

where $\widehat{T}_{11}$ is an $r \times r$ matrix equal to the sample covariance of the first r rows of X_1. It is clear from the block structure that the first r eigenvalues of $\widehat{T}_1$ are equal to the eigenvalues of $\widehat{T}_{11}$, and that the remaining $(m - r)$ eigenvalues of $\widehat{T}_1$ are equal to zero. The size of $\widehat{T}_{11}$ is fixed, and therefore as n tends to infinity, its entries converge in probability to those of $\mathrm{diag}(\tau_1 - 1, \ldots, \tau_r - 1)$, thus

$$\big\| \frac{1}{n} X_1 X_1' - (T - I) \big\|_F^2 \;\xrightarrow{P}\; 0. \tag{1.20}$$

Consequently, for each $1 \le j \le r$, as n tends to infinity

$$\lambda_j^2(n^{-1/2} X_1) \;=\; \lambda_j(\widehat{T}_1) \;=\; \lambda_j(\widehat{T}_{11}) \;\xrightarrow{P}\; \tau_j - 1 \tag{1.21}$$

and

$$\big\langle u_j(\widehat{T}_{11}), e_j \big\rangle^2 \;\xrightarrow{P}\; 1, \tag{1.22}$$

where e_j is the j-th canonical basis element in $\mathbb{R}^r$. An easy argument shows that $u_j(n^{-1/2} X_1) = u_j(\widehat{T}_1)$, and it then follows from (1.22) that

$$\big\langle u_j(n^{-1/2} X_1), e_j \big\rangle^2 \;\xrightarrow{P}\; 1, \tag{1.23}$$

where e_j is the j-th canonical basis element in $\mathbb{R}^m$.

Proof of Proposition 9

Proposition 9 is derived from existing results on the limiting singular values of $\widehat{T}$ in the spiked population model. These results are summarized in the following theorem, which is a combination of Theorems 1.1, 1.2 and 1.3 in Baik & Silverstein (2006).

Theorem A. *If $\widehat{T}$ is derived from the spiked population model with parameters $\tau_1, \ldots, \tau_r > 1$, then for $1 \leq j \leq r$,*

$$\lambda_j(\widehat{T}) \xrightarrow{P} \begin{cases} \tau_j + c\frac{\tau_j}{\tau_j - 1} & \text{if} \quad \tau_j > 1 + \sqrt{c} \\ (1+\sqrt{c})^2 & \text{if} \quad 1 < \tau_j \leq 1 + \sqrt{c} \end{cases}$$

as n tends to infinity. The remaining sample eigenvalues $\lambda_{r+1}(\widehat{T}), \ldots, \lambda_{m \wedge n}(\widehat{T})$ are associated with the unit eigenvalues of T. Their empirical distribution converges weakly to the Marčenko-Pastur distribution.

We also require the following inequality of Mirsky (1960).

Theorem B. *If B and C are $m \times n$ matrices then $\sum_{j=1}^{m \wedge n} [\lambda_j(C) - \lambda_j(B)]^2 \leq \|C - B\|_F^2$.*

Proof of Proposition 9: Fix $n \geq 1$ and let Y follow the asymptotic reconstruction model (1.8), $Y = A + n^{-1/2}W$, where the target matrix A has fixed rank r and non-zero singular values $\lambda_1(A), \ldots, \lambda_r(A)$. Without loss of generality, we will assume that the target matrix $A = \text{diag}(\lambda_1(A), \ldots, \lambda_r(A), 0, \ldots, 0)$.

We begin by considering a spiked population model whose parameters match those of the matrix reconstruction model. Let X have the same dimensions as Y and be derived from a spiked population model with covariance matrix T having r non-unit eigenvalues

$$\tau_j = \lambda_j^2(A) + 1, \quad j = 1, \ldots, r. \tag{1.24}$$

As noted above, we may represent X as $X = X_1 + Z$, where X_1 has independent $N(0, T - I)$ columns, Z has independent $N(0.1)$ entries and X_1 is independent of Z. Recall that the limit results (1.20)-(1.23) hold for this representation.

The matrix reconstruction problem and spiked population model may be coupled in a natural way. Let random orthogonal matrices U_1 and V_1 be defined for each sample point in such a way that $U_1 D_1 V_1'$ is the SVD of X_1. By construction, the matrices U_1, V_1 depend only on X_1, and are therefore independent of Z. Consequently $U_1' Z V_1$ has the same distribution as Z. If we define $\tilde{W} = U_1' Z V_1$, then $\tilde{Y} = A + n^{-1/2}\tilde{W}$ has the same distribution as the observed matrix Y in the matrix reconstruction problem.

We apply Mirsky's theorem with $B = \tilde{Y}$ and $C = n^{-1/2} U_1' X V_1$ in order to bound the difference between the singular values of $\tilde{Y}$ and those of $n^{-1/2}X$:

$$\begin{aligned}
\sum_{j=1}^{m\wedge n} \left[\lambda_j(n^{-1/2}X) - \lambda_j(\tilde{Y})\right]^2 &\leq \left\|n^{-1/2}U_1' X V_1 - \tilde{Y}\right\|_F^2 \\
&= \left\|(n^{-1/2}U_1' X_1 V_1 - A) + n^{-1/2}(U_1' Z V_1 - \tilde{W})\right\|_F^2 \\
&= \left\|n^{-1/2}U_1' X_1 V_1 - A\right\|_F^2 \\
&= \sum_{j=1}^{m\wedge n} \left[\lambda_j(n^{-1/2}U_1' X_1 V_1) - \lambda_j(A)\right]^2 \\
&= \sum_{j=1}^{m\wedge n} \left[\lambda_j(n^{-1/2}X_1) - \lambda_j(A)\right]^2.
\end{aligned}$$

The first inequality follows from Mirsky's theorem and the fact that the singular values of $n^{-1/2}X$ and $n^{-1/2}U_1' X V_1$ are the same, even though U_1 and V_1 may not be independent of X. The next two equalities follow by expanding X and $\tilde{Y}$ and the fact that $\tilde{W} = U_1' Z V_1$. The third equality is a consequence of the fact that both $U_1' X_1 V_1$ and A are diagonal, and the final equality follows from the equality of the singular values of X_1 and $U_1' X_1 V_1$. In conjunction with (1.21) and (1.24), the last display implies that

$$\sum_j \left[\lambda_j(n^{-1/2}X) - \lambda_j(\tilde{Y})\right]^2 \xrightarrow{P} 0.$$

Thus the distributional and limit results for the eigenvalues of $\widehat{T} = n^{-1}XX'$ hold also for the eigenvalues of $\tilde{Y}\tilde{Y}'$, and therefore for YY' as well. The relation $\lambda_j(Y) = \sqrt{\lambda_j(YY')}$ completes the proof. □

Proof of Proposition 10

Proposition 10 may be derived from existing results on the limiting singular vectors of the sample covariance $\widehat{T}$ in the spiked population model. These results are summarized in Theorem C below. The result was first established for Gaussian models and aspect ratios $0 < c < 1$ by Paul (2007). Nadler (2008) extended Paul's results to $c > 0$. Recently Lee et al. (2010) further extended the theorem to $c \geq 0$ and non-Gaussian models.

Theorem C. *If $\widehat{T}$ is derived from the spiked population model with distinct parameters $\tau_1 > \cdots > \tau_r > 1$, then for $1 \leq j \leq r$,*

$$\left\langle u_j(\widehat{T}), u_j(T) \right\rangle^2 \xrightarrow{P} \begin{cases} \left(1 - \frac{c}{(\tau_j - 1)^2}\right) / \left(1 + \frac{c}{\tau_j - 1}\right) & \text{if} \quad \tau_j > 1 + \sqrt{c} \\ 0 & \text{if} \quad 1 < \tau_j \leq 1 + \sqrt{c} \end{cases}$$

Moreover, for $\tau_j > 1 + \sqrt{c}$ and $k \neq j$ such that $1 \leq k \leq r$ we have

$$\left\langle u_j(\widehat{T}), u_k(T) \right\rangle^2 \xrightarrow{P} 0.$$

Although the last result is not explicitly stated in Paul (2007), it follows immediately from the central limit theorem for eigenvectors (Theorem 5 in Paul).

We also require the following result, which is a special case of an inequality of Wedin (Wedin 1972, Stewart 1991).

Theorem D. *Let B and C be $m \times n$ matrices and let $1 \leq j \leq m \wedge n$. If the j-th singular value of C is separated from the singular values of B and bounded away from zero, in the sense that*

$$\min_{k \neq j} \left| \lambda_j(C) - \lambda_k(B) \right| > \delta \qquad \text{and} \qquad \lambda_j(C) > \delta$$

for some $\delta > 0$, then

$$\left\langle u_j(B), u_j(C) \right\rangle^2 + \left\langle v_j(B), v_j(C) \right\rangle^2 \geq 2 - \frac{2\|B - C\|_F^2}{\delta^2}.$$

Proof of Proposition 10: Fix $n \geq 1$ and let Y follow the asymptotic reconstruction model (1.8),

$Y = A + n^{-1/2}W$, where the target matrix A has fixed rank r and non-zero singular values $\lambda_1(A), \ldots, \lambda_r(A)$. Without loss of generality, we will assume that the target matrix $A = \mathrm{diag}(\lambda_1(A), \ldots, \lambda_r(A), 0, \ldots, 0)$.

We consider a spiked population model with parameters matching those of the matrix reconstruction problem and couple it with the matrix reconstruction model exactly as in the proof of Proposition 9. Please refer to the proof of Proposition 9 for the definition of τ_j and matrices $T, X, X_1, Z, U_1, V_1, \tilde{W}$, and $\tilde{Y}$.

For the rest of the proof fix j such that $\lambda_j(A) > \sqrt[4]{c}$. We apply Wedin's theorem with $B = \tilde{Y}$ and $C = n^{-1/2}U_1'XV_1$. There is $\delta > 0$ such that both conditions of Wedin's theorem are satisfied for the given j with probability converging to 1 as $n \to \infty$. The precise choice of δ is presented at the end of this proof. It follows from Wedin's theorem that

$$\left\langle u_j(\tilde{Y}), u_j(n^{-1/2}U_1'XV_1)\right\rangle^2 = \left\langle u_j(B), u_j(C)\right\rangle^2 \geq 1 - \frac{2\|B-C\|_F^2}{\delta^2}$$

In the proof of Proposition 9 we have shown that $\|B - C\|_F^2 = \|n^{-1/2}U_1'XV_1 - \tilde{Y}\|_F^2 \xrightarrow{P} 0$ as $n \to \infty$. Thus, substituting $u_j(n^{-1/2}U_1'XV) = U_1'u_j(X)$ we get

$$\left\langle u_j(\tilde{Y}), U_1'u_j(X)\right\rangle^2 \xrightarrow{P} 1. \tag{1.25}$$

Recall that we fixed j such that $\tau_j > 1 + \sqrt{c}$. Fix $1 \leq k \leq r$. Theorem C states that $\langle u_j(\widehat{T}), e_k\rangle^2$ has a non-random limit in probability which we will denote as θ_{jk}^2. It follows from connection $\tau_j = \lambda_j^2(A) + 1$ that $\theta_{jk}^2 = [1 - c\lambda_j^{-4}(A)]/[1 + c\lambda_j^{-2}(A)]$ if $j = k$ and $\theta_{jk}^2 = 0$ otherwise. As $u_j(\widehat{T}) = u_j(X)$, it follows that

$$\left\langle u_j(X), e_k\right\rangle^2 \xrightarrow{P} \theta_{jk}^2.$$

Recall that the matrix U_1 consists of the left singular vectors of X_1, namely $u_k(X_1) = U_1e_k$. In (1.23) we have established that $\langle U_1e_k, e_k\rangle^2 = \langle u_k(X_1), e_k\rangle^2 \xrightarrow{P} 1$, so we can replace e_k with U_1e_k in the equation above, to get

$$\left\langle u_j(X), U_1e_k\right\rangle^2 \xrightarrow{P} \theta_{jk}^2.$$

Using the basic properties of inner products we move U_1 to the left part of the inner product:

$$\left\langle U_1' u_j(X), e_k \right\rangle^2 \xrightarrow{P} \theta_{jk}^2.$$

Now, using (1.25) we replace the left term in the inner product by $u_j(\tilde{Y})$, so that

$$\left\langle u_j(\tilde{Y}), e_k \right\rangle^2 \xrightarrow{P} \theta_{jk}^2.$$

As $A = \operatorname{diag}(\lambda_1(A), \ldots, \lambda_r(A), 0, \ldots, 0)$ we have $e_k = u_k(A)$. By construction matrix $\tilde{Y}$ has the same distribution as Y, so it follows from the last display equation that

$$\left\langle u_j(Y), u_k(A) \right\rangle^2 \xrightarrow{P} \theta_{jk}^2.$$

This equation is exactly the statement we sought to proof.

Now, we choose $\delta > 0$ and establish that for any j such that $\lambda_j(A) > \sqrt[4]{c}$ the conditions of Wedin's theorem are satisfied with probability going to 1. It follows from Proposition 9 that for $k = 1, \ldots, r$ the k-th singular value of Y has a non-random limit in probability

$$\lambda_k^* \;=\; \lim \lambda_k(n^{-1/2} X) \;=\; \lim \lambda_k(\tilde{Y}).$$

Let r_0 be the number of eigenvalues of A such that $\lambda_j(A) > \sqrt[4]{c}$ (i.e. the inequality holds only for $j = 1, \ldots, r_0$). It follows from the formula for λ_k^* that $\lambda_k^* > 1 + \sqrt{c}$ for $k = 1, \ldots, r_0$. Note also that in this case λ_k^* is a strictly increasing function of $\lambda_k(A)$. All non-zero $\lambda_j(A)$ are distinct by assumption, so all λ_k^* are distinct for $k = 1, \ldots, r_0$. Note that $\lambda_{r_0+1}^* = 1 + \sqrt{c}$ is smaller that $\lambda_{r_0}^*$. Thus the limits of the first r_0 singular values of Y are not only distinct, they are bounded away from all other singular values. Define

$$\delta \;=\; \frac{1}{3} \min_{k=1,\ldots,r_0} (\lambda_k^* - \lambda_{k+1}^*) \;>\; 0.$$

For any $k = 1, \ldots, r_0 + 1$ the following inequalities are satisfied with probability going to 1 as

$n \to \infty$

$$|\lambda_k(Y) - \lambda_k^*| < \delta \quad \text{and} \quad |\lambda_k(n^{-1/2}X) - \lambda_k^*| < \delta. \tag{1.26}$$

In applying Wedin's theorem to $B = \tilde{Y}$ and $C = n^{-1/2}U_1'XV_1$ we must verify that for any $j = 1, \ldots, r_0$ its two conditions are satisfied with probability going to 1. The first condition is $\lambda_j(C) > \delta$. When inequalities (1.26) hold

$$\lambda_j(C) = \lambda_j(n^{-1/2}U_1'XV_1) = \lambda_j(n^{-1/2}X) > \lambda_j^* - \delta > (\lambda_j^* - \lambda_{j+1}^*) - \delta > 3\delta - \delta = 2\delta,$$

so the first condition is satisfied with probability going to 1. The second condition is $|\lambda_j(C) - \lambda_k(B)| > \delta$ for all $k \neq j$. It is sufficient to check the condition for $k = 1, \ldots, r_0 + 1$ as asymptotically $\lambda_j(C) > \lambda_{r_0+1}(B)$. From the definition of δ and the triangle inequality we get

$$3\delta < |\lambda_j^* - \lambda_k^*| \leq |\lambda_j^* - \lambda_j(n^{-1/2}X)| + |\lambda_j(n^{-1/2}X) - \lambda_k(\tilde{Y})| + |\lambda_k(\tilde{Y}) - \lambda_j^*|.$$

When inequalities (1.26) hold the first and the last terms on the right hand side sum are no larger than δ, thus

$$3\delta < \delta + |\lambda_j(n^{-1/2}X) - \lambda_k(\tilde{Y})| + \delta.$$

It follows that the second condition $|\lambda_j(C) - \lambda_k(B)| = |\lambda_j(n^{-1/2}X) - \lambda_k(\tilde{Y})| > \delta$ also holds with probability going to 1. □

CHAPTER 2

Finding Large Average Submatrices in High Dimensional Data

Summary

The search for sample-variable associations is an important problem in the exploratory analysis of high dimensional data. Biclustering methods search for sample-variable associations in the form of distinguished submatrices of the data matrix. (The rows and columns of a submatrix need not be contiguous.) In this chapter we propose and evaluate a statistically motivated biclustering procedure (LAS) that finds large average submatrices within a given real-valued data matrix. The procedure operates in an iterative-residual fashion, and is driven by a Bonferroni-based significance score that effectively trades off between submatrix size and average value. We examine the performance and potential utility of LAS, and compare it with a number of existing methods, through an extensive three-part validation study using two gene expression datasets. The validation study examines quantitative properties of biclusters, biological and clinical assessments using auxiliary information, and classification of disease subtypes using bicluster membership. In addition, we carry out a simulation study to assess the effectiveness and noise sensitivity of the LAS search procedure. These results suggest that LAS is an effective exploratory tool for the discovery of biologically relevant structures in high dimensional data.

Software is available at https://genome.unc.edu/las/.

2.1 Introduction

Unsupervised exploratory analysis plays an important role in the study of large, high-dimensional datasets that arise in a variety of applications, including gene expression microarrays. Broadly

speaking, the goal of such analysis is to find patterns or regularities in the data, without *ab initio* reference to external information about the available samples and variables. One important source of regularity in experimental data are associations between sets of samples and sets of variables. These associations correspond to distinguished submatrices of the data matrix, and are generally referred to as biclusters, or subspace clusters. In gene expression and related analyses, biclusters, in conjunction with auxiliary clinical and biological information, can provide a first step in the process of identifying disease subtypes and gene regulatory networks.

In this chapter we propose and evaluate a statistically motivated biclustering procedure that finds large average submatrices within a given real-valued data matrix. The procedure, which is called LAS (for Large Average Submatrix), operates in an iterative fashion, and is based on a simple significance score that trades off between the size of a submatrix and its average value. A connection is established between maximization of the significance score and the minimum description length principle.

We examine the performance and utility of LAS, and compare it with a number of existing methods, through an extensive validation study using two independent gene expression datasets. The validation study has three parts. The first concerns quantitative properties of the biclustering methods such as bicluster size, overlap and coordinate-wise statistics. The second is focused biological and clinical assessments using auxiliary information about the samples and genes under study. In the the third part of the study, the biclusters are used to perform classification of disease subtypes based in their sample membership. In addition, we carry out a simulation study to assess the effectiveness and noise sensitivity of the LAS search procedure.

2.1.1 Biclustering

Sample-variable associations can be defined in a variety of ways, and can take a variety of forms. The simplest, and most common, way of identifying associations in gene expression data is to independently cluster the rows and columns of the data matrix using a multivariate clustering procedure [Weinstein et al. (1997), Eisen et al. (1998), Tamayo et al. (1999), Hastie et al. (2000)]. When the rows and columns of the data matrix are reordered so that each cluster forms a contiguous group, the result is a partition of the data matrix into nonoverlapping rectangular cells. The search for sample variable associations then consists of identifying cells whose entries

are, on average, bright red (large and positive) or bright green (large and negative) [Weigelt et al. (2005)]. In some cases, one can improve the results of independent row-column clustering by simultaneously clustering samples and variables, a procedure known as co-clustering [Hartigan (1972), Kluger et al. (2003), Dhillon (2001), Getz et al. (2000)].

Independent row-column clustering (IRCC) has become a standard tool for the visualization and exploratory analysis of microarray data, but it is an indirect approach to the problem of finding sample-variable associations. By contrast, biclustering methods search directly for sample-variable associations, or more precisely, for submatrices U of the data matrix X whose entries meet a predefined criterion. Submatrices meeting the criterion are typically referred to as biclusters. It is important to note that the rows and columns of a bicluster (and more generally a submatrix) need not be contiguous. A number of criteria for defining biclusters U have been considered in the literature, for example: the rows of U are approximately equal to each other [Aggarwal et al. (1999)]; the columns of U are approximately equal [Friedman & Meulman (2004)]; the elements of U are well-fit by a 2-way ANOVA model [Cheng & Church (2000), Lazzeroni & Owen (2002), Wang et al. (2002)]; the rows of U have equal [Ben-Dor et al. (2003)] or approximately equal [Liu et al. (2004)] rank statistics; and all elements of U are above a given threshold [Prelic et al. (2006)].

The focus of this chapter is the simple criterion that the average of the entries of the submatrix U is large and positive, or large and negative. Submatrices of this sort will appear red or green in the standard heat map representation of the data matrix, and are similar to those targeted by independent row-column clustering.

2.1.2 Features of Biclustering

While its direct focus on finding sample-variable associations makes biclustering an attractive alternative to row-column clustering, biclustering has a number of other features, both positive and negative, that we briefly discuss below.

Row-column clustering assigns each sample and each variable to a unique cluster. By contrast, the submatrices produced by biclustering methods may overlap, and need not cover the entire data matrix, features that better reflect the structure of many scientific problems. For example, the same gene can play a role in multiple pathways, and a single sample may

belong to multiple phenotypic or genotypic subtypes. Multiple bicluster membership for rows and columns can directly capture this aspect of experimental data.

In row-column clustering, the group to which a sample is assigned depends on all the available variables, and the group to which a variable is assigned depends on the all the available samples. By contrast, biclusters are locally defined: the inclusion of samples and variables in a bicluster depends only on their expression values inside the associated submatrix. Locality allows biclustering methods to target relevant genes and samples while ignoring others, giving such methods greater exploratory power and flexibility than row-column clustering. For more on the potential advantages of biclustering, see Madeira & Oliveira (2004), Jiang et al. (2004), Parsons et al. (2004).

Figure 2.1 illustrates the differences between the blocks arising from independent row-column clustering and those arising from biclustering. Note that while one may display an individual bicluster as a contiguous block of variables and samples by suitably reordering the rows and columns of the data matrix, when considering more than two biclusters, it is not always possible to display them simultaneously as contiguous blocks.

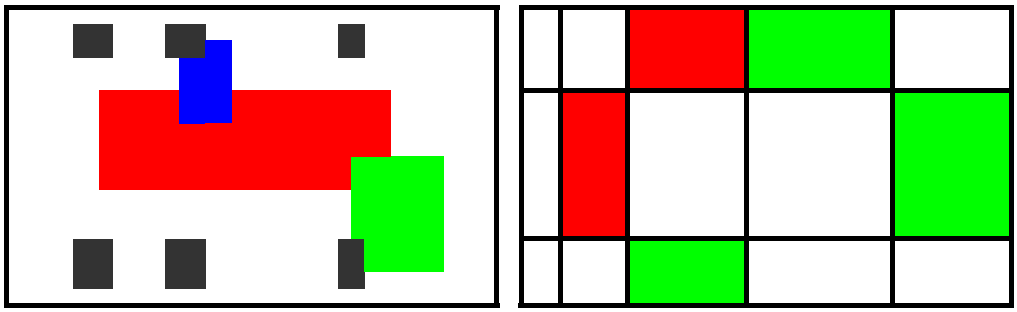

Figure 2.1: Illustration of bicluster overlap (left) and row-column clustering (right).

The flexibility and exploratory power of biclustering methods comes at the cost of increased computational complexity. Most biclustering problems are NP complete, and even the most efficient exact algorithms (those that search for every maximal submatrix satisfying a given criterion) can be prohibitively slow, and produce a large number of biclusters, when they are applied to large datasets. The LAS algorithm relies on a heuristic (nonexact), randomized search to find biclusters, a feature shared by many existing methods.

2.2 The LAS algorithm

In this chapter we present and assess a significance-based approach to biclustering of real-valued data. Using a simple Gaussian null model for the observed data, we assign a significance score to each submatrix U of the data matrix using a Bonferroni-corrected p-value that is based on the size and average value of the entries of U. The Bonferroni correction accounts for multiple comparisons that arise when searching among many submatrices for a submatrix having a large average value. In addition, the correction acts as a penalty that controls the size of discovered submatrices. The connections between LAS and the Minimum Description Length principle are discussed in Section 2.2.3 below.

2.2.1 Basic Model and Score Function

Let $X = \{x_{i,j} : i \in [m], j \in [n]\}$ be the observed data matrix. (Here and in what follows, $[k]$ denotes the set of integers from 1 to k.) A submatrix of X is an indexed set of entries $U = \{x_{i,j} : i \in A, j \in B\}$ associated with a specified set of rows $A \subseteq [m]$ and columns $B \subseteq [n]$. In general, the rows in A and the columns in B need not be contiguous.

The LAS algorithm is motivated by an additive submatrix model under which the data matrix X is expressed as the sum of K constant, and potentially overlapping, submatrices plus noise. More precisely, the model states that

$$x_{i,j} = \sum_{k=1}^{K} \alpha_k \, I(i \in A_k, j \in B_k) + \varepsilon_{ij}, \qquad i \in [m], \; j \in [n], \tag{2.1}$$

where $A_k \subseteq [m]$ and $B_k \subseteq [n]$ are the row and column sets of the kth submatrix, $\alpha_k \in \mathbb{R}$ is the level of the kth submatrix, and $\{\varepsilon_{ij}\}$ are independent $N(0,1)$ random variables. Here $I(\cdot)$ is an indicator function equal to one when the condition in parentheses holds. When $K = 0$, the model (2.1) reduces to the simple null model

$$\{x_{i,j} : i \in [m], j \in [n]\} \;\text{ are i.i.d with } x_{i,j} \sim \mathcal{N}(0,1) \tag{2.2}$$

under which X is an $m \times n$ Gaussian random matrix.

The null model (2.2) leads naturally to a significance based score function for submatrices.

In particular, the score assigned to a $k \times l$ submatrix U of X with average $\text{Avg}(U) = \tau > 0$ is defined by

$$S(U) \;=\; -\log\left[\binom{m}{k}\binom{n}{l}\Phi\left(-\tau\sqrt{kl}\right)\right]. \tag{2.3}$$

The term in square brackets is an upper bound on the probability of the event A that there exists a $k \times l$ submatrix with average greater than or equal to τ in an $m \times n$ Gaussian random matrix. More precisely, by the union bound, $P(A) \leq \sum P(\text{Avg}(V) \geq \tau)$, where the sum ranges over all $k \times l$ submatrices V of X. Under the Gaussian null, each probability in the sum is equal to $\Phi(-\tau\sqrt{kl})$, so that $P(A) \leq N\,\Phi(-\tau\sqrt{kl})$, where $N = \binom{m}{k}\binom{n}{l}$ is the number of $k \times l$ submatrices of an $m \times n$ matrix. From a testing point of view, the term in brackets can be thought of as a Bonferroni corrected p-value associated with the null model (2.2) and the test function $\text{Avg}(U)$.

The score function $S(\cdot)$ measures departures from the null (2.2) in a manner that accounts for the dimensions and average value of a submatrix. It provides a simple, one-dimensional yardstick with which one can compare and rank observed submatrices of different sizes and intensities. Among submatrices of the same dimensions, it favors those with higher average.

2.2.2 Description of Algorithm

The LAS score function is based on the normal CDF, and is sensitive to departures from normality that arise from heavy tails in the empirical distribution of the expression values. Outliers can give rise to submatrices that, while highly significant, have very few samples or variables. As a first step in the algorithm we consider the standard Q-Q plot of the empirical distribution of the entries of the column-standardized data matrix against the standard normal CDF. Both the breast cancer and lung cancer datasets considered in Section 2.4 exhibited heavy tails. To address this, we applied the transformation $f(x) = \text{sign}(x)\log(1 + |x|)$ to each entry of the data matrix. After transformation, the Q-Q plot indicated excellent agreement with the normal distribution. Other alternatives to the squashing function $f()$ can also be considered.

The LAS algorithm initially searches for a submatrix of X maximizing the significance score $S(\cdot)$. Once a candidate submatrix U^* has been identified, a residual matrix X' is computed by subtracting the average of U^* from each of its elements in X. Formally, $x'_{i,j} = x_{i,j} - \text{Avg}(U^*)$ if

$x_{i,j}$ is in U^*, and $x'_{i,j} = x_{i,j}$ otherwise. The search procedure is then repeated on the residual matrix X'. The core of the algorithm is a randomly initialized, iterative search procedure for finding a maximally significant submatrix of a given matrix. The pseudo code for the algorithm is as follows:

Input: Data matrix X

Search: Find a submatrix U^* of X that approximately maximizes the score function $S(\cdot)$.

Residual: Subtract the average of U^* from each of its elements in X.

Repeat: Return to Search.

Stop: When $S(U^*)$ falls below a threshold, or a user-defined number of submatrices are produced.

The output of the algorithm is a collection of submatrices having significant positive averages. Repeating the algorithm with X replaced by $-X$ yields submatrices with significant negative averages.

It is not feasible in the search procedure to check the score of each of the 2^{n+m} possible submatrices of X. Instead, the procedure iteratively updates the row and column sets of a candidate submatrix in a greedy fashion until a local maximum of the score function is achieved. For fixed k, l, the basic search procedure operates as follows:

Initialize: Select l columns of B at random.

Loop: Iterate until convergence of A, B

Let $A := k$ rows with the largest sum over the columns in B.

Let $B := l$ columns with the largest sum over the rows in A.

Output: Submatrix associated with final A, B.

As currently implemented, the initial values of k and l are selected at random from the sets $\{1, \ldots, \lceil m/2 \rceil\}$ and $\{1, \ldots, \lceil n/2 \rceil\}$ respectively, and are held fixed until the algorithm finds a local maximum of the score function. On subsequent iterations, the algorithm adaptively selects the number of rows and columns in order to maximize the significance score. Each run of the

basic search procedure yields a submatrix that is a local maximum of the score function, *that is*, a submatrix that cannot be improved by changing only its column set or its row set. The basic search procedure is repeated 1000 times, and the most significant submatrix found is returned in the main loop of the algorithm. In experiments on real data (see Section 2.5.3), we found that 1000 iterations of the main loop of the search procedure was sufficient to ensure stable performance of the algorithm.

Many biclustering methods require the user to specify a number of operational parameters, and in many cases, getting optimal performance from the method can require careful choice and tuning of the parameters. In addition, for exact algorithms, minor alteration of the parameters can result in substantial changes in the size and interpretability of the output. The only operational parameters of the LAS algorithm are the number of times the basic search procedure is run in each main loop of the algorithm, and the stopping criterion. This minimal number of parameters is an important feature of LAS, one that makes application of the method to scientific problems relatively straightforward.

2.2.3 Penalization and MDL

The score function employed by LAS can be written as a sum of two terms. The first, $-\log \Phi(-\sqrt{kl}\tau)$, is positive and can be viewed as a "reward" for finding a $k \times l$ submatrix with average τ. The second, $-\log[\binom{m}{k}\binom{n}{l}]$, is negative and is a multiple comparisons penalty based on the number of $k \times l$ submatrices in X. The penalty depends separately on k and l, and its combinatorial form suggests a connection with the Minimum Description Length Principle (MDL), following Rissanen, Grunwald (2004), and Barron & Yu (1998). The MDL principle is a formalization of Occam's Razor, in which the best model for a given set of data is the one that leads to the shortest overall description of the data.

In Section 2.6 we describe a code for matrices based on a family of additive models, and show that the description length of a matrix with an elevated submatrix is approximately equal to a linear function of its LAS score. The penalty term in the LAS score function corresponds to the length of the code required to describe the location of a $k \times l$ submatrix, while the "reward" is related to the reduction in code length achieved by describing the residual matrix instead of the original matrix. The connection with MDL provides support for the significance based

approach to biclustering adopted here.

2.3 Description of Competing Methods

In this section we describe the methods to which we will compare the LAS algorithm in the validation sections below. We considered biclustering methods that search directly for sample variable associations, as well as biclusters derived from independent row-column clustering.

2.3.1 Biclustering Methods

Initially, we compared LAS with six existing biclustering methods: Plaid, CC, SAMBA, ISA, OPSM, and BiMax. These methods employ a variety of objective functions and search algorithms. We limited our comparisons to methods that (i) have publicly available implementations with straightforward user interfaces, (ii) can efficiently handle large datasets arising from gene expression and metabolomic data, and (iii) are well suited to use by biologists. The methods are described in more detail below.

The Plaid algorithm of Lazzeroni & Owen (2002) employs an iterative procedure to approximate the data matrix X by a sum of submatrices whose entries follow a two-way ANOVA model. At each stage, Plaid searches for a submatrix maximizing explained variation, as measured by reduction in the overall sum of squares. We set the parameters of Plaid so that at each stage it fits a constant submatrix (with no row or column effects). With these settings, the Plaid method is most closely related to LAS, and also derives from a block-additive model like (2.1). We have also run Plaid with settings under which it fits biclusters by a general ANOVA model. The two versions of Plaid exhibit similar validation results; we present only those for which Plaid fits biclusters by a constant. Various modifications of the Plaid model and algorithm have been proposed in the literature: Turner, Bailey & Krzanowski (2005) have developed an improved algorithm and Segal et al. (2003), Gu & Liu (2008), and Caldas & Kaski (2008) have considered the Plaid problem in the Bayesian framework. We have chosen to focus on the original Plaid algorithm of Lazzeroni & Owen, as it is both the first and most representative method of its type.

The Cheng and Church (CC) biclustering algorithm [Cheng & Church (2000)] searches for submatrices such that the sum of squared residuals from a two-way ANOVA fit falls below a

given threshold. These biclusters are locally maximal in the sense that addition of any more rows or columns will increase the mean squared error beyond the threshold. Whereas Plaid searches for a submatrix maximizing explained variation, CC searches for large submatrices with small unexplained variation. The LAS, Plaid, and CC algorithms discover biclusters sequentially. Once a candidate target is identified, LAS and Plaid form the associated residual matrix before proceeding to the next stage. By contrast, CC replaces the values of the target submatrix by Gaussian noise.

The SAMBA algorithm of Tanay et al. (2002) adopts a graph theoretic approach, in which the data matrix is organized into a bipartite graph, with one set of nodes corresponding to genes and the other corresponding to samples. Weights are then assigned to edges that connect genes and samples based on the data matrix, and the subgraphs with the largest overall weights are returned.

Ihmels et al. (2002) developed a biclustering algorithm (ISA) that searches for maximal submatrices whose row and column averages exceed preset thresholds. Both LAS and ISA rely on iterative search procedures that are variants of EM and Gibbs type algorithms. In both methods, the search procedure alternately updates the columnset (given the current rowset) and then the rowset (given the current columnset) until converging to a local optimum.

The OPSM algorithm Ben-Dor et al. (2003) searches for maximal submatrices whose rows have the same order statistics. Like LAS, the OPSM algorithm makes use of a multiple comparison corrected p-value in assessing and comparing biclusters of different sizes.

Each of the algorithms above employs heuristic strategies to search for distinguished submatrices. By contrast, the Bimax algorithm of Prelic et al. (2006) uses a divide-and-conquer approach to find *all* inclusion-maximal biclusters whose values are above a user-defined threshold. Bimax is the only exact algorithm among those considered here.

We ran all methods except Plaid and CC with their default parameter settings. LAS, CC, and Plaid allow the user to choose the number of biclusters produced; we selected 60 biclusters for each method. The settings of Plaid were chosen so that the submatrix fit at each stage is a constant, without row and column effects. Once the CC method identifies a bicluster, it removes it from the data matrix by replacing its elements by noise. When the CC method was run with the default parameter $\delta = 0.5$, it initially produced a single bicluster that contained

most of the available genes and samples, leaving very little information from which additional biclusters could be identified. To solve this problem, we reduced the δ parameter in CC from 0.5 to 0.1.

2.3.2 Running Configurations for Other Methods

All biclustering methods described in this chapter were run on the same machine: AMD64 FX2 2.8GHz, 4GB RAM, running Windows XP Professional (64 bit). The same imputed dataset as run through LAS was loaded into the other programs. If a method was written in Java, the 'Xmx1024m' key was added to the command line for proper memory allocation. In all cases, we preferred to use the default running parameters as given by the software used to run the algorithms (*BicAT* for BiMax, CC, ISA, OPSM, and *Expander* for SAMBA).

Running parameters. **Plaid**, as it is scripting based, a script was written to iterate over the steps $findm$, accept, shuffle 60 times, to produce 60 biclusters. **Cheng-Church**: $seed = 13$, $\Delta = 0.1$, $\alpha = 1.2$, $NumberOutput = 30$, **ISA**: $seed = 13$, $t\_g = 2$, $t\_c = 2$, $StartingNum = 100$, **OPSM**: $PassedModels = 10$, **BiMax**: $Gene_{min} = 10$, $Sample_{min} = 5$, **SAMBA**: try covering all probes, $OptionFiles = valsp\_3ap$, $OverlapPrior = 0.1$, $ProbesToHash = 100$, $Memory_{max} = 500$, $HashKernel_{min} = 4$, $HashKernel_{max} = 7$. The $OverlapPrior$ value ensures that for each new cluster generated, its elements were 90% unique to any previously discovered bicluster.

2.3.3 Independent Row-Column Clustering (IRCC)

In addition to the methods described above, we also produced biclusters from k-means and hierarchical clustering. We applied k-means clustering independently to the rows and columns of the data matrix, with values of k ranging from 3 to 15. In each case, we produced 30 clusterings and selected the one with the lowest sum of within-cluster sum of squares. The set of $85 \times 117 = 9,945$ submatrices (not all column clusters were unique) obtained from the Cartesian product of the row and column clusters is denoted by KM.

We applied hierarchical clustering independently to the rows and columns of the data matrix using a Pearson correlation based distance and average linkage. All clusters associated with subtrees of the dendrogram were considered, but row clusters with less that 10 rows, and column clusters with less than 8 columns, were discarded. The resulting set of $34 \times 2806 = 95,404$

submatrices obtained from the Cartesian product of the row and column clusters is denoted by HC.

2.4 Comparison and Validation

Existing biclustering methods differ widely in their underlying criteria, as well as the algorithms they employ to identify biclusters that satisfy these criteria. As such, simulations based on the additive submatrix model (1) cannot fairly be used to assess the performance of competing methods that are based on different models and submatrix criteria. For this reason our assessment of LAS relies more heavily on biological validation rather than simulations: the former provides a direct comparison of the methods in terms of their practical utility.

We applied LAS and the biclustering methods described in the previous section to two existing gene expression datasets: a breast cancer study from Hu et al. (2006), and a lung cancer study from Bhattacharjee et al. (2001). The datasets can be downloaded from the University of North Carolina Microarray Database (UMD, `http://genome.unc.edu`) and `http://www.broad.mit.edu/mpr/lung/` respectively. In this section we describe and implement a number of validation measures for assessing and comparing the performance of the biclustering methods under study. The validation results for the breast cancer study are detailed below; the results for the lung cancer data are summarized in Section 2.4.6. The validation measures considered here are applicable to any biclustering method and most gene expression type datasets.

2.4.1 Description of the Hu Data

This dataset is from a previously published breast cancer study of Hu et al. (2006) that was based on 146 Agilent 1Av2 microarrays. Initial filtering and normalization followed the protocol in Hu et al.: genes with intensity less than 30 in the red or green channel were removed; for the remaining genes, red and green channels were combined using the $\log_2$ ratio. The initial log-transformed dataset was row median centered, and missing values were imputed using a k-nearest neighbor algorithm with k = 10. Among the 146 samples, there were 29 pairs of biological replicates in which RNA was prepared from different sections of the same tumor. To avoid giving these samples more weight in the analysis, we removed the replicates, keeping only the primary tumor profiles. After preprocessing, the dataset contained 117 samples and 13,666

genes. In what follows, the dataset will be referred to as **Hu**.

2.4.2 Quantitative Comparisons

LAS, Plaid, and CC were set to produce 60 biclusters. The number of biclusters produced by other methods was determined by their default parameters, with values ranging from 15 (OPSM) to 1977 (BiMax). KM and HC produced 9,945 and 95,404 biclusters, respectively. Table 2.1 shows the number of biclusters produced by each method.

All biclustering methods were run on the same computer. The specifications of the computer and the parameters of each biclustering method are provided in Section 2.3.2. The running time of LAS was 85 minutes; ISA and OPSM finished in about 30 minutes; CC, Plaid, and SAMBA finished in less then 10 minutes. The Bimax algorithm took approximately 5 days. Hierarchical clustering took 2 minutes, while k-means clustering (with $k = 3, ..., 15$ and 30 repeats) took 1 hour 40 minutes. Our primary focus in validation was output quality.

Bicluster sizes

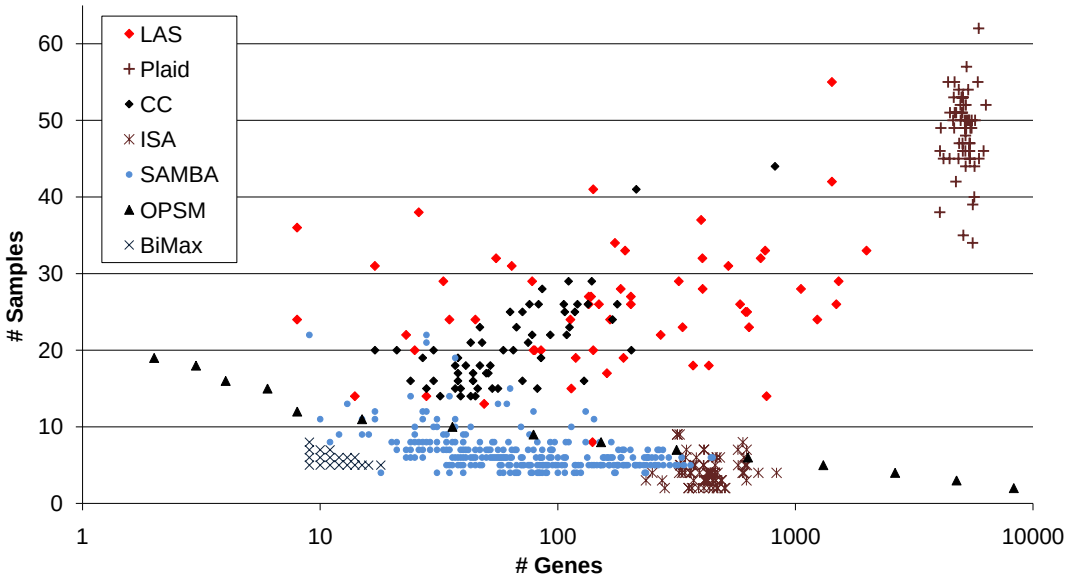

Figure 2.2: Bicluster sizes for different methods.

In Figure 2.2 we plot the row and column dimensions of the biclusters produced by the

different methods. The resulting scatter plot shows marked differences between the methods under study, and provides useful insights into their utility and potential biological findings. (A similar figure could be used, for example, to assess the effects of different parameter settings for a single method of interest.) Both LAS and CC produce a relatively wide range of bicluster sizes, with those of LAS ranging from 8×8 (genes $\times$ samples) to 1991×55. The other methods tested produced biclusters with a more limited range of sizes. Biclusters produced by ISA, OPSM, and SAMBA have a relatively small number of samples, less than 10 samples per bicluster on average in each case. (Some of the points denoting OPSM clusters have been obscured in the figure.) The biclusters produced by Bimax had at most 8 samples, and at most 18 genes. By contrast, Plaid produced large biclusters, having an average of 49 samples and 5130 genes per bicluster.

The differences between LAS and Plaid bear further discussion. We ran Plaid with settings (constant fit, no row, and column effects) that made it most similar to LAS. With these settings, both methods rely on similar models, and proceed in stages via residuals, but differ in their objective functions. Plaid seeks to maximize the explained variation $kl\tau^2$, or equivalently, $-\log \Phi(-\sqrt{kl}\tau)$. By contrast, the score function maximized by LAS includes a combinatorial penalty term involving k and l that acts to control the size of the discovered submatrices. In this, and other, experiments, the penalty excludes very large submatrices, and produces a relatively wide range of bicluster sizes. (While the combinatorial penalty is small for values of k close to m and l close to n, submatrices of this size tend to have a small average value.)

Effective number of biclusters

Distinct biclusters produced by the same method may exhibit overlap. On the one hand, the flexibility of overlapping gene and sample sets has the potential to better capture underlying biology. On the other hand, extreme overlap of biclusters can reduce a method's effective output: two moderate sized biclusters that differ only in a few rows or columns do not provide much more information than either bicluster alone. Whatever the source of overlap, it is helpful to keep it in mind when evaluating other features of a method, such as the number of biclusters it produces that are deemed to be statistically significant. To this end, we measure the effective

number of biclusters in a family $U_1, \ldots, U_K$ by

$$F(U_1, \ldots, U_K) = \sum_{k=1}^{K} \frac{1}{|U_k|} \sum_{x \in U_k} \frac{1}{N(x)},$$

where $N(x) = \sum_{k=1}^{K} I\{x \in U_k\}$ is the number of biclusters containing matrix entry x. The measure $F(\cdot)$ has the property that if, for any $1 \leq r \leq K$, the biclusters $U_1, \ldots, U_K$ can be divided into r nonoverlapping groups of identical biclusters, then $F(U_1, \ldots, U_K) = r$.

Method	# of Clusters	Eff. # of Clusters	Ratio
LAS	60	48.6	0.810
Plaid	60	6.4	0.106
CC	60	60.0	1.000
ISA	72	42.3	0.588
OPSM	15	9.1	0.605
SAMBA	289	171.7	0.594
BiMax	1,977	42.9	0.022
KM	9,945	78.7	0.008
HC	95,404	800.4	0.008

Table 2.1: Output summary for different biclustering methods. From left to right: total number of biclusters produced; effective number of biclusters; the ratio of the effective number to the total number of biclusters.

Table 2.1 shows the effective number of biclusters produced by each method. The low overlap of the CC algorithm is due to the fact that it replaces the values in discovered submatrices by Gaussian noise, so that a matrix element is unlikely to belong to more than one bicluster. Bimax is an exhaustive method with no pre-filtering of its output; it produced a large number of small, highly overlapping biclusters. Biclusters produced by LAS had modest levels of overlap, less than those of all other methods, except CC. The high overlap of Plaid biclusters is explained in part by their large size.

Score-Based comparison of LAS and standard clustering

Ideally, a direct search for large average submatrices should improve on the results of independent row-column clustering. To test this, we computed the significance score $S(C)$ for every cluster produced by KM and HC, and compared these to the scores obtained with LAS. The

highest scores achieved by KM and HC biclusters were 6316 and 5228, respectively. The first LAS biclusters had scores 12883 (positive average) and 10070 (negative average); the scores of the first 6 LAS biclusters were higher than scores of all the biclusters produced by KM or HC. The highest score achieved by a Plaid bicluster was 12542, which also dominated the scores achieved by KM and HC. These results show that LAS is capable, *in practice*, of finding submatrices that cannot be identified by standard clustering methods. We also note that LAS produces only 60 biclusters, while KM and HC produce 9,945 and 95,404 biclusters, respectively.

Summary properties of row and column sets

One potential benefit of biclustering methods over independent row-column clustering is that the sample-variable associations they identify are defined locally: they can, in principle, identify patterns of association that are not readily apparent from the summary statistics across rows and columns that drive k-means and hierarchical clustering. Nevertheless, local associations can sometimes be revealed by summary measures of variance and correlation, and it is worthwhile to consider the value of these quantities as a way of seeing (a) what drives different biclustering methods, and (b) the extent to which the local discoveries of these methods are reflected in more global summaries.

	Correlation		Std. Deviation		Within
	Gene	Sample	Gene	Sample	Variance
Matrix	0.00	0.01	0.89	1.00	1.00
LAS	0.34	0.10	1.40	1.00	1.96
Plaid	0.02	0.03	0.99	1.00	1.24
CC	0.09	0.05	1.02	1.00	0.49
ISA	0.22	0.31	0.99	1.00	1.99
OPSM	0.48	0.06	0.93	1.00	1.18
SAMBA	0.26	0.02	1.66	1.00	3.36
BiMax	0.09	0.26	3.42	1.00	27.75
KM	0.19	0.22	0.91	1.00	0.96
HC	0.44	0.24	0.93	1.00	0.88
Subtypes		0.13		1.00	

Table 2.2: Average standard deviation and average pairwise correlation of genes and samples, for biclusters, KM and HC clusters, and the whole data matrix. As a reference point, the last row shows the summary statistics for samples belonging to the same disease subtype.

For each method under study, the first four columns of Table 2.2 show the average, across

the biclusters, of the following summary statistics: (i) the average pairwise correlation of their constituent genes, (ii) the average pairwise correlation of their constituent samples, (iii) the average standard deviation of their constituent genes, and (iv) the average standard deviation of their constituent samples. Average values for the entire matrix are shown in the first row of the table. (Recall that the data matrix is column standardized, so the column standard deviations are all equal to one.) In each case, the statistics associated with the biclustering methods are higher than the average of these statistics over the entire matrix. As HC is based entirely on gene and sample correlations, we expect its correlation values to be large compared with other methods, and this is the case. The moderate values of gene correlation for KM result from the fact that we are using a relatively small numbers of gene clusters, which tend to have a large number of genes and therefore low average pairwise correlations. Similar remarks apply to the low gene (and sample) correlation values associated with Plaid.

BiMax appears to be driven by all summary measures, with gene correlation playing a relatively minor role, while ISA is not affected by gene standard deviation. LAS appears to be driven by a mix of gene correlation and standard deviation. The average summary statistics of LAS do not appear to be extreme, or to reflect overtly global behavior. In each column, the average for LAS is less than and greater than those of two other methods. The remaining biclustering methods appear to depend on two, or in some cases only one, of the measured summary statistics. We note that the average pairwise correlation of the samples in LAS biclusters best matches the average pairwise correlation of samples in the cancer subtypes (described in Subsection 2.4.3 below).

Tightness of biclusters

For each method under consideration we calculated the average of the within bicluster variances. The results are presented in the rightmost column of Table 2.2. BiMax and SAMBA, which operate on thresholded entries, find biclusters with high average variance. LAS, Plaid, and ISA search for biclusters with high overall or high row/column averages; they find biclusters with variance above one. Biclusters identified by the CC algorithm have the smallest average variance, as CC searches for biclusters with low unexplained variation. The IRCC methods find biclusters with average variance only slightly lower that one.

2.4.3 Biological Comparisons

The previous section compares LAS with other biclustering and IRCC methods on the basis of quantitative measures that are not directly related to biological or clinical features of the data. In this section we consider several biologically motivated comparisons. In particular, we carry out a number of tests that assess the gene and sample sets of each bicluster using auxiliary clinical information and external annotation. The next subsection considers sample-based measures of subtype capture.

Subtype capture

Breast cancer encompasses several distinct diseases, or subtypes, which are characterized by unique and substantially different expression signatures. Each disease subtype has associated biological mechanisms that are connected with its pathologic phenotype and the survival profiles of patients [*cf.* Golub et al. (1999), Sorlie et al. (2001), Weigelt et al. (2005), Hayes et al. (2006)]. Breast cancer subtypes were initially identified using hierarchical clustering of gene expression data, and have subsequently been validated in several datasets [*cf.* Fan et al. (2006)] and across platforms [*cf.* Sorlie et al. (2003)]. They are one focal point for our biological validation.

Hu et al. (2006) assigned each sample in the dataset to one of 5 disease subtypes (Basal-like, HER2-enriched, Luminal A, Luminal B, and Normal-like) using a nearest shrunken centroid predictor [Tibshirani et al. (2002)] and a pre-defined set of 1300 intrinsic genes. The centroids for the predictor were derived from the hierarchical clustering of 300 samples chosen both for data quality and the representative features of their expression profiles. In addition, each sample in the Hu dataset was assigned via a clinical assay to one of two estrogen receptor groups, denoted ER+ and ER-, which constitute the ER status of the tumor. The ER status of tumors is closely related to their subtypes: in the Hu dataset, HER2-enriched and Basal-like samples are primarily ER-negative (74% and 94% respectively), while Normal-like, Luminal A and Luminal B are primarily ER-positive (83%, 86%, and 91% respectively) .

Here we compare the ability of biclustering methods to capture the disease subtype and ER status of the samples. In order to assess how well the set of samples associated with a bicluster captures a particular subtype, we measured the overlap between the two sample groups using a p-value from a standard hypergeometric test (equivalent to a one-sided Fisher's exact test).

For each biclustering method, we identified the bicluster that best matched each subtype, and recorded its associated p-value. As a point of comparison, we include the subtype match of column clusters produced by k-means and hierarchical clustering. The results are shown in Figure 2.3.

The figure indicates that LAS captures ER status and disease subtypes better than the other biclustering methods, with the single exception of the Luminal A subtype, which was better captured by CC. In addition, LAS is competitive with KM and HC, performing better or as well as these methods on the Luminal A, Luminal B, Basal-like, and HER2-enriched subtypes.

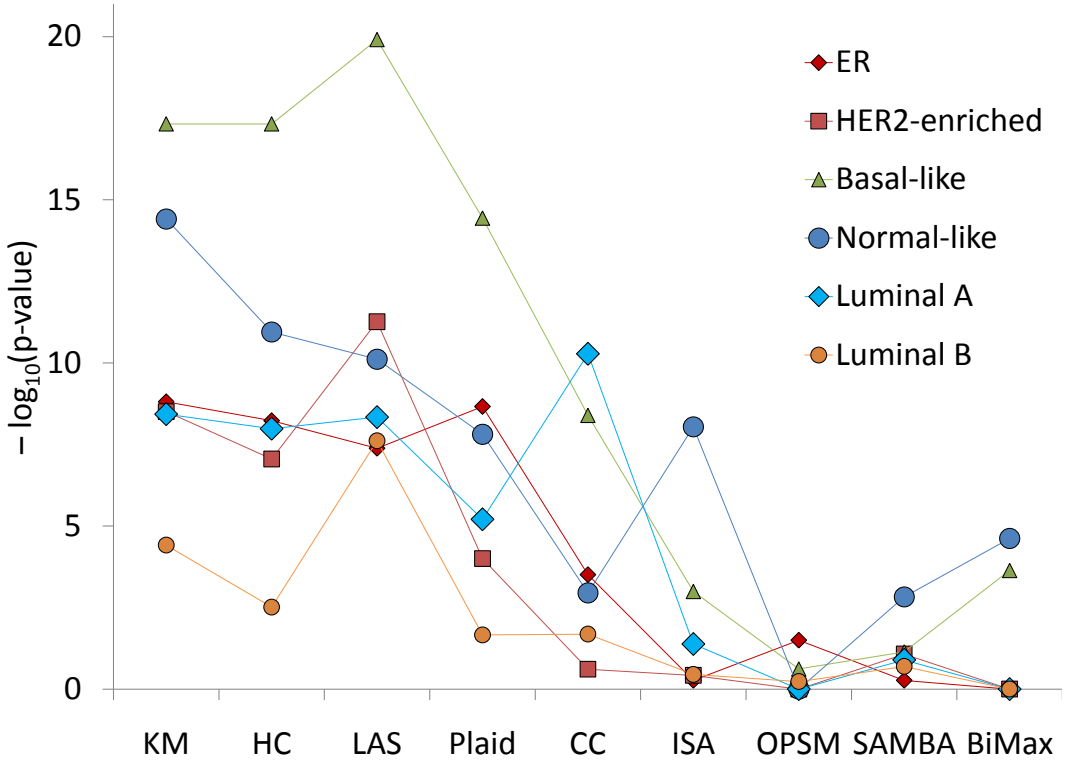

Figure 2.3: The minus $\log_{10}$ p-values of best subtype capture for different biclustering and sample clustering methods.

Another view of subtype capture is presented in the bar-plot of Figure 2.4. For the Basal-like disease subtype, the figure shows the number of true, missed, and false discoveries associated with the the best sample cluster (as judged by the hypergeometric p-value) that was produced by

each method. The Basal-like subtype contains 32 samples. The best LAS bicluster captured 27 of the 32 Basal-like samples with no false positives. Plaid had fewer missed samples, but a larger number of false positives, due to the large size of its sample clusters. As the disease subtypes were identified in part through the use of hierarchical clustering, the strong performance of KM and HC is unsurprising. Other biclustering methods were not successful in capturing Basal-like or other subtypes, due in part to the small number of samples in their biclusters.

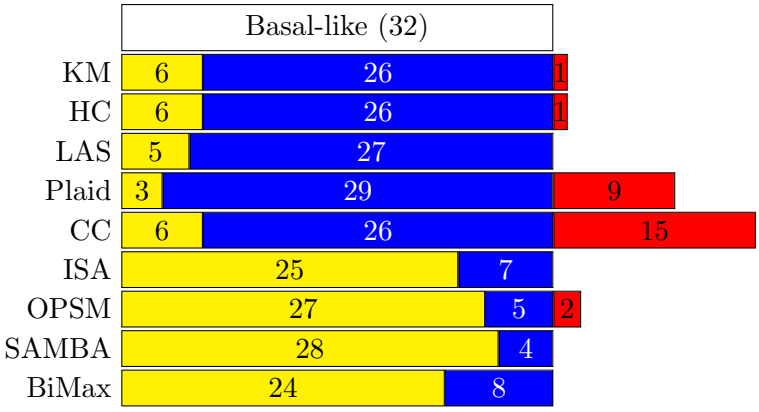

Figure 2.4: Bar-plot of missed, true, and false discoveries for different biclustering methods and the Luminal A subtype. Bars represent: light - missed discoveries, dark - true discoveries, gray - falsely discoveries.

2.4.4 Biclusters of Potential Biological Interest

In order to assess the potential biological and clinical relevance of the biclustering methods under consideration, we applied three different tests to the gene and sample sets of each bicluster. The first test makes use of clinical information concerning patient survival. The second tests for over-representation of functional categories and genomic neighborhoods (cytobands) among the gene sets of different biclusters, and the third tests for the differential expression of these same gene categories between the sample set of a bicluster and its complement. The tests are described in more detail below.

We chose not to include KM and HC in this analysis for several reasons. The tests conducted here are intended to provide a rough biological assessment of the gene and sample sets of biclusters that are produced with the primary goal of capturing gene-sample associations. In this sense, the tests here are assessing secondary features of these methods. By contrast, gene and sample based tests are separately assessing the primary features of KM and HC, for which

biclusters are a byproduct of their independent gene and sample grouping.

For 105 samples out of 117 in the dataset, we have information regarding overall survival (OS) and relapse free survival (RFS). We applied the standard logrank test [see Bewick et al. (2004)] to determine if there are significant differences between the survival times associated with samples in a bicluster and the survival times associated with samples in its complement. Biclusters whose associated patients have significantly lower (or higher) survival rates are of potential clinical interest, as their gene sets may point to biological processes that play a deleterious (or beneficial) role in survival. A bicluster was called significant if its samples passed the log-rank test for overall or relapse free survival at the 5% level. The number of biclusters meeting the criterion is presented in the **Survival** column of Table 2.3.

We next tested the gene set of each bicluster for over-representation of biologically derived functional categories and genomic neighborhoods. For the former, we considered KEGG categories [Kyoto Encyclopedia of Genes and Genomes, Kanehisa & Goto (2000), http://www.genome.jp/kegg/]. For the latter we considered cytobands, which consist of disjoint groups of genes such that the genes in a group have contiguous genomic locations. Definitions of KEGG and cytoband categories were taken from R metadata packages on Bioconductor (Bioconductor v 1.9, packages hgug4110b and hgu95av2).

For each bicluster gene set we computed a Bonferroni corrected hypergeometric p-value to assess its overlap with each KEGG category, and computed a similar p-value for each cytoband. We considered 153 KEGG categories and 348 cytobands that contained at least 10 genes (post filtering) on our sample arrays. A gene set was deemed to have significant overlap if any of the p-values computed in this way was less than 10^{-10}. This threshold was selected to adjust for the anti-conservative behavior of the hypergeometric test in the presence of even moderate levels of gene correlation [see Barry et al. (2005) for more details]. The column **Gene** of Table 2.3 shows the number of biclusters having signficant overlap with at least one KEGG category or cytoband.

The third test concerns the differential expression of KEGG and cytoband categories across the sample set of a bicluster and its complement. From each bicluster we formed a treatment group consisting of the samples in the bicluster and a control group consisting of the complementary samples that are not in the bicluster. We tested for KEGG categories showing

differential expression across the defined treatment and control groups using the SAFE procedure of Barry et al., and counted the number of categories passing the test at the 5% level. The permutation based approach in SAFE accounts for multiple comparisons and the (unknown) correlation among genes. A similar testing procedure was carried out for cytobands.

If no KEGG category were differentially expressed across the treatment and control groups corresponding to a particular bicluster, roughly 5% of the categories would exhibit significant differential expression by chance. We considered a bicluster sample set to yield significant differential expression of KEGG categories if the number of significant categories identified by SAFE exceeds the 5th percentile of the $\text{Bin}(153, .05)$ distribution. An analogous determination was made for cytobands. The number of biclusters whose sample sets yield significant differential expression for KEGG categories or cytobands is presented in the **Sample** column of the Table 2.3.

	# of BC's	Survival 5% level	KEGG/Cytoband Gene	KEGG/Cytoband Sample	2 out of 3	All 3
LAS	60	10	15	24	11	1
Plaid	60	10	3	17	2	0
CC	60	8	0	12	2	0
ISA	72	2	18	23	5	0
OPSM	15	0	0	3	0	0
SAMBA	289	15	20	72	10	1
BiMax	1977	329	0	0	0	0

Table 2.3: The number of biclusters passing tests for survival, gene-set enrichment, and sample-set differential expression of KEGG categories and cytobands. A detailed description of the tests is given in the text.

The rightmost columns of Table 2.3 show the number of biclusters passing two or three tests. From an exploratory point of view, these biclusters are of potential interest, and represent a natural starting point for further experimental analysis. Accounting for the number (or effective number) of biclusters produced by each method, specifically the large output of SAMBA and the small output of OPSM, LAS outperformed the other methods under study, particularly in regards to biclusters satisfying two out of the three tests.

2.4.5 Classification

Biclustering algorithms identify distinguished sample-variable associations, and in doing so, can capture useful information about the data. To assess how much information about disease subtypes and ER status is captured by the *set* of biclusters produced by different methods, we examined the classification of disease subtypes using patterns of bicluster membership in place of the original expression measurements. Similar applications of biclustering for the purpose of classification were previously investigated in Tagkopoulos et al. (2005) and unpublished work [Grothaus (2005), Asgarian & Greiner (2006)].

Once biclusters have been produced from the data matrix, we replaced each sample by a binary vector whose jth entry is 1 if the sample belongs to the jth bicluster, and 0 otherwise. A simple k-nearest neighbor classification scheme based on weighted Hamming distance was applied to the resulting binary matrix: the classification scheme used the subtype assignments of training samples to classify unlabeled test samples. The number of rows in the derived binary matrix is equal to the number of biclusters; in every case this is far fewer than the number of genes in the original data.

To be more precise, let $X = [x_1, \ldots, x_n]$ be an $m \times n$ data matrix, and let $C_1, \ldots, C_K$ be the (index sets of) the biclusters produced from X by a given biclustering method. We map each sample (column) x_i into a binary vector $\pi(x_i) = (\pi_1(x_i), ..., \pi_K(x_i))'$ that encodes its bicluster membership:

$$\pi_k(x_i) \;=\; \begin{cases} 1 & \text{if } x_i \text{ belongs to the sample set of bicluster } C_k \\ 0 & \text{otherwise.} \end{cases}$$

The original data matrix X is then replaced by the $K \times n$ "pattern" matrix $\Pi = \{\pi(x_1), \ldots, \pi(x_n)\}$. In the Hu data, for example, the 13,666 real variables in X are replaced by $K < 300$ binary variables in Π. Subtype and ER designations for the initial data matrix X carry over to the columns of Π.

For each of the breast cancer subtypes in the Hu data, we used 10-fold cross validation to assess the performance of a 5-nearest neighbor classification scheme applied to the columns of the binary pattern matrix Π. The nearest neighbor scheme used a weighted Hamming

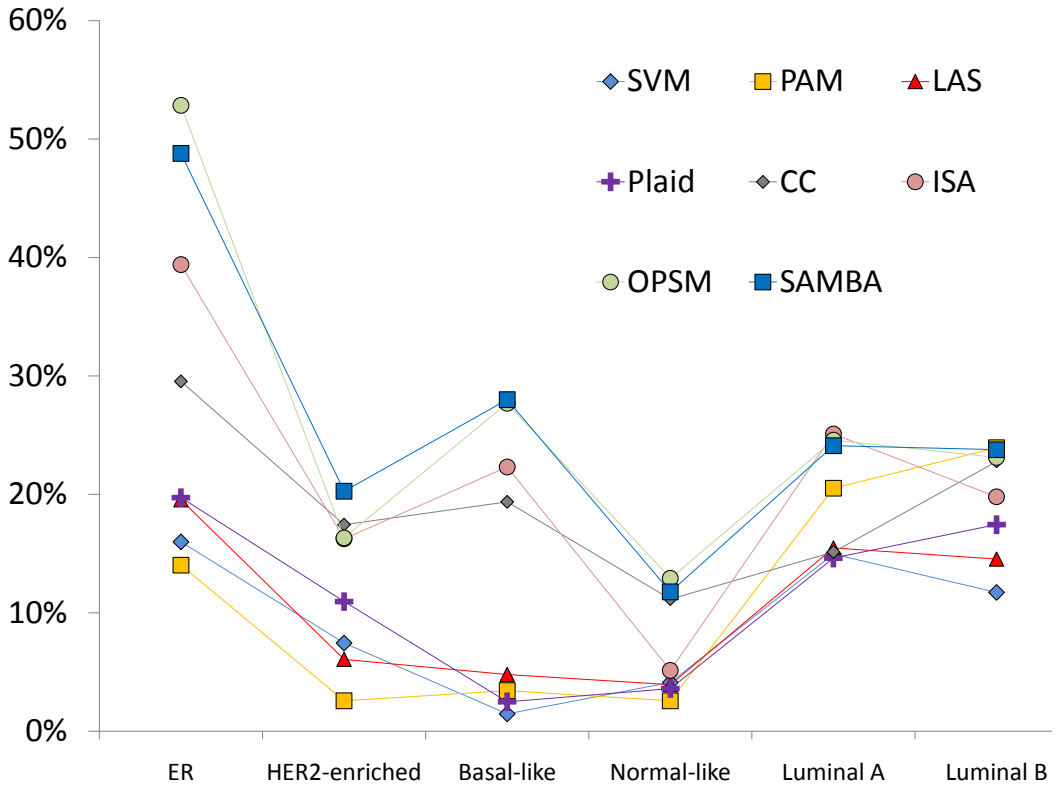

Figure 2.5: Classification error rates for SVM on the original data and the 5-nearest neighbor with weighted Euclidean distance applied to the "pattern" matrix.

distance measure, in which the weight of each row is equal to the square of the t-statistic for the correlation r between the row and the response, $t^2 = (n-2)r^2/(1-r^2)$. In each case, the weights were calculated using only the set of training samples. For each subtype, the average number of cross-validated errors was divided by the total number of samples, in order to obtain an overall error rate. The results are displayed in Figure 2.5. For comparison, we include 10-fold cross validation error rates of a support vector machine (SVM) classifier applied to the original expression matrix X. As the figure shows, the error rates of LAS and Plaid are similar to those of SVM across the phenotypes under consideration. Using the pattern information from 60 biclusters, LAS and Plaid were able to distinguish individual subtypes with the same degree of accuracy as SVM applied to the original data with 13,666 variables.

2.4.6 Lung Data

We have also performed the validation analysis described above on the lung cancer date from Bhattacharjee et al. (2001). The results are similar to those for the breast cancer data. The principle difference was the improved performance of ISA in tests of subtype capture. While ISA biclusters continued to have small sample sets, the disease subtypes for the lung data contained fewer samples than those in the breast data.

2.5 Simulations

In addition to real data, we also investigated the behavior of the LAS algorithm on a variety of artificially created datasets. Our primary goals were to assess (i) the ability of the algorithm to discover significant submatrices under the additive model (2.1), (ii) the stability of the algorithm with respect to the initial random number seed, and (iii) the sensitivity of the algorithm to noise.

2.5.1 Null Model with One Embedded Submatrix

The key step of the LAS algorithm is to identify a submatrix of a given matrix that maximizes the score function. The approach taken by LAS is heuristic. As there are no efficient algorithms for finding optimal matrices, outside of small examples, we cannot check directly if the submatrix identified by LAS is optimal. In order to evaluate the LAS search procedure, we generated a number of data matrices of the same size as the Hu dataset, with i.i.d. N(0,1) entries. For $k = 4, 8, 16, \ldots, 4096$ and $l = 4, 8, 16, 32$, we added a constant $\alpha = 0.1, 0.2, ..., 1$ to a $k \times l$ submatrix of the initial Gaussian matrix. The basic LAS search was carried out on each of the $11 \times 4 \times 10 = 440$ resulting matrices, and was considered a success if the search returned a bicluster whose score was at least as high as that of the embedded submatrix. The LAS search failed in only three cases; in each the embedded submatrix had relatively low scores (less than 100, while scores of other submatrices ranged up to 61,415.5). The search procedure was successful in all cases when the number of iterations used in the procedure was increased from 1,000 to 10,000.

2.5.2 Null Model with Multiple Embedded Submatrices

We also tested the ability of LAS to discover multiple embedded submatrices. Simulations were performed with a varying number of embedded biclusters (up to 50), with 10 simulations for each number of biclusters. In each simulation, we first generated a 1000×1000 Gaussian random matrix. Then we randomly selected size and position of each bicluster, independently assigning rows and columns of the matrix to the bicluster with probability .02, so that the expected size of a bicluster is 20×20. Biclusters were generated independently, allowing for overlap. The elements of every bicluster were increased by 2. Then LAS was applied to the resulting matrix set to search for the correct number of biclusters with the default 1000 iterations per bicluster.

Number of Biclusters	Average Match
1	1.000
2	0.997
3	0.997
4	1.000
10	1.000
20	0.999
30	0.993
50	0.989

Table 2.4: Discovery of multiple biclusters.

For every embedded bicluster U, we assessed its overlap with each detected bicluster $\tilde{U}$ using the minimum of specificity $|U \cap \tilde{U}|/|U|$ and sensitivity $|U \cap \tilde{U}|/|\tilde{U}|$, equivalently, $|U \cap \tilde{U}|/\max(|U|, |\tilde{U}|)$, and matched U with the closest $\tilde{U}$. The average overlap across embedded biclusters and simulations for various numbers of true biclusters is presented in Table 2.4. The numbers indicate consistent accuracy of LAS in the detection of multiple embedded biclusters.

2.5.3 Stability

In order to check the stability of LAS with respect to the randomization used in the basic search procedure, we ran the algorithm 10 times on the Hu dataset with different random seeds. In order to assess the stability of the performance of the algorithm, rather than its raw output, for each of the 10 runs we calculated the validation measures from Section 2.4. The effective number of biclusters, average size, p-values for subtype capture (as in Section 2.4.3) are presented in Table 2.5. The number of biclusters that passed different biological tests (as in Section 2.4.4) is presented in Tables 2.6 and 2.7.

There is little variation in the calculated measures across different runs of the algorithm. The effective number of biclusters ranged from 48.2 to 49.0, and average size ranged from

				Average # of	
	N clusters	Eff num	ratio	Samples	Genes
LAS01	60	49.0	0.816	26.2	360.7
LAS02	60	48.6	0.811	26.0	358.5
LAS03	60	48.5	0.809	26.4	357.6
LAS04	60	48.3	0.805	26.7	357.4
LAS05	60	49.0	0.817	26.2	361.8
LAS06	60	48.6	0.810	26.0	360.8
LAS07	60	48.5	0.808	26.2	360.5
LAS08	60	48.2	0.804	27.1	358.7
LAS09	60	48.5	0.809	26.7	355.6
LAS10	60	48.8	0.814	25.9	363.0

Table 2.5: Summary table for 10 runs of LAS on the Hu data with different random seeds.

	ER	HER2-enriched	Basal-like	Normal-like	Luminal A	Luminal B
LAS01	9.1	10.9	18.5	8.8	8.5	4.2
LAS02	5.9	10.9	18.5	10.9	9.7	5.0
LAS03	7.4	11.3	19.9	10.1	8.5	6.3
LAS04	7.4	12.2	18.5	10.1	9.1	7.2
LAS05	7.4	10.9	16.4	10.1	10.0	4.7
LAS06	7.4	10.9	19.9	10.1	9.4	5.9
LAS07	7.4	10.9	19.9	10.1	9.4	6.9
LAS08	7.4	10.9	18.5	10.1	9.4	8.5
LAS09	8.3	11.3	18.5	8.3	9.1	4.2
LAS10	7.4	10.9	19.9	10.1	8.9	6.1

Table 2.6: Minus $\log_{10}$ p-values of best subtype capture for 10 runs of LAS with different random seeds.

355×26 to 363×27. The number of biclusters with significant survival ranged from 9 to 13, and the number of biclusters having significant overlap with at least one KEGG category or cytoband ranged from 13 to 16. The SAFE analysis is computationally intensive, so we did not perform it for these simulations. Although the output of LAS is not deterministic, its summary statistics for average size and overlap are stable, and it is consistently successful in capturing cancer subtypes.

2.5.4 Noise Sensitivity

In order to assess the effects of noise on the LAS output, we added zero mean Gaussian noise with standard deviation $\sigma = 0, 0.1, 0.2, \ldots, 1$ to the normalized Hu dataset (after tail transformation

	# of BC's	Survival 5% level	KEGG/Cytoband Gene	2 out of 2
LAS01	60	9	16	4
LAS02	60	11	16	4
LAS03	60	12	14	3
LAS04	60	13	14	4
LAS05	60	9	15	3
LAS06	60	10	13	3
LAS07	60	8	13	2
LAS08	60	13	13	3
LAS09	60	12	14	4
LAS10	60	9	14	2

Table 2.7: The number of LAS biclusters (for 10 runs of LAS with different random seeds) passing tests for survival, and gene-set enrichment.

and column standardization). The resulting matrix was then column standardized, and LAS was applied to produce 60 biclusters.

For each level of noise we calculated validation measures for the LAS output. The results are presented in Tables 2.8, 2.9 and 2.10. As the level of noise increases, the average number of genes in the LAS biclusters decreased, as did the number of biclusters with having a significant association with Cytoband or KEGG categories. However, within the tested range of noise levels, the average number of samples did not change noticeably, and the subtype capture performance did not markedly decrease. The results indicate both high noise resistance of LAS and the strength of subtype signal.

2.6 Minimum Description Length Connection

Let the data matrix X be standardized, so that its elements have zero mean and unit variance, and let U be the selected bicluster. The code describing the data matrix must describe both the bicluster (size, location, average of its elements) and the residual matrix.

It is not possible to code real-valued data precisely with a finite-length code, so we construct a code describing the data with a precision of C binary digits after the decimal point.

The size of the submatrix U is described by variables $k \in [m]$ and $l \in [n]$, so coding these variables requires $\log_2(mn)$ bits. (We ignore rounding issues here and in what follows.) There are a total of $\binom{m}{k}\binom{n}{l}$ different $k \times l$ submatrices in a $m \times n$ matrix, so the code describing

				Average # of	
	N clusters	Eff num	ratio	Samples	Genes
$\sigma = 0.0$	60	48.3	0.806	26.2	357.3
$\sigma = 0.1$	60	49.2	0.819	26.6	359.1
$\sigma = 0.2$	60	49.4	0.823	26.9	342.8
$\sigma = 0.3$	60	47.7	0.795	26.0	338.0
$\sigma = 0.4$	60	49.5	0.825	26.9	319.8
$\sigma = 0.5$	60	50.3	0.838	26.7	297.2
$\sigma = 0.6$	60	50.3	0.839	26.8	274.5
$\sigma = 0.7$	60	50.9	0.849	26.5	252.9
$\sigma = 0.8$	60	51.8	0.863	25.5	227.3
$\sigma = 0.9$	60	51.7	0.862	25.1	208.5
$\sigma = 1.0$	59	51.6	0.875	25.8	183.0

Table 2.8: Summary statistics of LAS biclusters for data with added noise.

	ER	HER2-enriched	Basal-like	Normal-like	Luminal A	Luminal B
$\sigma = 0.0$	6.3	10.9	18.5	10.9	8.4	5.8
$\sigma = 0.1$	7.4	10.9	18.5	10.9	10.0	6.7
$\sigma = 0.2$	8.3	11.3	15.3	9.6	8.0	6.3
$\sigma = 0.3$	6.9	13.9	18.0	10.1	10.0	8.6
$\sigma = 0.4$	5.4	11.3	17.6	10.6	8.8	6.9
$\sigma = 0.5$	6.3	12.2	18.5	8.8	8.4	6.6
$\sigma = 0.6$	6.7	10.9	15.3	11.6	7.7	4.8
$\sigma = 0.7$	8.3	12.2	15.3	10.1	8.1	5.8
$\sigma = 0.8$	7.6	12.2	17.3	8.8	8.1	7.2
$\sigma = 0.9$	10.0	10.9	17.3	7.6	8.4	5.0
$\sigma = 1.0$	9.6	12.2	16.2	8.0	11.3	10.7

Table 2.9: Minus $\log_{10}$ p-values of best subtype capture for LAS on data with added noise.

the location of the submatrix requires $\log_2[\binom{m}{k}\binom{n}{l}]$ bits. To code the submatrix average τ, we assume that it lies within the interval $[-8, 8]$ (we did not observe $|\tau| > 1.5$ in our experiments). Then the code describing the average τ of the submatrix U takes $\log_2 16 + C = 4 + C$.

Finally, we describe the residual matrix. The data set is standardized, so its total variation (sum of squares) is nm. A $k \times l$ submatrix with average τ explains variation $\tau^2 kl$, so that the variation of the residual matrix is $nm - \tau^2 kl = nm\left[1 - \frac{kl\tau^2}{nm}\right]$. Thus, under model (2.1) the elements of the residual matrix are approximately distributed as $N(0, 1 - \frac{kl\tau^2}{nm})$.

Coding of a random variable X with density $f(x)$ requires $-\log_2(f(X)) + C$ bits, or, on average, $-\int f(x)\log_2(f(x))dx + C$ bits. Let C_N be the average code length for standard normal

	# of BC's	Survival 5% level	KEGG/Cytoband Gene	2 out of 2
$\sigma = 0.0$	60	13	16	3
$\sigma = 0.1$	60	9	15	2
$\sigma = 0.2$	60	12	14	2
$\sigma = 0.3$	60	10	14	3
$\sigma = 0.4$	60	9	13	2
$\sigma = 0.5$	60	9	14	3
$\sigma = 0.6$	60	8	13	3
$\sigma = 0.7$	60	7	13	1
$\sigma = 0.8$	60	8	12	1
$\sigma = 0.9$	60	11	9	3
$\sigma = 1.0$	59	5	9	1

Table 2.10: The number of LAS biclusters (on data with added noise) passing tests for survival, and gene-set enrichment of KEGG categories and cytobands.

random variable, $C_N = -E \log_2(\phi(Z)) + C$, where $z \sim N(0,1)$ and $\phi(z)$ is density of standard normal distribution. Then for $X \sim N(0, \sigma^2)$ the average code length is $C_N - \log_2(\sigma^2)/2$. Thus, coding of the residual matrix takes $nm \left[C_N - \log_2 \left[1 - \frac{kl\tau^2}{nm} \right] \right]$ bits on average.

Combining the codelengths above, the length of the code describing the X using a $k \times l$ bicluster U with average τ is

$$MDL(U) \;=\; \log_2(nm) + 4 + C + \log_2 \left[\binom{m}{k} \binom{n}{l} \right] + nm \left[C_N - \frac{1}{2} \log_2 \left(1 - \frac{kl\tau^2}{nm} \right) \right].$$

In the applications we considered, the explained variation was a small fraction (typically less than 1/1000) of the total variation. Thus, we can apply first order approximation: $\log_2[1+x] = x/\ln(2) + o(x)$. Then

$$MDL(U) \;\approx\; const + \log_2 \left[\binom{m}{k} \binom{n}{l} \right] - kl\tau^2/2\ln(2).$$

We pulled the constant terms and terms depending on n and m out, as they do not depend on the selected bicluster.

Let's now consider the LAS score function,

$$S(U) \;=\; -\log \binom{m}{k} - \log \binom{n}{l} - \log \left(\Phi(-\sqrt{v^2 kl}) \right).$$

For large x we can approximate $\Phi(-x) = \exp[-x^2/2]/x + o(x)$, getting

$$S(U) \approx \ln(2) \left[-\log_2 \binom{m}{k} \binom{n}{l} + \tau^2 kl/2 \ln(2) - \log_2(\tau^2 kl)/2 \right].$$

Easy to see that except for the small factor of $\log_2(\tau^2 kl)/2$ the code length and score function approximations are linearly dependent:

$$S(U) \approx const - \ln(2) MDL(U).$$

2.6.1 LAS model and low rank signal detection

Note that the matrix with B biclusters (and no noise) has rank at most B, so the LAS model is a particular case of low-rank signal detection model considered in Chapter 1. Applying the results from Chapter 1 to the LAS model one can determine that SVD can only detect biclusters (of fixed average) with number of elements more that square root of the number of elements in the whole data matrix. However, using LAS algorithm, which aims to find LAS biclusters, not arbitrary rank one signal, we can find biclusters with number of elements logarithmically small compared to the size of the matrix.

To be expanded.

2.7 Discussion

Biclustering methods are a potentially useful means of identifying sample-variable associations in high-dimensional data, and offer several advantages over independent row-column clustering. Here we have presented a statistically motivated biclustering algorithm called LAS that searches for large average submatrices. The algorithm is driven by a simple significance-based score function that is derived from a Bonferroni corrected p-value under a Gaussian random matrix null model. We show that maximizing the LAS score function is closely related to minimizing the overall description length of the data in an additive submatrix Gaussian model.

The LAS algorithm operates in a sequential-residual fashion; at each stage the search for a submatrix with maximum score is carried out by a randomly initialized iterative search procedure that is reminiscent of EM type algorithms. The only operational parameters of

LAS are the number of biclusters it produces before halting, and the number of randomized searches carried out in identifying a bicluster. In our experiments on real data, we found that 1000 randomized searches per bicluster were sufficient to ensure stable performance of the algorithm.

We evaluated LAS and a number of competing biclustering methods using a variety of quantitative and biological validation measures. On the quantitative side, LAS produced biclusters exhibiting a wide range of gene and sample sizes, and low to moderate overlap. The former feature implies that LAS is capable of capturing sample-variable associations across a range of different scales, while the latter indicates that distinct LAS biclusters tend to capture distinct associations. Other methods varied considerably in their sizes and overlap, with a number of methods producing biclusters having a small number of samples and genes.

Many LAS biclusters had significantly higher scores than biclusters obtained by more traditional approaches based on k-means and hierarchical clustering. This suggests that the constraints associated with independent row-column clustering (considering rows and columns separately, assigning each row or column to a single cluster) substantially limit the ability of these methods to identify significant biclusters, and that more flexible methods may yield substantially better results.

In regards to capturing disease subtypes, LAS was competitive with, and often better than, KM and HC. Other methods did not perform particularly well, though we note that ISA did a good job of capturing and classifying the smaller disease subtypes present in the lung cancer data. In tests for survival, over-representation of functional categories, and differential expression of functional categories, LAS outperformed the other biclustering methods. These tests, unlike the quantitative measures of size and overlap, were based on clinical and biological information.

The classification study in Section 2.4.5 shows that simple binary profiles of bicluster membership can contain substantive information about sample biology. In particular, nearest neighbor classification of disease subtypes using membership profiles derived from LAS or Plaid biclusters was competitive with a support vector machine classifier applied to the full set of expression data. We note that the biclustering methods applied here are unsupervised, and depend only on the expression matrix: none makes use of auxiliary information about samples

or variables.

Our simulation study shows that the LAS search procedure is effective at capturing embedded submatrices (or more significant ones) having moderate scores. Although the search procedure makes use of random starting values, its performance is stable across different random seeds. The ability of the algorithm to capture subtypes does not substantially deteriorate when a moderate amount of noise is added to the data matrix.

While the validation of biclustering here has focused on gene expression measurements, it is important to note that LAS and other biclustering methods are applicable to a wide variety of high-dimensional data. In preliminary experiments on high density array CGH data produced on the Agilent 244k Human Genome CGH platform, LAS was able to capture known regions of duplications and deletion (data not shown). The dataset contained roughly 250 samples and 240,000 markers. We note that among the seven biclustering methods compared in the chapter only the current implementations of LAS and Plaid were able to handle datasets of this size.

LAS biclusters capture features of the data that are of potential clinical and biological relevance. Although some findings, such as disease subtypes, are already known, very often the methods used to establish them involve a good deal of subjective intervention by biologists or disciplinary scientists. LAS provides a statistically principled alternative, in which intervention (such as selecting biclusters of potential interest) can take place after the initial discovery process is complete.

We note that the LAS score function and search procedure can readily be extended to higher dimensional arrays, *for example*, three-dimensional data matrices of the form $\{x_{i,j,k} : i \in [m], j \in [n], k \in [p]\}$. Related extensions of the Plaid model have been developed by Turner, Bailey, Krzanowski & Hemingway (2005).

As noted in Section 2.1.1, our use of the large average criterion is motivated by current biological practice in the analysis of gene expression and related data types. The validation experiments in the chapter establish the efficacy of the large average criterion, and the LAS search procedure, for standard gene expression studies, and there is additional evidence to suggest that the criterion will be effective in the analysis of CGH data. Nevertheless, we note that the large average criterion is one of many that may be used in the exploratory analysis of high dimensional data. Other criteria and methods can offer additional insights into a given

data set of interest, and may provide valuable information in cases, and for questions, where the large average criterion is not appropriate.

CHAPTER 3

FastMap: Fast eQTL Mapping in Homozygous Populations

Summary

Gene expression Quantitative Trait Locus (eQTL) mapping measures the association between transcript expression and genotype in order to find genomic locations likely to regulate transcript expression. The availability of both gene expression and high density genotype data has improved our ability to perform eQTL mapping in inbred mouse and other homozygous populations. However, existing eQTL mapping software does not scale well when the number of transcripts and markers are on the order of 10^5 and 10^5 - 10^6, respectively.

We propose a new method, FastMap, for fast and efficient eQTL mapping in homozygous inbred populations with binary allele calls. FastMap exploits the discrete nature and structure of the measured SNPs. In particular, SNPs are organized into a Hamming distance based tree that minimizes the number of arithmetic operations required to calculate the association of a SNP by making use of the association of its parent SNP in the tree. FastMap's tree can be used to perform both single marker mapping and haplotype association mapping over an m-SNP window. These performance enhancements also permit permutation based significance testing.

The FastMap program and source code are available at the website: http://cebc.unc.edu/fastmap86.html

3.1 Introduction

Quantitative Trait Locus (QTL) mapping is a set of techniques that locates genomic loci associated with phenotypic variation in a genetically segregating population. QTL mapping has been

highly successful in determining causative loci underlying several disease phenotypes (Cervino et al. 2005, Wang et al. 2004, Hillebrandt et al. 2005) and can broadly be subdivided into two classes: linkage mapping and association mapping. For standard linkage mapping in experimental crosses, likelihood or regression approaches are used to map QTL, with flanking markers used to infer genotypes in the intervals between widely spaced markers (i.e. $>$ 1cM) (Haley & Knott 1992, Lander & Botstein 1989). As marker density increases, linkage statistics may be computed at individual marker loci, with minimal loss in precision or power (Kong & Wright 1994). In contrast, simple association mapping does not attempt to explicitly consider the linkage disequilibrium structure between marker loci, and thus typically considers association statistics computed only at the marker loci. In either case, the statistics computed at the markers in experimental cross linkage designs, and in association studies, are often identical, e.g. t-statistics to detect differences in phenotype means as a function of genotype. Here, we consider the case of markers collected at sufficient density so that association statistics may be calculated only at the observed markers.

Recent advances in gene expression and single nucleotide polymorphism (SNP) microarray technology have lowered the cost of collecting gene expression and high density genotype data on the same population. These technologies have been used to produce high density SNP data sets with thousands of transcripts and millions of allele calls in both mice (Frazer, Eskin, Kang, Bogue, Hinds, Beilharz, Gupta, Montgomery, Morenzoni, Nilsen et al. 2007, Szatkiewicz et al. 2008) and humans (Frazer, Ballinger, Cox, Hinds, Stuve, Gibbs, Belmont, Boudreau, Hardenbol, Leal et al. 2007). eQTL mapping has been successfully carried out in several inbred mouse populations (Bystrykh et al. 2005, Chesler et al. 2005, Gatti et al. 2007, McClurg et al. 2007, Pletcher et al. 2004, Schadt et al. 2003). These studies have provided a revealing genome-wide view of the genetic basis of transcriptional regulation in multiple tissues, and form a necessary foundation for systems genetics (Kadarmideen et al. 2006, Mehrabian et al. 2005).

The calculation of associations between tens of thousands of transcripts and thousands to millions of SNPs creates a computational challenge that can stretch or overwhelm existing tools. These challenges are further compounded by multiple comparison issues arising from the large number of available SNPs and transcripts. Various methods have been used to address these issues. A resampling approach (Carlborg et al. 2005, Churchill & Doerge 1994, Peirce et al.

2006) is one common way of addressing multiple comparisons among markers, and it is used by several available QTL mapping tools (Broman et al. 2003, Manly et al. 2001, Wang et al. 2003). Multiple comparisons among transcripts has been previously addressed by thresholding transcripts using q-values (Storey & Tibshirani 2003) obtained from transcript specific testing of association with SNPs using likelihood ratio statistic (Chesler et al. 2005) or the mixture over markers method (Kendziorski et al. 2006).

While parallel computation has been suggested as a potential solution to the computational challenges associated with eQTL analysis, (Carlborg et al. 2005), however many researchers have neither the expertise nor the resources required to administer and maintain a computing cluster. To address the growing need for eQTL mapping in high density SNP data sets, and the poor scalability of the existing computational tools, we developed the FastMap algorithm and implemented it as a Java-based, desktop software package that performs eQTL analysis using association mapping. We achieved computational efficiency through the use of a data structure called a Subset Summation Tree, which is described in the Methods section below. FastMap performs either single marker mapping (SMM) or haplotype association mapping (HAM) by sliding an m-SNP window across the genome (Pletcher et al. 2004). FastMap is currently intended for use with inbred mouse strains. Significance thresholds and p-values are calculated for each transcript using multiple permutations of transcript expression values. In order to address multiple comparisons across transcripts, FastMap assigns a q-value (Storey & Tibshirani 2003) assessing FDR, to each transcript. We apply our software tool to two publicly available data sets consisting of gene expression measurements in panels of inbred mice and compare our results to other software tools.

3.2 The FastMap Algorithm

This section describes the calculations of test statistics (correlations) for SMM in a 1 SNP sliding window. First we introduce the concept of a subset sum $M_g(s)$ and a Subset Summation Tree. Subset sums are quantities that can be efficiently calculated using the subset summation tree, and are used in the calculation of correlations. We then show how the subset sums and Subset Summation Tree can be adapted to the fast calculation of ANOVA test statistics for m-SNP sliding windows ($m > 1$).

In association mapping for homozygous inbred strains, the input data consists of two matrices: the first contains real-valued transcript expression measurements and the second contains SNP allele calls, coded as 0 for the major allele and 1 for minor allele. Each matrix has the same number of samples (strains) n. Let S be the number of SNPs and let G be the number of transcripts.

3.2.1 Test Statistic for 1-SNP–Transcript Association

Homozygous SNPs: 1 SNP window. We use the Pearson correlation as an association statistic in the case of a 1 SNP window. For a given transcript g and SNP s the correlation between g and s is

$$\text{cor}(g,s) \;=\; \frac{\text{cov}(g,s)}{\sqrt{Var(g)Var(s)}} \;=\; \frac{\frac{1}{n}\sum_{i=1}^{n} g_i s_i - \frac{1}{n^2}\sum_{i=1}^{n} g_i \sum_{i=1}^{n} s_i}{\sqrt{Var(g)Var(s)}}.$$

To simplify the formula, we assume without loss of generality that each transcript expression vector g is centered and standardized such that

$$\sum_{i=1}^{n} g_i \;=\; 0 \quad \text{and} \quad \sum_{i=1}^{n} g_i^2 \;=\; 1. \tag{3.1}$$

In this case, the correlation expression reduces to

$$\text{cor}(g,s) \;=\; \frac{\sum_{i=1}^{n} g_i s_i}{\sqrt{n}\sqrt{\frac{1}{n}\sum_{i=1}^{n} s_i^2 - (\frac{1}{n}\sum_{i=1}^{n} s_i)^2}}.$$

The denominator can be calculated once for each SNP, because it depends only upon the Hamming weight of s. By contrast, the numerator must be calculated for every SNP-transcript pair ($S \times G$ computations). Our goal is to speed up calculation of the numerator. Denote the numerator by $M_g(s)$:

$$M_g(s) \;=\; \text{cov}(g,s) \;=\; \sum_{i=1}^{n} g_i s_i \;=\; \sum_{i:s_i=1} g_i.$$

As the SNPs are binary, $M_g(s)$ is simply the sum of transcript expression values over a subset of samples defined by the minor allele of the SNP.

To illustrate how the calculation of the $M_g(s)$ can be simplified, consider two SNPs s and

s' that differ only at the ith position (thus s and s' have Hamming distance of 1):

$$s = (s_1, s_2, ..., s_{i-1}, s_i = 0, s_{i+1}, ..., s_n)$$

$$s' = (s_1, s_2, ..., s_{i-1}, s'_i = 1, s_{i+1}, ..., s_n).$$

In this case, the quantity $M_g(s')$ can be calculated quickly (in one arithmetic operation) from $M_g(s)$ as follows:

$$M_g(s') = \sum_{i=1}^{n} g_i s'_i = \sum_{i=1}^{n} g_i s_i + g_i(s'_i - s_i) = M_g(s) + g_i(1-0) = M_g(s) + g_i. \tag{3.2}$$

For any given transcript, the association statistic is the same for SNPs with the same strain distribution pattern (SDP). Hence, we calculate the association statistic once for each unique SDP. The McClurg mouse data used in this chapter contains 156,525 SNPs, but has only 64,157 unique SDPs.

Gene	g_1	g_2	g_3	g_4	g_5	g_6	g_7
SNP	s_1	s_2	s_3	s_4	s_5	s_6	s_7
1	0	1	0	0	0	0	0
2	0	1	1	0	0	0	0
3	0	1	1	0	1	0	0
4	0	0	0	1	1	0	1
5	1	0	0	0	1	0	0
6	1	0	0	0	1	1	0

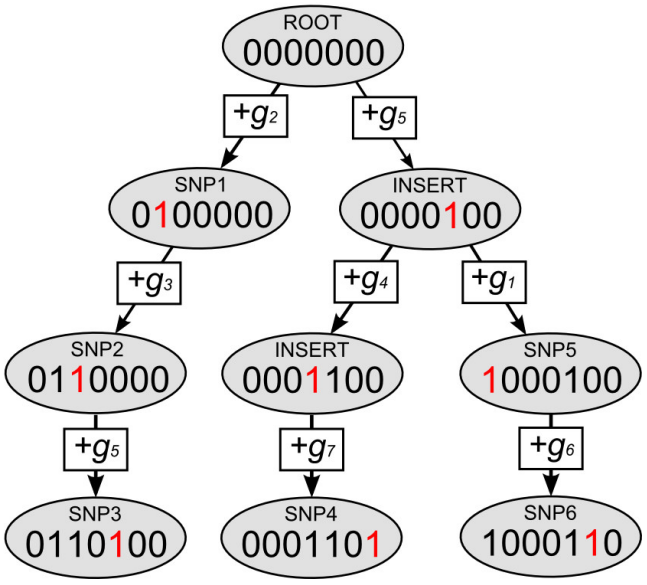

Figure 3.1: Illustration of the Subset Summation Tree. The table shows one gene expression vector and 6 corresponding SNP vectors for 7 strains. At each node, the covariance of the gene expression with each SNP is calculated with one addition operation.

3.2.2 Subset Summation Tree

Additional improvements are based on Formula 3.2. To take full advantage of this relationship between correlations, we construct a tree, which we call a Subset Summation Tree. The vertices of the tree correspond to unique subsets of samples. Each SDP defines a subset of samples associated with its minor allele. The tree contains all SDPs appearing in the SNP matrix. By construction, the edges of tree connect SDPs which differ in one position (i.e. Hamming distance 1). The process of tree construction is described later in this section. It ensures that the tree is at least as efficient (in terms of weight based on the Hamming distance) as the minimum spanning tree connecting all SDPs from the SNP matrix. An illustration of a subset summation tree is given in Figure 3.1.

Traversing the tree we can calculate the covariance $M_g(s)$ for all SDPs in the tree with 1 arithmetic operation per SDP. One additional arithmetic operation is required to calculate the correlation from $M_g(s)$.

3.2.3 Test Statistic for m-SNP–Transcript Association

The use of a consecutive 3-SNP sliding window has been shown to improve the associations that can be detected in mouse studies (Pletcher et al. 2004). FastMap is capable of employing any m-SNP window specified by the user. Within each m-SNP window, the strains form haplotypes that partition strains into ANOVA groups. A one way ANOVA test statistic is then used to assess the relationship between a gene g and an m-SNP window.

Consider a 3-SNP window that contains k unique haplotype (ANOVA) groups across the n stains. Let A_i denote the set of samples in the i-th ANOVA group, and let the transcript expression values in the i-th ANOVA group be g_i. The associated ANOVA test statistic is calculated as

$$F = \frac{(n-k)SSB}{(k-1)SSW},$$

where the between group sum of squares SSB, and within group sum of squares SSW are calculated as follows:

$$SSB = \sum_{i=1}^{n} n_i(\bar{g}_i - \bar{g})^2, \qquad SSW = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(g_{ij} - \bar{g}_i)^2,$$

$$\bar{g}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} g_{ij}, \qquad \bar{g} = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n_i} g_{ij}.$$

The sums of squares are related by $SST = SSB + SSW$.

For a given transcript, the total sum of squares (SST) remains constant across all SNPs. As in the 1-SNP window case, the gene expression values are standardized to satisfy the conditions in Equation(3.1). For standardized expression measurements, the SST and SSB calculations simplify as follows:

$$SST = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(g_{ij} - \bar{g})^2 = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(g_{ij} - 0)^2 = 1,$$

$$SSB = \sum_{i=1}^{k} n_i(\bar{g}_i - \bar{g})^2 = \sum_{i=1}^{k} n_i\left(\frac{\sum_{j=1}^{n_i} g_{ij}}{n_i}\right)^2 = \sum_{i=1}^{k}\frac{M_g^2(A_i)}{n_i},$$

where $M_g(A_i)$ is sum of the transcript expression values for the ith ANOVA group. As before, $M_g(A_i)$ can be calculated efficiently using the Subset Summation Tree. The difference is that the tree for these calculations connects subsets of samples defining the m-SNP ANOVA groups, as opposed to SDPs defined by single SNPs. Once the SSB is calculated, the F statistic is calculated as:

$$F = \frac{(n-k)SSB}{(k-1)(1-SSB)}.$$

3.2.4 Construction of Subset Summation Tree

The Subset Summation Tree is used for fast calculation of $M_g(A_i)$ - sums of transcript expression values over subsets of samples $\{A_i\}$. Tree construction is initiated by obtaining the family of sample subsets of interest $\{A_i\}$ from the set of SNPs. The tree is grown starting from single root element (empty subset) by sequential addition of the nearest element from $\{A_i\}$ to the tree.

All the subsets $\{A_i\}$ are put in a hash table (HT) that stores the subsets that are not yet members of the tree. The tree is grown by connecting subsets that are at the minimum distance from the tree. Node selection and connection to the tree can be optimized by taking advantage of two facts. First, the Hamming distances are positive integers. Thus, once we find a subset in the HT within distance 1 of a particular tree vertex, we connect them, adding the subset to

the tree and removing it from the HT. To find such an SDP in the HT we use the second fact: for any subset, there are only n possible subsets that are within hamming distance 1 from it. Thus, instead of calculating distances from a certain tree vertex to all subsets in the HT we can check if the HT contains any of the n possible neighbor subsets. This approach reduces the complexity of the search for close (within distance 1) neighbors of a given tree vertex from $O(nS)$ to $O(n)$.

The procedure above is applicable as long as there are SDPs in the HT within distance 1 from the tree. Once there are no SDPs in the HT within distance 1 from tree vertices, the search continues for SDPs within distance 2. The same optimizations are applicable here - once an SDP within distance 2 is found, it should be connected to the tree and there are $n(n-1)/2$ possible SDPs within distance 2 from a given tree vertex. The same technique is applied even for the search for subsets within distance 3. When the remaining vertices are at Hamming distance 4 or greater, an exhaustive search is performed to find a node in HT that is a minimum distance from the tree. This process is repeated until all SNPs have been inserted into the tree.

Permutation Based Significance Thresholds. For a single transcript, the association statistic is calculated between the observed values of that transcript and all SNPs. The transcript data is then permuted while the SNP data is held fixed. Association statistics are calculated between the permuted transcript values and all SNPs and the maximum association statistic is stored. The distribution of the maximum association statistics obtained from 1000 permutations of the transcript's values is used to define significance thresholds for individual (transcript, SNP) pairs, and to assign a percentile based p-value to the observed maximum association of the transcript across SNPs.

Significance Across Multiple Transcripts. The procedure above assigns a p-value to each transcript that accounts for multiple comparisons across SNPs through the use of the maximum association statistic. In order to correct for multiple comparisons across transcripts, we calculate q-values (Storey & Tibshirani 2003) for each transcript, using the p-values obtained from the permutation based maximum association test.

3.2.5 FastMap Application

FastMap is written in the Java programming language and is driven by a simple graphical user interface (GUI, Figure 3.2a). The required input files are 1) a transcript expression file with mean expression values for each mouse strain and 2) a SNP file containing allele calls for all strains, with the major and minor alleles coded as 0 and 1 respectively. Once the SNP file has been loaded, FastMap constructs a Subset Summation Tree (see Methods) for the SNP data, a computational task that is performed only once for a given set of strains. FastMap allows the user to perform either SMM by calculating the Pearson correlation of each transcript expression measurement with each SNP, or HAM by sliding an m-SNP window across the genome and calculating the ANOVA F-statistic for the phenotype vs. the distinct haplotypes observed in the window (Pletcher et al. 2004). The association statistic at each SNP is displayed in a zoomable panel that links to the University of California at Santa Cruz Genome Browser (Kent et al. 2002, Pontius et al. 2007) (Figure 3.2b & c). Association plots may be exported as text files or as images.

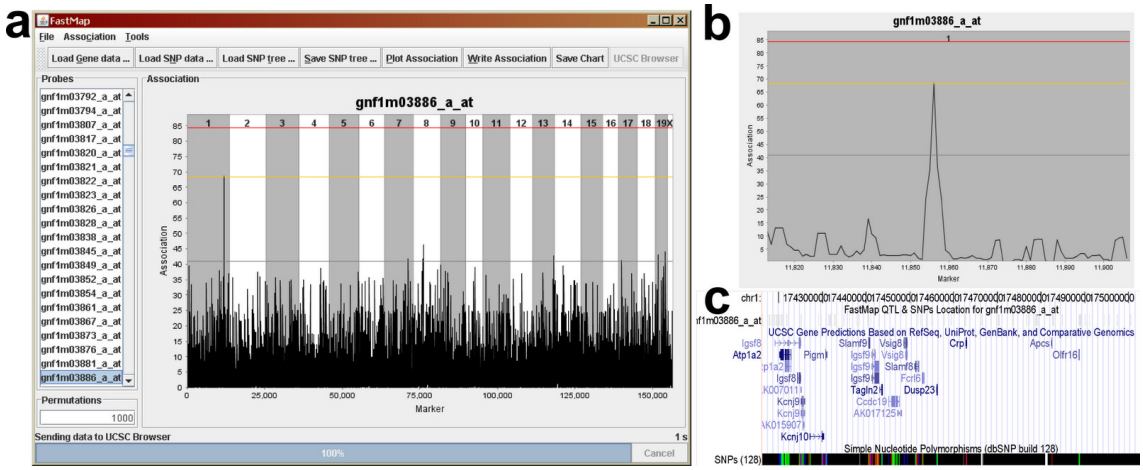

Figure 3.2: FastMap application GUI. Panel (a) shows FastMap with a list of probes on the left and the QTL plot on the right. Panel (b) shows a zoomed in view of the significant QTL on Chr 1. Panel (c) shows the same region in the UCSC Genome browser, to which FastMap can connect.

QTL mapping with sparsely distributed markers has traditionally used Maximum Likelihood methods and has employed the Likelihood Ratio Statistic (LRS) or the related Log of the Odds ratio (LOD) as a measure of the association between genotype and phenotype (LRS $= 2\ln(10)\times$

LOD). When marker density is high, regression techniques applied only at the observed markers will produce results which are numerically equivalent to the LRS or LOD (Kong & Wright 1994). In fact, the LRS, Student t-statistic, Pearson correlation and the standard F-statistic, can be shown to be equivalent when they are applied at the marker locations (see Supplementary Materials). While previous literature has shown that regression methods produce estimates with a higher mean square error and have less power (Kao 2000), these results apply primarily to the case of interval mapping when the spacing between markers is wide ($>$ 1cM). For these reasons, FastMap employs the Pearson correlation for SMM and the F-statistic for HAM when employing high density SNP data sets.

The significance of eQTLs for a single transcript may be determined using a permutation-based approach (Churchill & Doerge 1994). The expression values of each transcript are permuted, the association statistics of each transcript with all SNPs are calculated and the maximum transcript specific association statistic is retained. This process is repeated 1,000 times, and a significance threshold is taken as the $1-\alpha$ percentile of the empirical distribution of the maxima. Both the number of permutations and the significance thresholds may be specified by the user. Since the various association statistics are equivalent when applied at the markers, the significant marker locations will be the same for any choice of these statistics. Once a QTL peak that exceeds a user selected threshold has been identified, the width of the QTL must be defined in order to identify potential candidate genes for further study. Given a local maximum d, a confidence region can be defined as all markers q in an interval around d such that $2\ln(LR(q)) \geq \max_d 2\ln(LR(d)) - x$ and this interval is referred to as an $(x/2\ln 10)$-LOD support interval (Dupuis & Siegmund 1999). The choice of $x = 4.6$ yields a 1-LOD confidence interval, which has been widely used in linkage analysis. A more conservative choice of $x = 6.9$ (a 1.5-LOD interval) is more appropriate to situations with dense markers, yielding approximate 95% coverage under dense marker scenarios. Intervals for non-LR association statistics can be calculated from the relationships between statistics provided in the Supplementary Materials. In practice, eQTL peak regions are limited by the effective resolution determined by breeding and recombination history.

FastMap assigns a p-value to each transcript that indicates the significance of the maximum association of that transcript across all the available markers. In situations where it is necessary

or of interest to simultaneously consider multiple transcripts, additional steps must be taken to account for the resulting multiple comparison problem. We address this by calculating the q-value (Storey & Tibshirani 2003) of every transcript. The q-value of a transcript is related to the false discovery rate. In particular, the q-value of a transcript is an estimate of the fraction of false discoveries among transcripts that are equally or more significant than it is. For example, if we create a list of transcripts consisting of a transcript with q-value equal to 10%, and all those transcripts having smaller permutation based values, then we expect 10% or less of the transcripts on the list to have a significant association with at least on SNP or haplotype.

Permutation based significance testing is frequently used in eQTL analysis (Doerge & Churchill 1996, Peirce et al. 2006), and typically forms the bulk of the computational burden in eQTL mapping. It is natural to ask whether a parametric approach, based on Gaussian p-values, would be just as effective and save a significant amount of time. We note that permutation based testing offers several advantages over parametric approaches. Permutation testing deals cleanly with the problem of multiple comparisons, and induces a null distribution under which there is no association between transcript expression and genotype, regardless of the underlying distributions from which the data are drawn, and the correlations between SNPs. In addition, the normality assumptions underlying parametric tests are often violated in practice.

3.3 Test of Real Data

3.3.1 Data

BXD Gene Expression Data. The BXD Liver data set is available from genome.unc.edu, and is described in (Gatti et al. 2007). Briefly, it consists of microarray derived expression measurements for 20,868 transcripts in 39 BXD recombinant inbred strains and the C57BL/6J & DBA/2J parentals. The data was normalized using the UNC Microarray database and QTL analysis was performed on all transcripts.

BXD Marker Data. The BXD marker data consists of 3,795 informative markers taken from a larger set of 13,377 markers. Briefly, consecutive markers with the same strain distribution pattern were removed and only the flanking markers of such regions were included. The data was downloaded from www.genenetwork.org/genotypes/BXD.geno; further information is

available at www.genenetwork.org/dbdoc/BXDGeno.html.

Hypothalamus Gene Expression Data. The mouse hypothalamus data set GSE5961 was downloaded from the NCBI Gene Expression Omnibus website. This data is described in (McClurg et al. 2007). The 58 CEL files were normalized using the gcrma package from Bioconductor (version 1.9.9) in R (version 2.4.1). The data was subset to include only the 31 male samples, and removing the NZB data because the entire array appeared as an outlier in hierarchical clustering of the arrays. There were 36,182 probes on the array; of these a subset of 3,672 transcripts having an expression value >200 and at least a 3-fold difference in expression in one strain were selected. Transcripts containing a single outlier strain with expression values >4 standard deviations from the mean were removed from the data set. There were 402 such transcripts, leaving 3,270 transcripts for analysis in FastMap.

Hypothalamus SNP Data. The SNP data was obtained from (McClurg et al. 2007) and originally contained 71 inbred strains. Missing genotype data were imputed using the algorithm of (Roberts, McMillan, Wang, Parker, Rusyn & Threadgill 2007). There were 156,525 SNPs, of which 99 were monomorphic across the 32 strains. These SNPs were removed from the analysis, leaving 156,426 SNPs. There were 64,790 unique SDPs in this final data set.

3.3.2 Existing Methods

In this section we compare FastMap performance with two other publicly available tools: SNPster (McClurg et al. 2006) and R/qtl (Broman et al. 2003). The setting used to run them are detailed below.

Snpster settings: SNPster runs were performed using the tool available at snpster.gnf.org. The following settings were selected and are listed in the order in which they appear on the website. (i) Log transform data: No. (ii) Test statistic: F-test. (iii) Method of Calculating Significance: parametric. (iv) Compute gFWER: No. The default settings were used for the remaining options on the web site.

R/qtl settings: R/qtl version 1.08-56 for R 2.7 was used to perform eQTL analysis on the BXD Liver data set. R/qtl was configured to perform Haley-Knott regression only at the observed markers. eQTL significance was determined by performing 1000 permutations for each transcript and selecting only those eQTLs above the 95% LOD threshold.

Software	DataSet	Method	Transcripts	Markers	Time (min)
R/qtl	BXD	LRS	100	3,795	33.73
FastMap		SMM	20,868		29.95
SNPster	McClurg	HAM	3,672	156,525	6609.6
FastMap					737.5

Table 3.1: FastMap eQTL mapping times.

Computer for performance testing. A Pentium 4 with a clock speed of 3.4GHz and 4GB of RAM running Microsoft Windows XP Professional(r), SP2 was used for all timing runs. No other applications were open during the runs.

3.3.3 Performance and Speed

In order to gauge the performance improvement provided by FastMap over existing software, we compared computation times using two microarray data sets. The first consists of 20,868 transcripts and 3,795 markers in 41 strains of mice (BXD data set; (Gatti et al. 2007)). This data set was selected because, unlike the following larger data set, it can be loaded into the widely used R/qtl package without exhausting computer memory. The second is a hypothalamus data set (McClurg et al. 2006) that consists of 3,672 transcripts, 156,525 markers in 32 strains of laboratory inbred mice. This data set was selected for its dense genotype information, which is on the scale of the expected high density SNP data for which we designed FastMap.

The amount of time required to perform eQTL mapping in these data sets is summarized in Table 3.1. In the BXD data set, FastMap performs SMM for the entire set of 20,868 transcripts in about half an hour, which is the same time required for R/qtl to analyze 100 transcripts. The hypothalamus data was previously analyzed with an association mapping tool called SNPster (McClurg et al. 2006), which is available as a web application hosted by the Genomic Institute of the Novartis Research Foundation (GNF). A single transcript typically requires less than 5 minutes to analyze, depending on the load on SNPster's web server. However, obtaining results for thousands of transcripts from submissions to an external website is impractical in most cases. Another version of SNPster runs at GNF in parallel on a 200 node cluster, which is not publicly available, in batches of 10 transcripts per node. It requires 18 minutes to process these 10 transcripts using 1,000,000 bootstrap resamplings for each transcript, and a $-\log(\text{p-value})$ threshold of 2.5, which implies $\sim$1.8 CPU-minutes per transcript (T. Wiltshire, pers.com.).

If these 3,672 transcripts were analyzed serially rather than in parallel, this would require 110.2 hours. In contrast, FastMap runs on a standard desktop computer and can perform eQTL mapping for these same 3,672 transcripts with 156K SNPs in 32 strains in 12.3 hours. Large computing clusters, and the expertise required to administer them, are not available to all laboratories. FastMap offers the convenience of running on a single, local computer in a reasonable amount of time (overnight, or over a weekend for more than ten thousand transcripts).

We evaluated the scalability of FastMap with increasing numbers of transcripts and SNPs using the hypothalamus data set. Since we are aware of no stand-alone software that can perform eQTL mapping with hundreds of thousands of SNPs, we compared FastMap's performance in these plots to a brute force approach in which all calculations are performed without any optimizations. In the case of both SMM and HAM, computation time for FastMap scales linearly with increasing numbers of transcripts (Figure 3.3a). FastMap also scales linearly with increasing number of SNPs (Figure 3.3b).

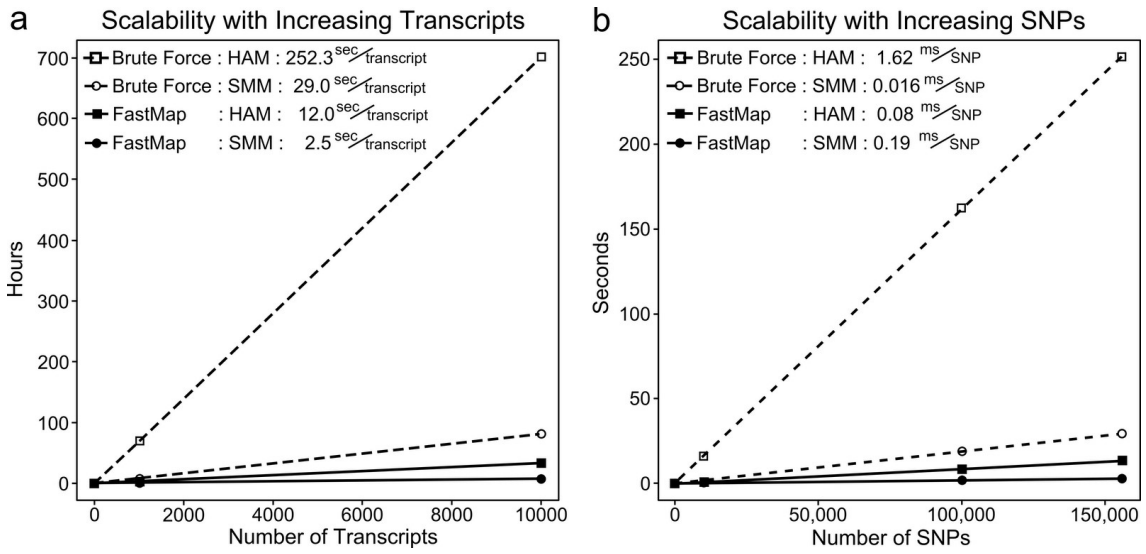

Figure 3.3: FastMap scales linearly with increasing numbers of genes and SNPs. Panels (a) & (b) show the time required to compute the association of increasing numbers of transcripts with 156K SNPs. Panels (c) & (d) show the time required to compute the association of one transcript with increasing numbers of SNPs. In all 4 cases, 1000 permutations per transcript were performed.

In order to examine the scalability of our algorithm with increasing numbers of strains, we

# Strains	1 SNP window		3 SNP window	
	Tree Constr.	SMM	Tree Constr.	HAP
16	1	0.05	8	2.8
32	168	2.5	320	12.9
54	3,791	2.8	27,138	16.5
71	13,672	4.6	81,186	25.5

Table 3.2: FastMap tree construction and association mapping times with increasing numbers of strains (in seconds)

determined tree construction times for various sets of inbred strains genotyped at ∼156,525 SNPs (Table 3.2). The amount of time required to construct the tree is a function of both the number of strains as well as their ancestral relationships. Strains that are closely related (i.e. all derived from *M.m.domesticus*) will produce nodes in the tree that are close to each other. As more distantly related strains are added (i.e. *M.m.domesticus* derived strains combined with *M.m.musculus* derived strains), the distance between SDPs becomes larger and tree construction times increase. Most existing eQTL studies in panels of inbred strains have used less than 40 strains (Bystrykh et al. 2005, Chesler et al. 2005, McClurg et al. 2007). Tree construction required 5.3 minutes for the 32 strains of the hypothalamus data set. In contrast, for a panel of 71 inbred strains derived from both *M.m.domesticus* and non-*M.m.domesticus* strains, tree construction requires ∼10 hours using a 1 SNP window and ∼24 hours using a 3 SNP window. Tree construction is carried out only once, and the resulting calculations still require less time than a brute force approach. Faster algorithms for tree construction that improve scalability with increasing numbers of strains are currently under investigation.

3.3.4 Differences between FastMap and Other QTL Software

We compared the eQTL results produced by FastMap to those produced by R/qtl. R/qtl was configured to use Haley-Knott regression (Haley & Knott 1992) and 1,000 permutations to determine significance thresholds. While R/qtl is designed to perform linkage mapping, we note that when linkage mapping is performed exclusively at the markers, the calculations are identical to those performed in eQTL (see Supplementary Materials). eQTLs may be broadly separated into two categories; eQTLs located within 1Mb of the transcript location (cis-eQTLs), and eQTLs located further than 1 Mb from the transcript location (trans-eQTLs).

Both FastMap and R/qtl found similar numbers of total eQTLs, *cis*-eQTLs and *trans*-eQTLs (Figure 3.4a). Figure 3.4b shows that the eQTL locations found by each software package are essentially identical; 98% of the eQTLs found by each method are within 5 Mb of each other, a margin of resolution consistent with the resolution of the BXD marker set. Since permutation based testing involves randomization, it should not be expected that 100% of the eQTLs would match between the two methods. Furthermore, the eQTL histograms produced by each method (Figure 3.4c & d) are similar, with differences being due to histogram binning effects (see insets).

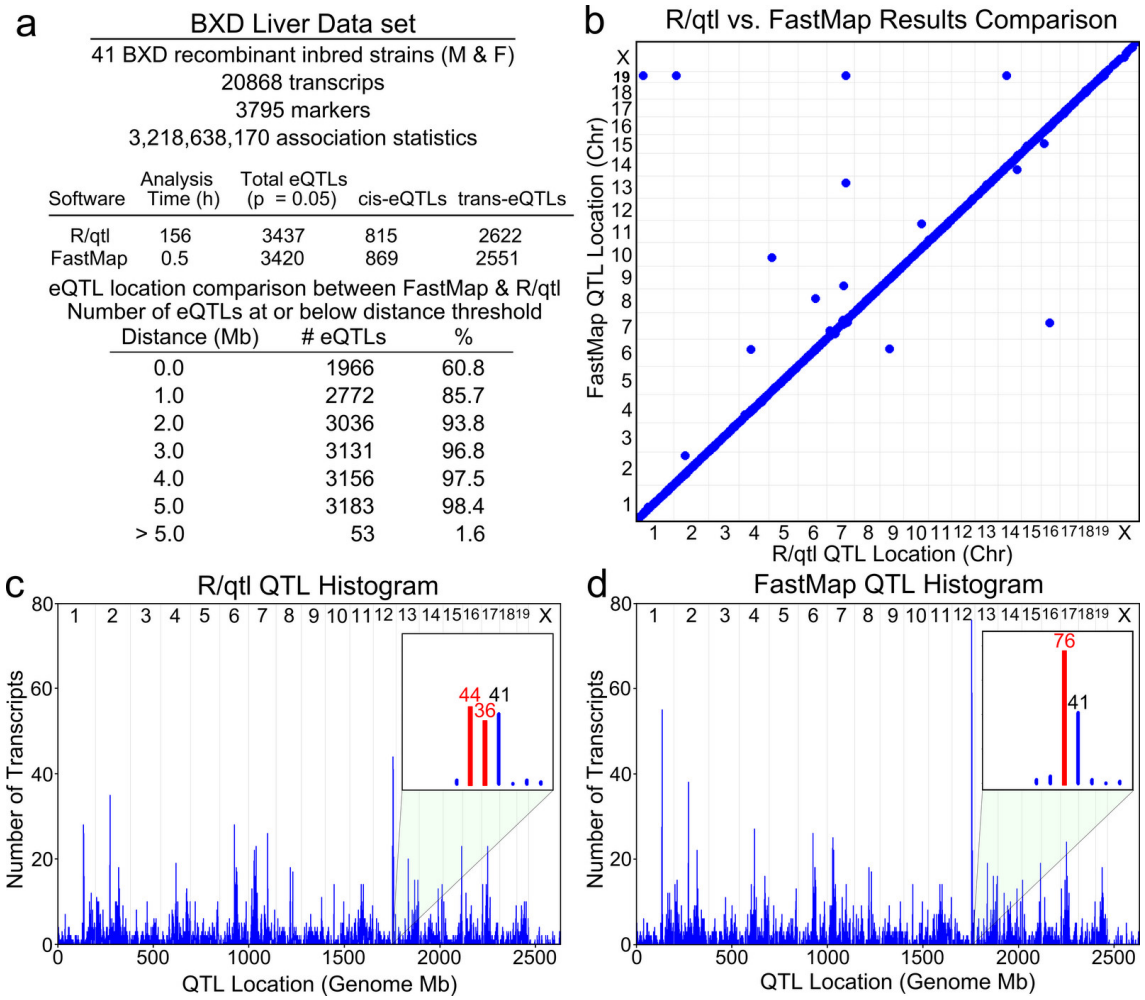

Figure 3.4: FastMap eQTL mapping results almost equivalent to those obtained with R/qtl. Panel (a) describes the BXD data set and shows the number of matching eQTLs between FastMap and R/qtl at varying distances. Panel (b) shows the high degree of concordance between FastMap and R/qtl. Panels (c) & (d) present eQTL histograms produced by FastMap and R/qtl. They are substantially equivalent with differences on Chr 1 and 12 being due to histogram binning effects (insets). Data is shown at 5% significance threshold.

eQTL mapping in the hypothalamus data set was performed to evaluate computational performance, rather than to compare the results with SNPster. However, it is natural to ask how the results of the two methods compare when we employ median centering in FastMap to correct for population stratification. We correct for population stratification by median centering transcript values within *M.m.domesticus* and non-*M.m.domesticus* derived strains.

Since SNPster does not provide a fixed threshold for significance, we selected 2,413 transcripts which had SNPster p-values less than 10^{-4}. Of these, 105 were *cis*-eQTLs and 2,308 were *trans*-eQTLs. FastMap produced eQTLs for 382 transcripts at or above a 0.05 significance threshold, of which 29 were *cis*-eQTLs and 353 were *trans*-eQTLs. The locations of 55 eQTLS were common between the two methods and all of these were *cis*-eQTLs, which have been reported to be more reproducible than *trans*-eQTLs (Peirce et al. 2006).

It should be noted that FastMap and SNPster differ in several important respects. SNPster uses a heuristic weighted F-statistic who's null distribution is not known, it employs a re-sampling approach that selects strains in a random manner with a non-uniform distribution. FastMap uses the standard F-statistic and conventional permutation-based significance thresholds. For these reasons, it is unclear whether the results of the two methods should be concordant, and biological validation of both eQTL mapping approaches may be necessary to address the differences.

3.3.5 Population Stratification

As noted by (McClurg et al. 2007), considerable population stratification is present when panels of laboratory inbred strains are used. Common laboratory inbred strains are a mixture of *M.m.domesticus*, *M.m.musculus*, *M.m.castaneus*, *M.m.molossinus* and *M. spretus*, which arose during the creation of the laboratory inbred strains (Beck et al. 2000, Yang et al. 2007). Figure 3.5 shows a SNP similarity matrix for the 32 inbred strains in the hypothalamus dataset, where each cell represents the proportion of SNPs that have the same allele between 2 strains (normalized Hamming distance) across all 156K SNPs. The non-*M.m.domesticus* derived strains cluster tightly in the lower left hand corner, indicating that they are more genotypicly similar to each other than to the *M.m.domesticus* derived strains. Numerous transcripts and SNPs exhibit systematic differences across these two strata. Consequently, each such transcript will

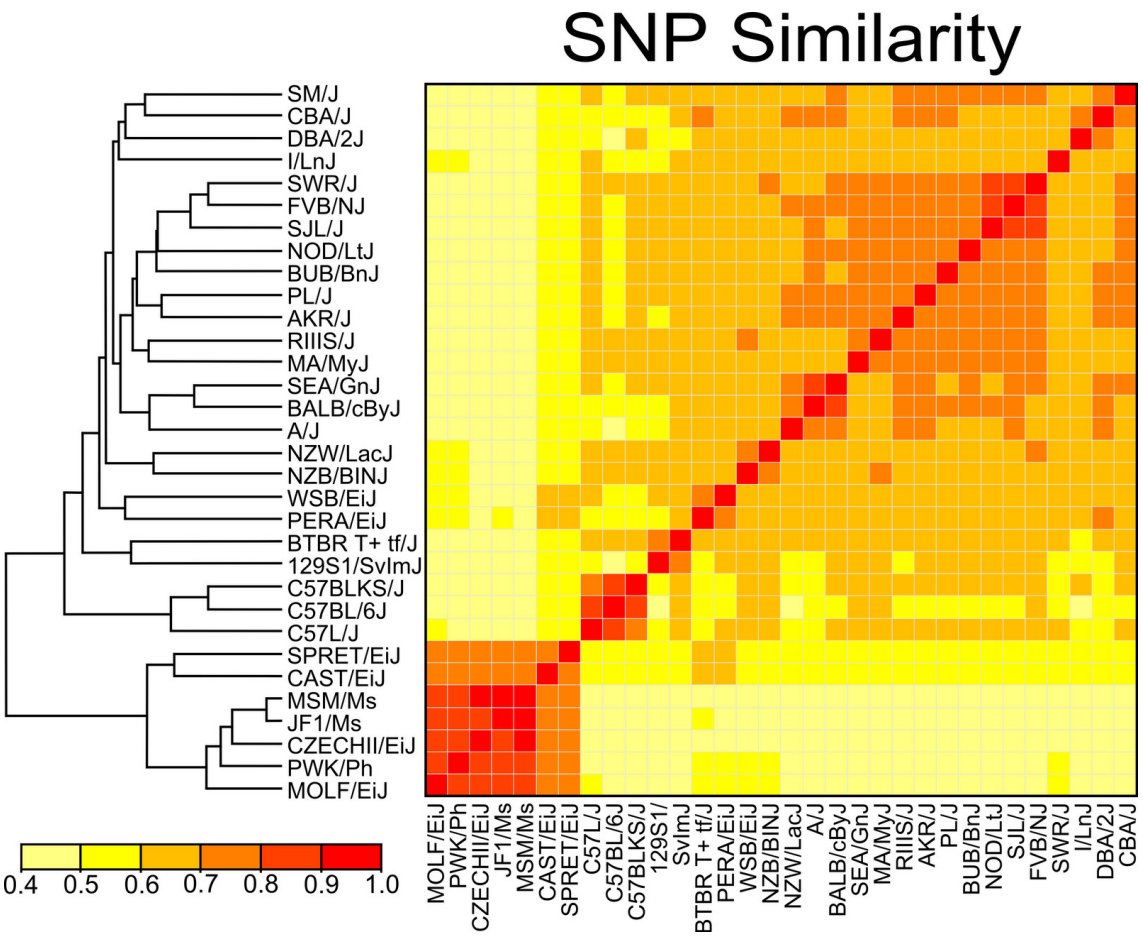

Figure 3.5: SNP similarity matrix demonstrates population stratification among laboratory inbred strains. In one row, each cell represents the proportion of SNPs (in the 156K data set) with the same allele in the other strains. The similarity matrix has been hierarchically clustered (distance = SNP similarity, linkage = average).

show a significant association with every such marker. In eQTL mapping, this produces numerous markers that show significant associations with the expression of a single transcript, leading to horizontal banding in the transcriptome map (Figure 3.6a,b). When such differences exist, most permutations of the transcript will yield a lower association statistic than the observed one, this leads to inappropriately low significance thresholds (Figure 3.6c). In order to remove this strata effect, we median center the values of each transcript within *M.m.domesticus* and non-*M.m.domesticus* strata. As shown in Figure 3.6d the resulting transcriptome map becomes interpretable with cis-eQTLs along the diagonal. The few horizontal bands that remain are due to a subset of the *M.m.musculus* derived strains with transcript expression levels that differ

from the other strains; this prevents the median subtraction method from removing the strata effect completely. We recommend removing those few transcripts that demonstrate this effect.

FastMap allows the user to select strata by genotype *a priori*, and subtracts strata means or medians from the transcript values in each stratum (Pritchard et al. 2000). While there are more sophisticated methods for addressing population stratification (Kang et al. 2008), FastMap is not primarily designed to address this problem. While laboratory inbred strains have been useful in mapping Mendelian traits, eQTL mapping with FastMap will have greater utility in well segregated populations like the Collaborative Cross (Churchill et al. 2004, Roberts, Pardo-Manuel de Villena, Wang, McMillan & Threadgill 2007), due to increased genetic diversity, as well as the finer recombination block structure. In such well-mixed populations, mean/median subtraction within strata or the non-uniform resampling technique used by SNPster should not be required.

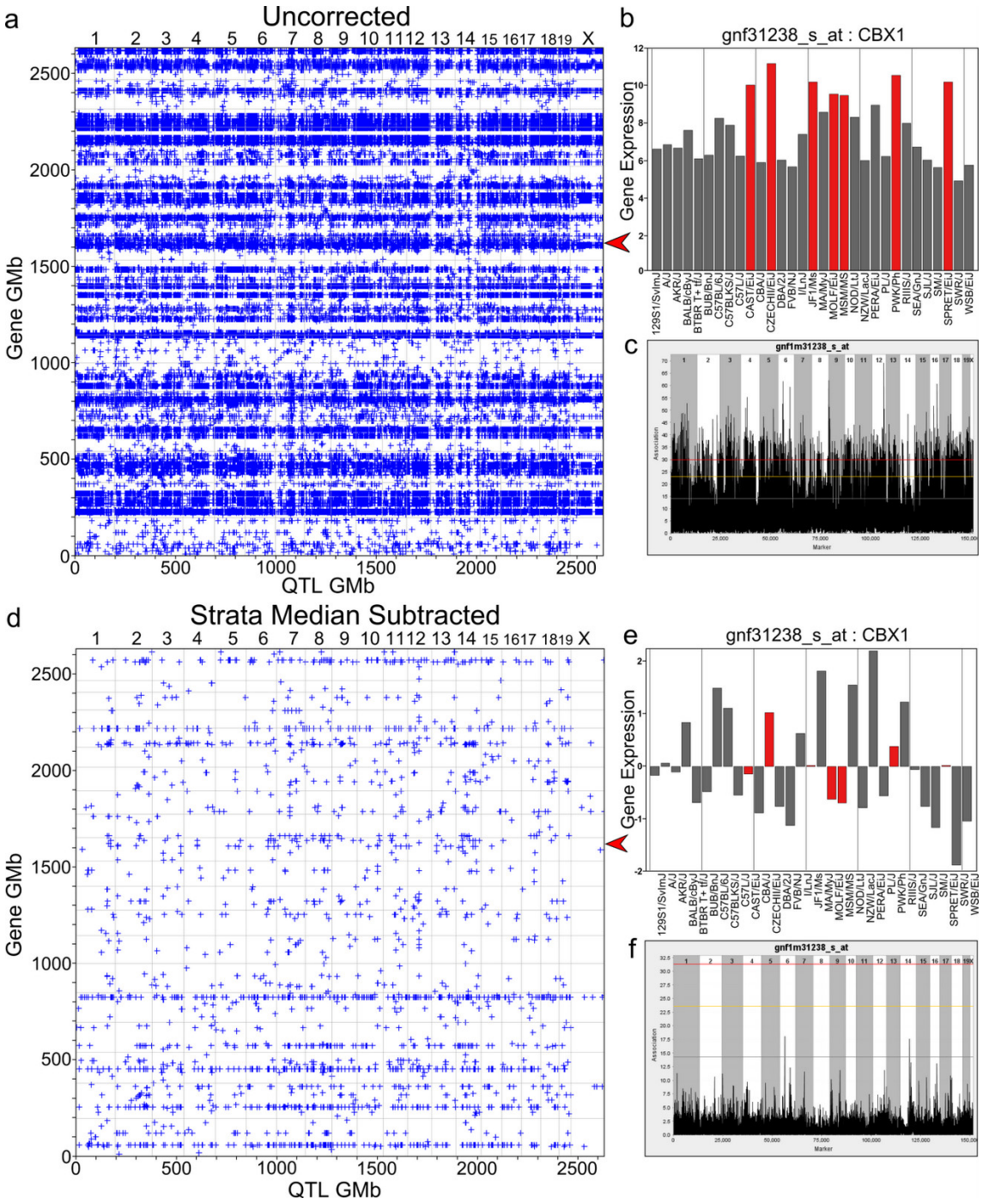

Figure 3.6: Strata median correction dramatically improves transcriptome map. Panel (a) shows the transcriptome map for 3,270 transcripts without correcting for the population structure for all eQTL above a transcript-specific 5% significance threshold. The horizontal bands dominate the plot and are due to gene expression profiles like the one in panel (b), which is marked by the red arrow in (a). The grey colored strains are the *M.m.domesticus* derived strains and the red ones are the non-domesticus derived strains. By subtracting out the strata median from each strata, the transcriptome map (d) is greatly improved. Gene expression values are no longer split by genotypic strata (e) and the permutation derived thresholds are appropriate (f).

CHAPTER 4

Merging Two Gene Expression Studies via Cross Platform Normalization

Summary

Motivation: Gene expression microarrays are currently being applied in a variety of biomedical applications. This chapter considers the problem of how to merge data sets arising from different gene-expression studies of a common organism and phenotype. Of particular interest is how to merge data from different technological platforms.

Results: The chapter makes two contributions to the problem. The first is a simple cross-study normalization method, which is based on linked gene/sample clustering of the given data sets. The second is the introduction and description of several general validation measures that can be used to assess and compare cross-study normalization methods. The proposed normalization method is applied to three existing breast cancer data sets, and is compared to several competing normalization methods using the proposed validation measures.

Availability: The Supplementary Materials and XPN Matlab code are publicly available at website: https://genome.unc.edu/xpn

4.1 Introduction

High-throughput gene expression microarrays are currently being applied in a wide variety of biomedical problems. There are now several widely used, commercially available, microarray platforms that measure gene expression in related, but different, ways. No matter which technology is used, the evaluation of gene expression experiments usually begins with statistical analyses that take a variety of forms, including exploratory analysis (such as clustering),

classification, and assessments of differential expression.

The increasing number and availability of large scale gene expression studies of human and other organisms provides strong motivation for cross-study analyses that combine existing and/or new data sets. In a cross-study analysis, the data, relevant test statistics, or conclusions of several studies are combined. The simultaneous analysis of different studies of a common organism and phenotype has the potential to strengthen and extend the results obtained from the individual studies. Cross-study analyses can be carried out using existing data sets, so their results hold out the promise of comparatively inexpensive, scientific "value-added".

On the other hand, combining data from different expression studies poses a number of statistical difficulties. These difficulties arise from the fact that the constituent data sets have often been produced using different gene expression platforms and different processing facilities. As a consequence, measurements from different platforms cannot be directly combined. Identifying and removing such systematic effects is the primary statistical challenge in cross-study analysis. We note that technological differences between studies may be confounded with biological differences arising from the choice of patient cohorts (e.g. age, gender or ethnicity). In many cases, technological artifacts are dominant, though care should be taken to verify this, and one can hope to remove them while leaving biological information intact.

There are several potential approaches to cross-study analysis, depending on what information is being synthesized. At the highest level, one may wish to combine, through meta-analysis or other techniques, the broad conclusions of different studies. Most existing work on multi-study gene expression analysis is focused on an intermediate level, where the goal is to combine information from primary statistics (such as t-statistics or p-values) or secondary statistics (such as gene lists) that are derived from the individual studies (Choi et al. 2003, Garrett-Mayer et al. 2004, Ghosh et al. 2003). Other approaches to meta-analysis of gene expression data are considered by (Garrett-Mayer et al. 2007, Parmigiani et al. 2004, Shen et al. 2004, Rhodes et al. 2002, 2004). This chapter deals with the problem of cross-study normalization: how to combine two available data sets in order to produce a single, unified data set to which standard statistical procedures (such as clustering, classification and measures of differential expression) can be applied.

There has been a great deal of work on the normalization of gene-expression data within a

single study (Bolstad et al. 2003, Irizarry et al. 2003, Irizarry, Hobbs, Collin, Beazer-Barclay, Antonellis, Scherf & Speed 2003, Yang et al. 2002). Much of that work can be applied, with little modification, to normalizing data from multiple studies that are based the same technological platform. The emphasis here is on the problem of combining data from different array platforms. We will use the term cross-platform normalization when this distinction is important.

4.2 Cross Platform Normalization (XPN) method

Here we describe the basic idea behind the XPN (cross platform normalization) method. We restrict our attention to merging two studies; the model and fitting procedure can be extended in a natural way to handle three or more studies.

XPN takes as input the gene expression measurements from two studies, after appropriate preprocessing and imputation. One may work with the set of common genes in the studies, or on a selected subset of these genes. Once an appropriate set G of genes has been identified, the available data can be represented as two matrices

$$X_p = \{x_{gsp} : g \in G, s = 1 \ldots n_p\} \quad p = 1, 2. \tag{4.1}$$

Here X_p denotes the available data from study p, and $x_{g,s,p}$ is the expression of gene g in sample s of study p. Let n_1 and n_2 denote the number of samples in studies 1 and 2, respectively, m denote the number of genes in G. The normalized data can be represented similarly, as two matrices $\tilde{X}_p = \{\tilde{x}_{gsp} : g \in G, s = 1 \ldots n_p\}$ with the same dimensions as X_1 and X_2.

4.2.1 Block Linear Model

The XPN procedure is based on a simple block-linear model. In this model, the observed value x_{gsp} is a scaled and shifted block mean plus noise. The block mean is constant over a range of gene and sample values, and is the same in each platform. The slope and offset of the linear transformation, as well as the variance of the noise, depend on the gene g and the platform p. More precisely, we assume that

$$x_{gsp} = A_{\alpha^*(g),\beta_p^*(s),p} \cdot b_{gp} + c_{gp} + \sigma_{gp}\, \varepsilon_{gsp}. \tag{4.2}$$

The functions $\alpha^* : \{1, ..., m\} \mapsto \{1, ..., K\}$ and $\beta_p^* : \{1, ..., n_p\} \mapsto \{1, ..., L\}$, $p = 1, 2$, define linked groups of genes and samples, respectively. The numbers A_{ijp} are block means, while b_{gp} and c_{gp} represent sensitivity and offset parameters, respectively, that are specific to each gene and platform. The noise variables ε_{gsp} are independent standard normals, so the final term in (4.2) has variance σ_{gp}^2. The model reflects the assumption that the samples of each available study fall roughly into one of L statistically homogenous groups, and that each group is defined by an associated gene profile that is constant within each of K groups of similar genes. The block means $\{A_{i,j} : i = 1, \ldots, K\}$ represent the profile of the jth group. Figure 4.1 illustrates the underlying block structure. Note that the basic studies may be of different sizes. A heatmap illustrating the same idea on real data is provided in the Supplementary Materials.

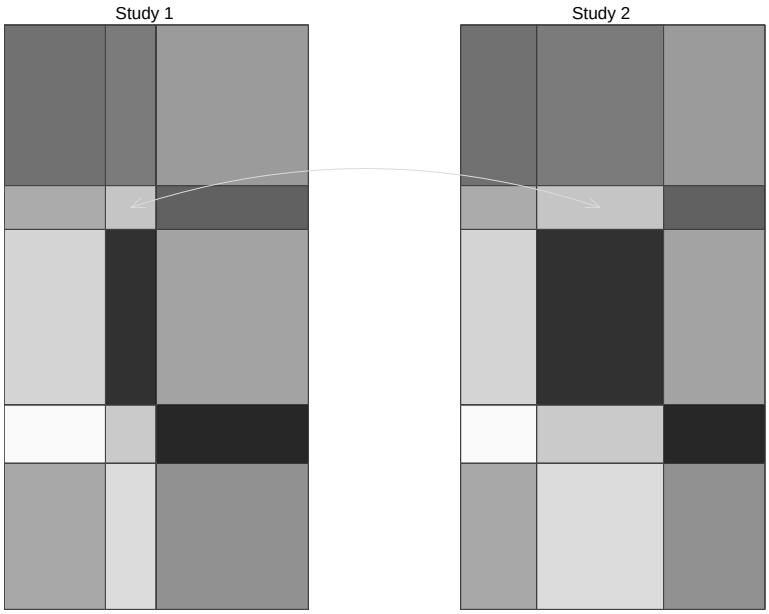

Figure 4.1: Studies 1 and 2 after row and column clustering of their combined data, with $K = 5$ gene groups and $L = 3$ sample groups. Shading indicates linked gene-sample blocks.

4.2.2 Description of XPN

Initially, the data from the available studies are sample standardized and gene median centered, in order to remove gross systematic differences, and then combined. Following the model (4.2), clustering is then used to identify homogenous groups of genes and samples in the combined data matrix. Specifically, k-means clustering is applied independently to the rows and columns

of the combined data matrix, using $k = K$ gene clusters and $k = L$ sample clusters, respectively. Application of k-means begins with a random choice of centroids for the clusters. In clustering rows, we select K rows of the data matrix at random, and use these as the initial centroids. Cluster assignments and centroids are then updated iteratively until convergence to a local minimum of the sum of squared Euclidean distances. A similar procedure is used for clustering of the columns.

The gene clusters in the combined data matrix are summarized by the assignment function $\alpha : G \to \{1, \ldots, K\}$. Gene clusters are naturally linked across studies, as we work with the same genes in each study. The column clusters in the combined data matrix are summarized by assignment functions $\beta_p : \{1, ..., n_p\} \mapsto \{1, ..., L\}$ for $p = 1, 2$. Specifically, $\beta_p(s)$ is the index of the combined sample cluster containing sample s from Study p. The ℓ'th combined cluster splits into linked clusters $\{s : \beta_1(s) = \ell\}$ in Study 1 and $\{s : \beta_2(s) = \ell\}$ in Study 2.

From the mappings $\alpha(g)$ and $\beta_p(s)$, estimates of the model parameters $\hat{A}_{ijp}$, $\hat{b}_{gp}$, $\hat{c}_{gp}$ and $\hat{\sigma}_{gp}$ are obtained using standard maximum likelihood methods. Details are given in the Appendix. Common model parameters $\hat{\theta}_g = (\hat{b}_g, \hat{c}_g, \hat{\sigma}_g^2)$ and $\hat{A}_{ij}$ are then calculated as weighted averages of the parameters in Study 1 and Study 2:

$$\hat{\theta}_g \;=\; \frac{n_1\hat{\theta}_{g,1} + n_2\hat{\theta}_{g,2}}{n_1 + n_2}, \qquad \hat{A}_{ij} \;=\; \frac{n_{j,1}\hat{A}_{i,j,1} + n_{j,2}\hat{A}_{i,j,2}}{n_{j,1} + n_{j,2}},$$

where $n_{j,p}$ is the number of samples in the jth sample group of platform p. The expression values of each platform are then modified in accordance with the estimated model parameters to produce normalized values

$$x^*_{gsp} \;=\; \hat{A}_{\alpha(g),\beta_p(s)}\,\hat{b}_g \;+\; \hat{c}_g \;+\; \hat{\sigma}_g \left(\frac{x_{gsp} - \hat{A}_{\alpha(g),\beta_p(s),p}\,\hat{b}_{gp} - \hat{c}_{gp}}{\hat{\sigma}_{gp}} \right).$$

The output of the XPN algorithm is based on multiple clusterings of the data. The procedure described above is applied 30 times, with different randomly chosen initial centroids for the row and column clusters. The output of the algorithm is the average of the normalized values obtained over the repeated runs.

There are several reasons for averaging the results of multiple clusterings of the combined

data matrix. To start, there is unlikely to be a single, "biologically correct" clustering of the available genes and samples: disease subtypes and gene pathways are not always uniquely defined, and they may exhibit moderate overlap. Multiple clusterings better capture the structure present in this situation. By combining normalization results from multiple clusterings (each of which yields a local minimum of the sum of squares cost function) the XPN algorithm performs a simple form of model averaging. Averaging also controls (minor) instability that may arise from use of the k-means clustering procedure, whose output is dependent on the initial choice of cluster centroids. In this latter respect, XPN is similar in spirit to resampling based approaches to cluster stability such as those in (Tseng 2007, Tseng & Wong 2005, Dudoit & Fridlyand 2002, Tibshirani et al. 2001).

In principle, the XPN method procedure can be used with any clustering method that produces a pre-specified number of clusters from a given set of vectors, or with resampling based improvements of such methods. We chose to use k-means clustering because of its simplicity and computational efficiency. The validation study below indicates that the XPN method performs well, and generally outperforms competing normalization methods, when it is used with basic k-means clustering. The validation results leave open the possibility of further improvements with alternative clustering methods, but a number of experiments with other clustering methods have not produced better results.

In the current implementation of XPN, the number of row and column clusters, $K \geq 1$ and $L \geq 1$ respectively, are fixed in advance, and will depend on the type and dimension of the data under study. In general, L should be large enough to capture principal sample groups or subtypes, and L should be large enough to capture large, homogenous groups of genes. In the numerical experiments below we chose $K = 5$ and $L = 25$. (In practice, XPN is not sensitive to the choice of K and L, see Section 4.6.1 below.) As an alternative, one may employ a method such as the GAP statistic (Tibshirani et al. 2001), implemented as an R function `kmeansGap` in library "`SLmisc`", to assess the number of row and column clusters in the data. Applied to the data set used in this chapter , the GAP statistic suggested 4-8 sample clusters and 8-9 gene clusters.

4.3 Other Methods

We compare XPN with several other normalization methods in the literature. The other methods have previously been applied to batch correction on single platforms, but are well adapted to more general cross-study situations. As a baseline, we standardized each available column (sample) (CS). Beginning with CS data, we median centered each gene in each study and then combined studies. The resulting procedure is denoted by (MC). The MC method is currently used in practice, and in spite of its simplicity, performs relatively well in our validation experiments. We also consider the Empirical Bayes (EB) method (Johnson et al. 2007). EB is based on the model

$$x_{gsp} \;=\; a_g + \gamma_{gp} + \delta_{gp}\,\sigma_g\,\varepsilon_{gsp}, \qquad \varepsilon_{gsp} \;\sim\; N(0,1).$$

The platform specific parameters γ_{gp} and δ_{gp} are estimated using an empirical Bayes approach, and are essentially equal to OLS estimates shrunken towards their respective cross platform means. Other parameters are estimated by gene-wise OLS. The data is then transformed to remove the effects of different γ_{gp} and δ_{gp} across platforms. Finally, we considered the DWD method for batch correction (Benito et al. 2004), which is based on the Distance Weighted Discrimination method (Marron & Todd 2004). DWD normalization finds a direction in which the sample-vectors from the two studies are well-separated, and then translates the samples from each study along that direction until their respective families of vectors have significant overlap.

The Probability of Expression (POE) method (Shen et al. 2004, Parmigiani et al. 2002), transforms each data value into a signed probability in the range $[-1, 1]$. While this transformation is useful for identifying meta-signatures, the resulting data is difficult to compare with normalized values produced by other methods, and we do not include its analysis here.

We note that each of the alternative normalization methods described above is gene-wise affine, that is, for each gene g there exist constants a_g and b_g, with $a_g > 0$, such that $\tilde{x}_{s,g} = a_g x_{s,g} + b_g$. As a result, the correlation between $x_{s,g}$ and $\tilde{x}_{s,g}$ across samples s is 1 for every g. By contrast, XPN seeks to simultaneously borrow strength across genes and samples via linked row and column clusters, and as a result, XPN is not gene-wise affine.

4.4 Data Sets and Preprocessing

We applied XPN and the methods described above to three existing breast cancer data sets. The first data set, from (Huang et al. 2003), has 89 samples and 8948 genes. Their experiments were performed with Affymetrix GeneChip U95Av2 arrays. The 89 samples were obtained at the Koo Foundation Sun Yat-Sen Cancer Centre (KF-SYSCC), Taipei. The second data set, which will be referred to as NKI, comes from (van 't Veer et al. 2002). It contains 97 samples and 16360 genes, and was obtained from Netherlands Cancer Institute and Rosetta Inpharmatics-Merck custom designed 25K Agilent oligonucleotide arrays. Most of the NKI patients had stage I or II breast cancer. The third data set, referred to as UNC, is from (Hu et al. 2006). It contains 114 samples representing 104 patients and 12065 genes, and was obtained using 22K Agilent oligonucleotide arrays. The UNC sample set represents an ethnically and geographically diverse cohort.

Initially, LOWESS normalization was applied to the NKI and UNC data sets; RMA was used to obtain expression values for the Huang data set. The raw expression values in each study were then log-2 transformed, and missing values were imputed with 1-nearest neighbor imputation (Troyanskaya et al. 2001*b*). Duplicated genes in each data sets were collapsed by median using Entrez Gene ID. There were 6092 common genes among the three platforms. Cross-study normalization methods were applied to this set of common genes, and subsequently to a smaller set of "intrinsic genes" (Perou et al. 2000) identified as playing an active role in the biology of breast cancer.

The next section presents validation results for the set of common genes. The same analysis for the set of intrinsic genes is presented in the Supplementary Materials. In our experiments, all cross-platform normalization methods worked better on the set of intrinsic genes, and more generally, on smaller gene sets selected using integrative correlation filtering. Prior to cross-study normalization, the log-2 transformed expression values in each platform were column standardized.

4.5 Validation

Broadly speaking, cross-study normalization methods can be assessed in terms of two competing criteria. Ideally, a normalization method should produce a single unified data set, in which samples originating in Study 1 are not distinguishable from those originating in Study 2 on the basis of non-biological features. A method that fails to remove systematic differences between studies under-corrects the data. On the other hand, excessive homogenization of the studies (over-correction) can result in a loss of biological information, and the combined data set may be less useful than its constituents.

The validation results presented below are intended to assess the performance of the methods under study, and their tendency towards over- and under-correction. We begin with the column-standardized data sets X_1, X_2 and X_3. Every method is applied to each pair X_i, X_j with $1 \leq i < j \leq 3$ to produce normalized data $\tilde{X}_{i,j} = [\tilde{X}_i, \tilde{X}_j]$. Validation measures are applied to each pair, and the average value of the measure over the three pairs is reported. For before and after comparisons, we take as a reference the initial data $[X_i, X_j]$ produced by column-standardization (denoted CS in what follows).

In order to better understand the baseline behavior and biases of the normalization methods under consideration, we also apply them to artificial studies obtained by randomly dividing the arrays in a given platform into two pseudo-studies, similar to the procedure in (Gentleman et al. 2006). To be more precise, from a single column-standardized data set X_i, we produce a pair X_i^1, X_i^2 of pseudo-studies by randomly assigning each sample to one of two groups. Different normalization methods are then applied to $[X_i^1, X_i^2]$, yielding a normalized data sets $\tilde{X}_i = [\tilde{X}_i^1, \tilde{X}_i^2]$. Validation measures are applied to compare the pseudo-study and its normalized version. Each of the three available data sets is randomly split ten times, and the average measure (over splits and studies) is reported.

By design, the data in each pair of pseudo-studies come from a common platform and study. Thus we anticipate that a cross-study normalization method should have relatively little effect, beyond its attempt to correct the unavoidable differences that result from splitting the studies in half. While these differences are not negligible, they are typically smaller than the differences between platforms.

4.5.1 Measures of Center and Spread

For a given array, the difference between the mean and the median of its values provides a rough measure of its asymmetry in regards to location. After normalization, it is desirable to see a similar distribution of asymmetry across both studies. Figure 4.2 shows the area between the CDFs of mean minus median in the two available studies. Graphs for both standard and split study validation are shown.

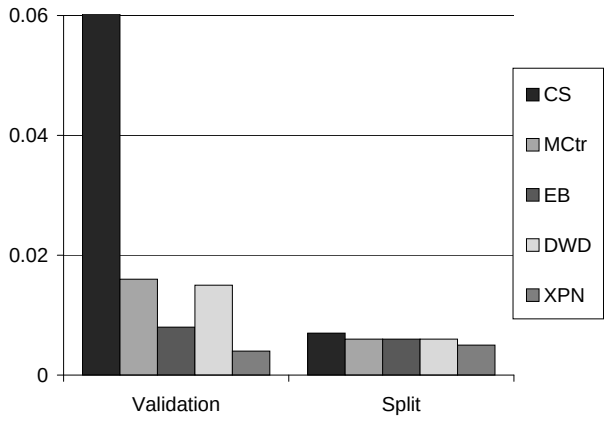

Figure 4.2: Area between the CDFs of array mean minus array median across platforms. Lower values indicate greater similarity of datasets after normalization.

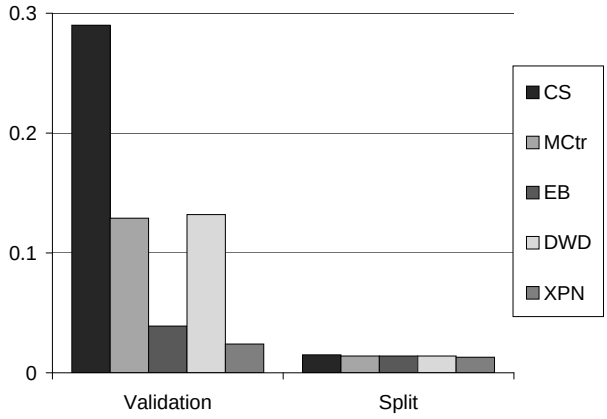

Figure 4.3: Area between the CDFs of $\sigma - MAD/\Phi(0.75)$ for arrays of different platforms. Lower values indicate greater similarity of datasets after normalization.

A similar comparison for scale can be carried out by considering the standard deviation (σ) and median absolute deviation from median (MAD). For the standard normal distribution with CDF Φ, we have $\sigma = MAD/\Phi(0.75)$. Figure 4.3 shows the area between CDFs of $\sigma -$

$MAD/\Phi(0.75)$ in each of the two available studies. XPN reduces both measures more than the other methods; the split study results show little bias for all methods.

4.5.2 Average distance to nearest array in another platform

The set of arrays in given platform can be viewed as a set of points in m-dimensional Euclidean space. After normalization it is reasonable to expect that the point "clouds" associated with distinct platforms will have substantial overlap. (This is one of the motivations behind the DWD normalization method.) To measure overlap in a pair of normalized studies, we measure the Euclidean distance from each array in the first study to the nearest array in the second study, then repeat, swapping the roles of the studies, and finally average the results. The results are presented in Figure 4.4, with smaller values indicating greater overlap. XPN and EB reduce the average distance more than other methods. The split study results show little bias for all the methods.

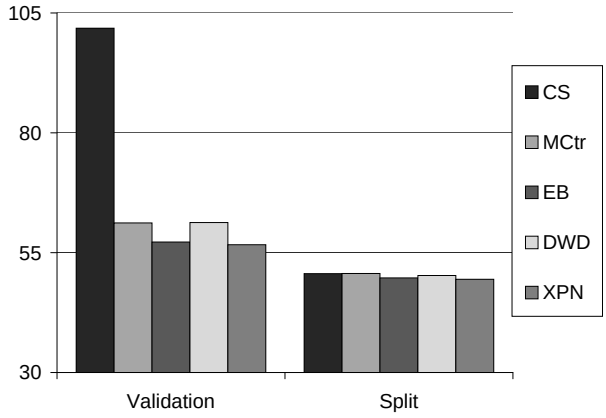

Figure 4.4: Average L_2 distance from the samples of one study to the nearest sample from the other study. Lower values indicate greater similarity of the study point "clouds" after normalization.

4.5.3 Correlation with Column Standardized Data

The previous validation measures assess the similarity of two data sets after normalization. A natural way to see how much the normalization methods affect the data is to calculate correlation between the data matrices before and after normalization, where "before" is represented by CS. This measure does not by itself support a given normalization method, but in choosing between methods that perform similarly across other validation measures, the method that has

less effect on the data should clearly be preferred. The average correlation of arrays before and after normalization for the different methods under study is shown in Figure 4.5. Median centering has the least effect on the data; the other three methods yield average correlations close to 0.8, with XPN lying between DWD and EB. Table 4.1 shows the average correlation of genes before and after normalization, averaged over both studies. As discussed above, all methods but XPN perform normalization by transforming each gene in an affine fashion; thus the gene correlation for these methods is equal to 1. Similar remarks apply to the integrative correlation and t-statistic measures described below. The gene correlation for XPN is .99, with a split-study value of .996.

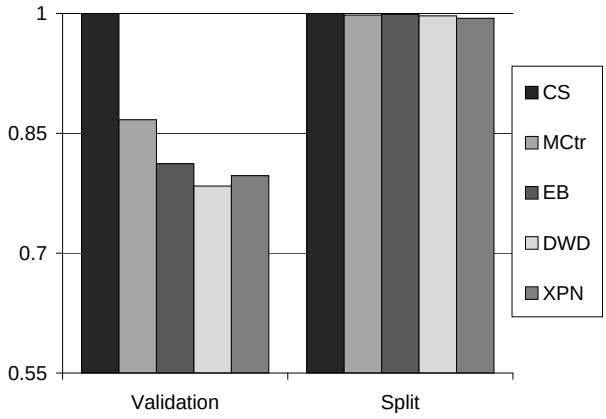

Figure 4.5: Average correlation of arrays with their values before normalization (CS). Larger values indicate less modification of the data by the normalization procedure.

4.5.4 Global Integrative Correlation

Integrative correlation (Cope et al. 2007) is a means of identifying genes with concordant expression in different studies. Let $r_1(g), r_2(g)$ be the g'th row of X_1 and X_2, respectively. The global integrative correlation (GIC) between X_1 and X_2 is the correlation between

$$(\,\mathrm{corr}(r_1(g), r_1(g')) : g, g' \in G\,) \quad (\,\mathrm{corr}(r_2(g), r_2(g')) : g, g' \in G\,),$$

here regarded as vectors with $|G|^2$ components. High values of $IC(g)$ indicate good concordance between the values in Studies 1 and 2. Global integrative correlations for different normalization methods are shown in Table 4.1. The results for CS shows that the average GIC between halves

of the same platform (0.556) is much higher then average GIC between different studies (0.255). XPN is the only method among those considered that affects GIC. It increases GIC by 33% to 0.338 in cross study validation, well below the split study level (0.556). XPN increases GIC between pseudo studies by a relatively small 7%.

Each tumor sample in the data sets under consideration has an associated, clinically based ER status (ER+ or ER-). We next consider several validation measures based on this biological information. The Huang data set has only 15 ER negative samples out of 89, making its split study results unstable, and is therefore excluded from the split study analysis of the ER based validation measures.

4.5.5 Correlation of t-statistics

For each platform, t-statistics measuring the association of gene expression values with the ER status are calculated. Ideally, the vectors of t-statistics for different platforms should become more concordant after platform normalization. Table 4.1 shows the Pearson correlation between the t-statistics for ER status for different normalization methods. (Results for rank correlation are similar.) As expected, the average correlation of t-statistics is higher in split study (0.446) than between platforms (0.312). XPN increases the correlation of t-statistics between platforms by 45% to 0.451. In split study validation it increased correlation by roughly 22%. Overall, XPN has greater effect than the other methods considered. The correlation measurements above show that, on average, XPN does not make dramatic changes in the rows of the data matrices, and we believe that much of the split study increase in t-statistic correlation is due to inherent differences between the randomly selected pseudo studies.

4.5.6 Cross platform prediction of ER status

If we regard ER status as a binary phenotype, we may explore misclassification rates associated with its prediction. Ideally, combining labeled studies via cross-platform normalization should lead to lower misclassification rates on test data sets. To test the compatibility of different studies after normalization in regards to classification, we treated the data from one study as a training set, and the data from the other study as a test set, and *vice versa*. Lower error rates indicate better concordance. Classification was performed using two methods: nearest

		CS, MCtr, EB, DWD	XPN	Change	Change (%)
Avg gene corr w/ CS	Valid'n	1.000	0.990	-0.010	-1.0%
	Split	1.000	0.996	-0.004	-0.4%
GIC	Valid'n	0.255	0.338	0.083	33%
	Split	0.556	0.597	0.041	7%
ER t-stat correlation	Valid'n	0.312	0.451	0.139	45%
	Split	0.446	0.543	0.096	22%

Table 4.1: The first row shows the average correlation of genes with their value before normalization (CS). The second row shows global integrative correlation (GIC) between platform pairs after normalization, with larger values indicating better concordance between platforms. The third row shows the average correlation of ER t-statistics across platforms, with larger values indicating better concordance.

shrunken centroids (PAM) (Tibshirani et al. 2002) and support vector machines (SVM) (Boser et al. 1992, Cortes & Vapnik 1995). The results are presented in Figures 4.6 and 4.7. As can be seen, all of the normalization methods greatly reduce cross platform prediction error, with the minimum error achieved by XPN. In the split study test, none of the methods produces significant reductions in classification error, as expected.

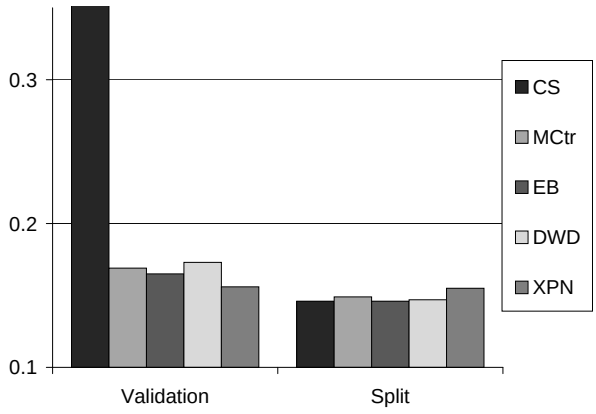

Figure 4.6: Cross platform prediction error of the PAM (nearest shrunken centroids) classifier. Smaller values indicate better concordance between platforms.

One might also be interested in the 5- or 10-fold cross validation prediction error rate on the combined studies. However, none of the normalization methods has a significant effect on the cross validated classification error. This appears to arise from the fact that, in cross validation, the classification methods are trained on elements of both platforms, and the distinguishing features of ER status are strong enough to enable the methods to perform well without prior

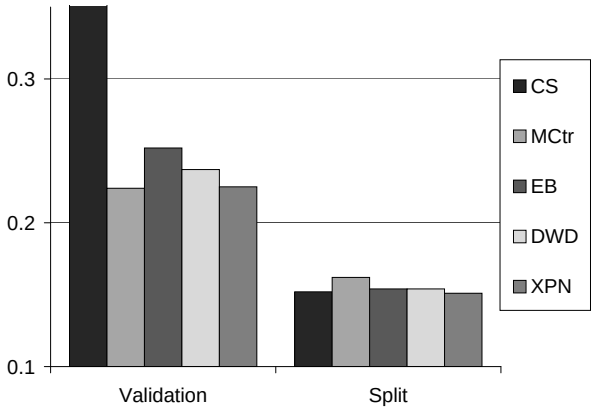

Figure 4.7: Cross platform prediction error of the SVM (Support Vector Machine) classifier. Smaller values indicate better concordance between platforms.

		CS	MCtr	EB	DWD	XPN
V_1	Valid'n	0.826	1	1	1	1
	Split	1	1	1	1	1
V_2	Valid'n	0.646	0.895	0.774	0.887	0.759
	Split	0.876	0.867	0.875	0.878	0.870

Table 4.2: V_1 (V_2) is the fraction of genes from the intersection (union) of platform-specific gene lists present in the list produced from the combined data $\tilde{X}$ at 0.1% level.

normalization.

4.5.7 Preservation of Significant Genes

Lastly, we consider gene lists produced using ER-based t-statistics at a nominal 0.1% significance threshold. Let L_i be the list of genes in Study $i = 1, 2$, and let $L_{1,2}$ be the list produced at the same nominal 0.1% level from the combined data $\tilde{X}$. Ideally, genes that are in both L_1 and L_2 should appear in $L_{1,2}$, and most genes that appear in at least one of the single study lists will be in the joint list. We assess these two types of overlap by measures $V_1 = |(L_1 \cap L_2) \cap L_{1,2}|/|L_1 \cap L_2|$ and $V_2 = |(L_1 \cup L_2) \cap L_{1,2}|/|L_1 \cup L_2|$, respectively. The results are presented in Table 4.2. The value of V_1 is 1 for all normalization methods except CS, showing the importance of platform normalization. The V_2 measure is increased by all methods, with the greatest increase achieved by MC and DWD.

4.6 Further discussion of XPN

4.6.1 Stability with respect to K and L parameters

To test stability of XPN with respect to the numbers K and L of row and column clusters, we applied XPN with a range of parameters. For $L = 5$ we tried $K = 2, 10, 20, 25, 30, 50, 100, 500$, and for $K = 25$ we tried $L = 2, 4, 5, 6, 7, 8, 10$. The results (presented in the Supplementary Materials) indicate that XPN is generally insensitive to the choice of the K and L. However, we do see (expected) degradation of performance in situations where K or L is below 4, in which case the clustering is too coarse to adequately capture homogenous blocks of samples or genes. At the other extreme, when L is large, one finds column clusters containing samples from a single platform. For such clusters the algorithm cannot combine information across platforms, and its results will be degraded accordingly. (In its current implementation, XPN excludes such clusterings from the average that forms its output.) Values of K larger than 25 make the algorithm slower and do not substantially improve its performance.

4.6.2 Stability of XPN output

The XPN algorithm averages the normalization results from B row/column clusterings. To assess the stability of XPN, we calculated the standard deviation of each element in the normalized matrix over the $B = 100$ runs of the basic procedure. The average standard deviation (over all elements and platform pairs) was 0.004. By contrast, the average standard deviation of the entries of the normalized matrices was 0.79. Thus, the variability of the normalized entries due to random clusterings was, on average, two orders of magnitude less than the variability between the final normalized entries.

4.7 Conclusion

The increasing number and public availability of large-scale gene expression studies provides impetus for cross-study analyses that combine existing, and potentially new, data sets. Properly combined data sets give researchers more power for biological and statistical analysis. In this chapter we propose a new, block model based method, called XPN, for cross-platform normalization. The block model distinguishes XPN from other platform normalization methods

such as DWD and EB, which are gene-wise linear.

We propose a set of validation measures for comparison of different normalization methods. The validation measures can be roughly split in two groups. One group assesses the ability of normalization methods to remove systematic differences across platforms, while the other measures how much the data is transformed by normalization procedures. Based on the proposed validation measures, XPN successfully combined three existing breast cancer data sets without incurring substantial overfitting. In particular, cross-platform ER prediction error rates indicate that XPN successfully preserved biological information while removing systematic differences between platforms.

The XPN method has three parameters: the number of row and column clusters (K and L) and the number of basic iterations B. Our experiments indicate that the results of XPN are robust to the choice of K and L (see Section 4.6.1). The analysis in Section 4.6.2 suggests setting $B = 30$ is sufficient for stable output.

4.8 Maximum Likelihood Estimation of the Model

The XPN algorithm estimates the parameters of the model (4.2) using maximum likelihood approach. The model has distinct sets of parameter for different gene clusters and different platforms. Thus the problem of parameter estimation can be split into $2K$ smaller tasks. Fix $i \in \{1, \ldots, K\}$ and $p \in \{1, 2\}$. The log-likelihood function associated with gene group i and platform p can be expressed as

$$\begin{aligned} 2l_{i,p} \;=\; C \;+ \sum_{(s,g):\alpha(g)=i} \ln(\sigma_{gp}^2) \;+ \\ \sum_{(s,g):\alpha(g)=i} (x_{gsp} - A_{i,\beta_p(s),p}\, b_{gp} - c_{gp})^2/\sigma_{gp}^2. \end{aligned}$$

To ensure identifiability of the coefficients $\{A_{ijp}\}$ and $\{b_{gp}\}$, we set

$$\sum_{j=1}^{L} A_{ijp} \;=\; 0, \quad \sum_{j=1}^{L} A_{ijp}^2 \;=\; L \;\text{ and } \sum_{g:\alpha(g)=i} b_{gp} \;>\; 0.$$

The parameters A_{ijp}, b_{gp}, c_{gp}, and σ_{gp}^2 are chosen to maximize the log-likelihood. To find them we take first derivative of the log-likelihood with respect to these parameters and set the

result equal to zero:

$$\begin{aligned}
dl/dc_{gp} &= 0 = \textstyle\sum_s (x_{gsp} - A_{i,\beta_p(s),p} b_{gp} - c_{gp}) \\
dl/db_{gp} &= 0 = \textstyle\sum_s A_{i,\beta_p(s),p} (x_{gsp} - A_{i,\beta_p(s),p} b_{gp} - c_{gp}) \\
dl/dA_{ijp} &= 0 = \sum_{(g,s):\beta_p(s)=j} b_g (x_{gsp} - A_{ijp} b_{gp} - c_{gp})/\sigma_g^2 \\
dl/d\sigma_{gp}^2 &= 0 = n_p \sigma_{gp}^{-2} - \textstyle\sum_s (x_{gsp} - A_{i,\beta_p(s),p} b_{gp} - c_{gp})^2 \sigma_{gp}^{-4}.
\end{aligned}$$

Here and in what follows, each sum is taken over all the genes in the i^{th} cluster. The above equations simplify to

$$\begin{aligned}
c_{gp} &= \bar{x}_{gp} - n^{-1} b_{gp} \textstyle\sum_j A_{ijp} n_{jp} \\
b_{gp} &= \left[\textstyle\sum_s A_{i,\beta_p(s),p} (x_{gsp} - c_{gp})\right] / \left[\textstyle\sum_j A_{ijp}^2 n_{jp}\right] \\
A_{ijp} &= \left[\sum_{(g,s):\beta_p(s)=j} (x_{gsp} - c_{gp}) b_{gp}/\sigma_{gp}^2\right] / \left[n_{jp} \textstyle\sum_g b_{gp}^2/\sigma_{gp}^2\right] \\
\sigma_{gp}^2 &= n_p^{-1} \textstyle\sum_s (x_{gsp} - A_{i,\beta_p(s),p} b_{gp} - c_{gp})^2.
\end{aligned}$$

Define the sample mean and variance of the expression values of a gene in sample block j:

$$\begin{aligned}
\bar{x}_{gjp} &= n_{jp}^{-1} \sum_{s:\beta_p(s)=j} x_{gsp} \\
s_{gjp}^2 &= n_{jp}^{-1} \sum_{s:\beta_p(s)=j} (x_{gsp} - \bar{x}_{gjp})^2.
\end{aligned}$$

This allows further simplification of the equations

$$\begin{aligned}
c_{gp} &= n_p^{-1} \textstyle\sum_j (\bar{x}_{gjp} - b_{gp} A_{ijp}) n_{jp} \\
b_{gp} &= \left[\textstyle\sum_j A_{ijp} (\bar{x}_{gjp} - c_{gp}) n_{jp}\right] / \left[\textstyle\sum_j A_{ijp}^2 n_{jp}\right] \\
A_{ijp} &= \left[\textstyle\sum_g b_{gp} (\bar{x}_{gjp} - c_{gp})/\sigma_{gp}^2\right] / \left[\textstyle\sum_g b_{gp}^2/\sigma_{gp}^2\right] \\
\sigma_{gp}^2 &= n_p^{-1} \textstyle\sum_j \left[(\bar{x}_{gjp} - A_{i,\beta_p(s),p} b_{gp} - c_{gp})^2 + s_{gjp}^2\right] n_{jp}.
\end{aligned}$$

There is no closed form solution for this system of equations. To obtain the estimates, the formulas are applied iteratively until convergence of the parameters. Each iteration increases the log-likelihood and the limit values satisfy all first order conditions.

CONCLUSION AND FUTURE WORK

In this dissertation we propose several new statistical methods for analysis of biological data sets, each computationally efficient, statistically principal, and validated on real data. The results of this research are presented in four chapters, three of them published as separate papers.

In Chapter 1 we investigate the problem of reconstruction of low rank matrix observed with noise. We begin by demonstrating that, that under minor conditions, an optimal reconstruction method has to be based on the singular value decomposition of the observed matrix, and acts only on its singular values, not affecting its singular vectors. Next, we study the effect of noise on the singular values and singular vectors of low rank matrices by building a connection between the matrix reconstruction problem and spiked population models in random matrix theory. We design a new matrix reconstruction method based on this knowledge, and conduct an extensive simulation study to compare it with existing hard and soft thresholding reconstruction methods. The simulations indicate that the proposed method greatly outperforms even oracle versions of soft and hard thresholding methods.

For future research we plan to assess practical applications of the new matrix reconstruction method in the context biomedical data. We also plan to extend the theoretical research to the matrix completion problem. Matrix completion focuses on recovery of partially observed low rank matrices in noise. It seems likely that the matrix completion problem would also have a solution more efficient that existing methods.

In Chapter 2 we present a new biclustering method, called LAS, that searches for a particular kind of low-rank signal in large matrices. LAS is based on a simple statistical model. We extensively validate it on both real and simulated datasets.

Applying the results from Chapter 1 one can show that LAS can find structures much smaller

than those detectable by singular value decomposition of the data matrix. In the future, we plan to investigate possible applications of LAS to spectral clustering, classification, and vertical integration of biomedical data.

In Chapter 3 we present a tool for fast eQTL analysis. The current version can only handle homozygous SNP data with a moderate number of samples. We are currently working on the next version of the tool, called FastMap 2.0, which is capable of handling heterozygous SNP data and, making use of multiple summation trees, handling datasets with large number of samples.

Finally, in Chapter 4 we present a method, called XPN, for combining gene expression data produced on different platforms. We tested the methods on several pairs of gene expression dataset and compared the method with existing methods. In future we plan to extend the method to combine more that two platforms simultaneously.

BIBLIOGRAPHY

Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C. & Park, J. S. (1999), Fast algorithms for projected clustering, *in* 'SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data', ACM, New York, NY, USA, pp. 61–72.
URL: *http://dx.doi.org/10.1145/304182.304188*

Alter, O., Brown, P. & Botstein, D. (2000), 'Singular value decomposition for genome-wide expression data processing and modeling', *Proceedings of the National Academy of Sciences* **97**(18), 10101.

Asgarian, N. & Greiner, R. (2006), 'Using rank-1 biclusters to classify microarray data', *Department of Computing Science, and the Alberta Ingenuity Center for Machine Learning, University of Alberta, Edmonton, AB, Canada, T6G2E8* .

Baik, J. & Silverstein, J. (2006), 'Eigenvalues of large sample covariance matrices of spiked population models', *Journal of Multivariate Analysis* **97**(6), 1382–1408.

Barron, A. & Yu, J. (1998), 'The minimum description length principle in coding and modeling', *Information Theory, IEEE Transactions on* **44**(6), 2743–2760.

Barry, W., Nobel, A. & Wright, F. (2005), 'Significance analysis of functional categories in gene expression studies: a structured permutation approach', *Bioinformatics* **21**(9), 1943–1949.

Bartlett, M. (1934), 'The vector representation of a sample', *Mathematical Proceedings of the Cambridge Philosophical Society* **30**(03), 327–340.

Beck, J., Lloyd, S., Hafezparast, M., Lennon-Pierce, M., Eppig, J., Festing, M., Fisher, E., Unit, M. & Building, H. (2000), 'Genealogies of mouse inbred strains', *Nature Genetics* **24**, 23–25.

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

Benito, M., Parker, J., Du, Q., Wu, J., Xiang, D., Perou, C. & Marron, J. (2004), 'Adjustment of systematic microarray data biases'.

Bewick, V., Cheek, L. & Ball, J. (2004), 'Statistics review 12: Survival analysis', *Critical Care* **8**(5), 389–394.

Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E. J., Lander, E. S., Wong, W., Johnson, B. E., Golub, T. R., Sugarbaker, D. J. & Meyerson, M. (2001), 'Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses', *Proceedings of the National Academy of Sciences* **98**(24), 13790–13795.
URL: *http://www.pnas.org/cgi/content/abstract/98/24/13790*

Bolstad, B., Irizarry, R., Åstrand, M. & Speed, T. (2003), 'A comparison of normalization methods for high density oligonucleotide array data based on variance and bias', *Bioinformatics* **19**(2), 185–193.

Boser, B., Guyon, I. & Vapnik, V. (1992), 'A training algorithm for optimal margin classifiers', *Proceedings of the fifth annual workshop on Computational learning theory* pp. 144–152.

Broman, K., Wu, H., Sen, S. & Churchill, G. (2003), 'R/qtl: QTL mapping in experimental crosses', *Bioinformatics* **19**(7), 889–890.

Bunea, F., She, Y. & Wegkamp, M. (2010), 'Adaptive Rank Penalized Estimators in Multivariate Regression', *Arxiv preprint arXiv:1004.2995* .

Bystrykh, L., Weersing, E., Dontje, B., Sutton, S., Pletcher, M. T., Wiltshire, T., Su, A. I., Vellenga, E., Wang, J., Manly, K. F., Lu, L., Chesler, E. J., Alberts, R., Jansen, R. C., Williams, R. W., Cooke, M. P. & de Haan, G. d. (2005), 'Uncovering regulatory pathways that affect hematopoietic stem cell function using 'genetical genomics", *Nature Genetics* **37**(3), 225–232.
URL: *http://dx.doi.org/10.1038/ng1497*

Caldas, J. & Kaski, S. (2008), 'Bayesian biclustering with the plaid model', *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)* .

Capitaine, M., Donati-Martin, C. & Féral, D. (2009), 'The largest eigenvalue of finite rank deformation of large Wigner matrices: convergence and non-universality of the fluctuations', *The Annals of Probability* **37**(1), 1–47.

Carlborg, O., De Koning, D., Manly, K., Chesler, E., Williams, R. & Haley, C. (2005), 'Methodological aspects of the genetic dissection of gene expression', *Bioinformatics* **21**(10), 2383–2393.

Cervino, A., Li, G., Edwards, S., Zhu, J., Laurie, C., Tokiwa, G., Lum, P., Wang, S., Castellini, L., Lusis, A. et al. (2005), 'Integrating QTL and high-density SNP analyses in mice to identify Insig2 as a susceptibility gene for plasma cholesterol levels', *Genomics* **86**(5), 505–517.

Cheng, Y. & Church, G. (2000), 'Biclustering of expression data.', *Proc Int Conf Intell Syst Mol Biol* **8**, 93–103.

Chesler, E., Lu, L., Shou, S., Qu, Y., Gu, J., Wang, J., Hsu, H., Mountz, J., Baldwin, N., Langston, M. et al. (2005), 'Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function', *Nature Genetics* **37**, 233–242.

Choi, J., Yu, U., Kim, S. & Yoo, O. (2003), 'Combining multiple microarray studies and modeling interstudy variation', *Bioinformatics* **19**(90001), 84–90.

Churchill, G., Airey, D., Allayee, H., Angel, J., Attie, A., Beatty, J., Beavis, W., Belknap, J., Bennett, B., Berrettini, W. et al. (2004), 'The Collaborative Cross, a community resource for the genetic analysis of complex traits', *Nature Genetics* **36**(11), 1133–1137.

Churchill, G. & Doerge, R. (1994), 'Empirical Threshold Values for Quantitative Triat Mapping', *Genetics* **138**(3), 963–971.

Cope, L., Garrett-Mayer, E., Gabrielson, E. & Parmigiani, G. (2007), 'The Integrative Correlation Coefficient: a Measure of Cross-study Reproducibility for Gene Expressionea Array Data', *Johns Hopkins University, Dept. of Biostatistics Working Papers* p. 152.

Cortes, C. & Vapnik, V. (1995), 'Support-vector networks', *Machine Learning* **20**(3), 273–297.

Dhillon, I. S. (2001), 'Co-clustering documents and words using bipartite spectral graph partitioning', pp. 269–274.
URL: *http://dx.doi.org/10.1145/502512.502550*

Doerge, R. & Churchill, G. (1996), 'Permutation Tests for Multiple Loci Affecting a Quantitative Character', *Genetics* **142**(1), 285–294.

Dozier, R. & Silverstein, J. (2007), 'On the empirical distribution of eigenvalues of large dimensional information-plus-noise-type matrices', *Journal of Multivariate Analysis* **98**(4), 678–694.

Dudoit, S. & Fridlyand, J. (2002), 'A prediction-based resampling method for estimating the number of clusters in a dataset', *Genome Biology* **3**(7).

Dupuis, J. & Siegmund, D. (1999), 'Statistical Methods for Mapping Quantitative Trait Loci From a Dense Set of Markers', *Genetics* **151**(1), 373–386.

Eisen, M. B., Spellman, P. T., Brown, P. O. & Botstein, D. (1998), 'Cluster analysis and display of genome-wide expression patterns', *Proceedings of the National Academy of Sciences* **95**(25), 14863–14868.
URL: *http://dx.doi.org/10.1073/pnas.95.25.14863*

Fan, C., Oh, D. S., Wessels, L., Weigelt, B., Nuyten, D. S., Nobel, A. B., van't Veer, L. J. & Perou, C. M. (2006), 'Concordance among Gene-Expression-Based Predictors for Breast Cancer', *N Engl J Med* **355**(6), 560–569.
URL: *http://content.nejm.org/cgi/content/abstract/355/6/560*

Féral, D. & Péché, S. (2007), 'The largest eigenvalue of rank one deformation of large Wigner matrices', *Communications in Mathematical Physics* **272**(1), 185–228.

Frazer, K., Ballinger, D., Cox, D., Hinds, D., Stuve, L., Gibbs, R., Belmont, J., Boudreau, A., Hardenbol, P., Leal, S. et al. (2007), 'A second generation human haplotype map of over 3.1 million SNPs', *Nature* **449**(7164), 851–861.

Frazer, K., Eskin, E., Kang, H., Bogue, M., Hinds, D., Beilharz, E., Gupta, R., Montgomery, J., Morenzoni, M., Nilsen, G. et al. (2007), 'A sequence-based variation map of 8.27 million SNPs in inbred mouse strains', *Nature* **448**(7157), 1050–1053.

Friedman, J. & Meulman, J. (2004), 'Clustering objects on subsets of attributes', *Journal of the Royal Statistical Society Series B(Statistical Methodology)* **66**(4), 815–849.

Fu, W. (1998), 'Penalized regressions: the bridge versus the lasso', *Journal of computational and graphical statistics* **7**(3), 397–416.

Garrett-Mayer, E., Parmigiani, G., Zhong, X., Cope, L. & Gabrielson, E. (2004), 'Cross-study Validation and Combined Analysis of Gene Expression Microarray Data', *Johns Hopkins University, Dept. of Biostatistics Working Papers* p. 65.

Garrett-Mayer, E., Parmigiani, G., Zhong, X., Cope, L. & Gabrielson, E. (2007), 'Cross-study validation and combined analysis of gene expression microarray data', *Biostatistics* .

Gatti, D., Maki, A., Chesler, E., Kirova, R., Kosyk, O., Lu, L., Manly, K., Williams, R., Perkins, A., Langston, M. et al. (2007), 'Genome-level analysis of genetic regulation of liver gene expression networks.', *Hepatology* **46**(2), 548–57.

Gatti, D., Shabalin, A., Lam, T., Wright, F., Rusyn, I. & Nobel, A. (2009), 'FastMap: Fast eQTL mapping in homozygous populations', *Bioinformatics* **25**(4), 482.

Geman, S. (1980), 'A limit theorem for the norm of random matrices', *The Annals of Probability* **8**(2), 252–261.

Gentleman, R., Ruschhaupt, M., Huber, W. & Lusa, L. (2006), 'Meta-analysis for microarray experiments', *Under revision* .
URL: *http://rss.acs.unt.edu/Rdoc/library/GeneMeta/doc/GeneMeta.pdf*

Getz, G., Levine, E. & Domany, E. (2000), 'Coupled two-way clustering analysis of gene microarray data', *Proceedings of the National Academy of Sciences* **97**(22), 12079.

Ghosh, D., Barette, T., Rhodes, D. & Chinnaiyan, A. (2003), 'Statistical issues and methods for meta-analysis of microarray data: a case study in prostate cancer', *Functional & Integrative Genomics* **3**(4), 180–188.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. & Lander, E. S. (1999), 'Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring', *Science* **286**(5439), 531–537.
URL: *http://www.sciencemag.org/cgi/content/abstract/286/5439/531*

Grothaus, G. (2005), Biologically-Interpretable Disease Classification Based on Gene Expression Data, Master's thesis, Virginia Polytechnic Institute and State University.

Grunwald, P. (2004), 'A tutorial introduction to the minimum description length principle', *Arxiv preprint math.ST/0406077* .

Gu, J. & Liu, J. (2008), 'Bayesian biclustering of gene expression data', *BMC Genomics* **9**(Suppl 1), S4.

Györfi, L., Vajda, I. & Van Der Meulen, E. (1996), 'Minimum Kolmogorov distance estimates of parameters and parametrized distributions', *Metrika* **43**(1), 237–255.

Haley, C. & Knott, S. (1992), 'A simple regression method for mapping quantitative trait loci in line crosses using flanking markers.', *Heredity* **69**(4), 315–24.

Hartigan, J. (1972), 'Direct clustering of a data matrix', *Journal of the American Statistical Association* **67**(337), 123–129.

Hastie, T., Tibshirani, R., Eisen, M., Alizadeh, A., Levy, R., Staudt, L., Chan, W., Botstein, D. & Brown, P. (2000), 'Gene shavingas a method for identifying distinct sets of genes with similar expression patterns', *Genome Biol* **1**(2), 1–21.

Hayes, D., Monti, S., Parmigiani, G., Gilks, C., Naoki, K., Bhattacharjee, A., Socinski, M., Perou, C. & Meyerson, M. (2006), 'Gene expression profiling reveals reproducible human lung adenocarcinoma subtypes in multiple independent patient cohorts', *Journal of Clinical Oncology* **24**(31), 5079.

Hillebrandt, S., Wasmuth, H., Weiskirchen, R., Hellerbrand, C., Keppeler, H., Werth, A., Schirin-Sokhan, R., Wilkens, G., Geier, A., Lorenzen, J. et al. (2005), 'Complement factor 5 is a quantitative trait gene that modifies liver fibrogenesis in mice and humans.', *Nat Genet* **37**(8), 835–43.

Hofmann, K. & Morris, S. (2006), *The structure of compact groups: a primer for the student, a handbook for the expert*, Walter De Gruyter Inc.

Holter, N., Mitra, M., Maritan, A., Cieplak, M., Banavar, J. & Fedoroff, N. (2000), 'Fundamental patterns underlying gene expression profiles: simplicity from complexity', *Proceedings of the National Academy of Sciences* **97**(15), 8409.

Hu, Z., Fan, C., Oh, D., Marron, J., He, X., Qaqish, B., Livasy, C., Carey, L., Reynolds, E., Dressler, L., Nobel, A., Parker, J., Ewend, M., Sawyer, L., Wu, J., Liu, Y., Nanda, R., Tretiakova, M., Orrico, A., Dreher, D., Palazzo, J., Perreard, L., Nelson, E., Mone, M., Hansen, H., Mullins, M., Quackenbush, J., Ellis, M., Olopade, O., Bernard, P. & Perou, C. (2006), 'The molecular portraits of breast tumors are conserved across microarray platforms', *BMC Genomics* **7**(1), 96.
URL: *http://www.biomedcentral.com/1471-2164/7/96*

Huang, E., Cheng, S. H., Dressman, H., Pittman, J., Tsou, M.-H., Horng, C.-F., Bild, A., Iversen, E. S., Liao, M., Chen, C.-M., West, M., Nevins, J. R. & Huang, A. T. (2003), 'Gene expression predictors of breast cancer outcomes', *The Lancet* **361**(9369), 1590–1596.

Ihmels, J., Friedlander, G., Bergmann, S., Sarig, O., Ziv, Y. & Barkai, N. (2002), 'Revealing modular organization in the yeast transcriptional network', *Nat Genet* **31**(4), 370–7.

Irizarry, R., Hobbs, B., Collin, F., Beazer-Barclay, Y., Antonellis, K., Scherf, U. & Speed, T. (2003), 'Exploration, normalization, and summaries of high density oligonucleotide array probe level data', *Biostatistics* **4**, 249–264.

Irizarry, R. et al. (2003), 'Summaries of Affymetrix GeneChip probe level data', *Nucleic Acids Research* **31**(4), 15–15.

Jiang, D., Tang, C. & Zhang, A. (2004), 'Cluster analysis for gene expression data: a survey', *Knowledge and Data Engineering, IEEE Transactions on* **16**(11), 1370–1386.

Johnson, W. E., Li, C. & Rabinovic, A. (2007), 'Adjusting batch effects in microarray expression data using empirical Bayes methods', *Biostatistics* **8**(1), 118–127.
URL: *http://biostatistics.oxfordjournals.org/cgi/content/abstract/8/1/118*

Johnstone, I. (2001), 'On the distribution of the largest eigenvalue in principal components analysis', *The Annals of Statistics* **29**(2), 295–327.

Kadarmideen, H., von Rohr, P. & Janss, L. (2006), 'From genetical genomics to systems genetics: potential applications in quantitative genomics and animal breeding', *Mammalian Genome* **17**(6), 548–564.

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

Kang, H., Zaitlen, N., Wade, C., Kirby, A., Heckerman, D., Daly, M. & Eskin, E. (2008), 'Efficient Control of Population Structure in Model Organism Association Mapping', *Genetics* **178**(3), 1709.

Kao, C. (2000), 'On the Differences Between Maximum Likelihood and Regression Interval Mapping in the Analysis of Quantitative Trait Loci', *Genetics* **156**(2), 855–865.

Kendziorski, C. M., Chen, M., Yuan, M., Lan, H. & D., A. A. (2006), 'Statistical Methods for Expression Quantitative Trait Loci (eQTL) Mapping', *Biometrics* **62**(1), 19–27.

Kent, W., Sugnet, C., Furey, T., Roskin, K., Pringle, T., Zahler, A. et al. (2002), 'The Human Genome Browser at UCSC', *Genome Research* **12**(6), 996.

Kluger, Y., Basri, R., Chang, J. T. & Gerstein, M. (2003), 'Spectral biclustering of microarray data: coclustering genes and conditions', *Genome Research* **13**(4), 703–716.
URL: *http://dx.doi.org/10.1101/gr.648603*

Kong, A. & Wright, F. (1994), 'Asymptotic theory for gene mapping.', *Proceedings of the National Academy of Sciences of the United States of America* **91**(21), 9705–9709.

Konstantinides, K., Natarajan, B. & Yovanof, G. (1997), 'Noise estimation and filtering using block-based singular value decomposition', *IEEE Transactions on Image Processing* **6**(3), 479–483.

Lander, E. & Botstein, D. (1989), 'Mapping Mendelian Factors Underlying Quantitative Traits Using RFLP Linkage Maps', *Genetics* **121**(1), 185–199.

Lazzeroni, L. & Owen, A. (2002), 'Plaid models for gene expression data', *Statistica Sinica* **12**(1), 61–86.

Lee, S., Zou, F. & Wright, F. A. (2010), 'Convergence and Prediction of Principal Component Scores in High-Dimensional Settings', *The Annals of Statistics* .

Liu, J., Yang, J. & Wang, W. (2004), Biclustering in gene expression data by tendency, *in* 'Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE', Washington, DC, USA, pp. 182–193.

Madeira, S. & Oliveira, A. (2004), 'Biclustering algorithms for biological data analysis: a survey', *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **1**(1), 24–45.

Maıda, M. (2007), 'Large deviations for the largest eigenvalue of rank one deformations of Gaussian ensembles', *The Electronic Journal of Probability* **12**, 1131–1150.

Manly, K., Cudmore, Jr, R. & Meer, J. (2001), 'Map Manager QTX, cross-platform software for genetic mapping', *Mammalian Genome* **12**(12), 930–932.

Marčenko, V. & Pastur, L. (1967), 'Distribution of eigenvalues for some sets of random matrices', *USSR Sbornik: Mathematics* **1**(4), 457–483.

Marron, J. S. & Todd, M. (2004), 'Distance weighted discrimination', *Under revision* .

McClurg, P., Janes, J., Wu, C., Delano, D., Walker, J., Batalov, S., Takahashi, J., Shimomura, K., Kohsaka, A., Bass, J. et al. (2007), 'Genomewide Association Analysis in Diverse Inbred Mice: Power and Population Structure', *Genetics* **176**(1), 675.

McClurg, P., Pletcher, M., Wiltshire, T. & Su, A. (2006), 'Comparative analysis of haplotype association mapping algorithms', *BMC Bioinformatics* **7**(1), 61.
URL: *http://www.biomedcentral.com/1471-2105/7/61*

Mehrabian, M., Allayee, H., Stockton, J., Lum, P., Drake, T., Castellani, L., Suh, M., Armour, C., Edwards, S., Lamb, J. et al. (2005), 'Integrating genotypic and expression data in a segregating mouse population to identify 5-lipoxygenase as a susceptibility gene for obesity and bone traits', *Nature Genetics* **37**, 1224–1233.

Mirsky, L. (1960), 'Symmetric gauge functions and unitarily invariant norms', *The Quarterly Journal of Mathematics* **11**(1), 50.

Nadakuditi, R. & Silverstein, J. (2007), 'Fundamental limit of sample eigenvalue based detection of signals in colored noise using relatively few samples', *Signals, Systems and Computers* pp. 686–690.

Nadler, B. (2008), 'Finite sample approximation results for principal component analysis: A matrix perturbation approach', *The Annals of Statistics* **36**(6), 2791–2817.

Negahban, S. & Wainwright, M. (2009), 'Estimation of (near) low-rank matrices with noise and high-dimensional scaling', *Arxiv preprint arXiv:0912.5100* .

Parmigiani, G., Garrett, E., Anbazhagan, R. & Gabrielson, E. (2002), 'A statistical framework for expression-based molecular classification in cancer', *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **64**(4), 717–736.

Parmigiani, G., Garrett-Mayer, E., Anbazhagan, R. & Gabrielson, E. (2004), 'A cross-study comparison of gene expression studies for the molecular classification of lung cancer', *Clin Cancer Res* **10**(9), 2922–2927.

Parsons, L., Haque, E. & Liu, H. (2004), 'Subspace clustering for high dimensional data: a review', *ACM SIGKDD Explorations Newsletter* **6**(1), 90–105.

Paul, D. (2007), 'Asymptotics of sample eigenstructure for a large dimensional spiked covariance model', *Statistica Sinica* **17**(4), 1617.

Péché, S. (2006), 'The largest eigenvalue of small rank perturbations of Hermitian random matrices', *Probability Theory and Related Fields* **134**(1), 127–173.

Peirce, J., Li, H., Wang, J., Manly, K., Hitzemann, R., Belknap, J., Rosen, G., Goodwin, S., Sutter, T., Williams, R. et al. (2006), 'How replicable are mRNA expression QTL?', *Mammalian Genome* **17**(6), 643–656.

Perou, C. M., Sørlie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S., Rees, C. A., Pollack, J. R., Ross, D. T., Johnsen, H., Akslen, L. A., Fluge, Ø., Pergamenschikov, A., Williams, C., Zhu, S. X., Lønning, P. E., Børresen-Dale, A.-L., Brown, P. O. & Botstein, D. (2000), 'Molecular portraits of human breast tumours', *Nature* **406**, 747–752.

Pletcher, M., McClurg, P., Batalov, S., Su, A., Barnes, S., Lagler, E., Korstanje, R., Wang, X., Nusskern, D., Bogue, M. et al. (2004), 'Use of a dense single nucleotide polymorphism map for in silico mapping in the mouse', *PLoS Biol* **2**(12), e393.

Pontius, J., Mullikin, J., Smith, D. et al. (2007), 'Initial sequence and comparative analysis of the cat genome', *Genome Research* **17**(11), 1675.

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

Pritchard, J., Stephens, M. & Donnelly, P. (2000), 'Inference of Population Structure Using Multilocus Genotype Data', *Genetics* **155**(2), 945–959.

Raychaudhuri, S., Stuart, J. & Altman, R. (2000), Principal components analysis to summarize microarray experiments: Application to sporulation time series, *in* 'in Pacific Symposium on Biocomputing', pp. 452–463.

Rhodes, D., Barrette, T., Rubin, M., Ghosh, D. & Chinnaiyan, A. (2002), 'Meta-Analysis of Microarrays Interstudy Validation of Gene Expression Profiles Reveals Pathway Dysregulation in Prostate Cancer 1'.

Rhodes, D., Yu, J., Shanker, K., Deshpande, N., Varambally, R., Ghosh, D., Barrette, T., Pandey, A. & Chinnaiyan, A. (2004), 'Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression', *Proceedings of the National Academy of Sciences* **101**(25), 9309–9314.

Rissanen, J. (n.d.), 'An introduction to the MDL principle', *Available online at www.mdl-research.org* .

Roberts, A., McMillan, L., Wang, W., Parker, J., Rusyn, I. & Threadgill, D. (2007), 'Inferring missing genotypes in large SNP panels using fast nearest-neighbor searches over sliding windows', *Bioinformatics* **23**(13), i401.

Roberts, A., Pardo-Manuel de Villena, F., Wang, W., McMillan, L. & Threadgill, D. (2007), 'The polymorphism architecture of mouse genetic resources elucidated using genome-wide resequencing data: implications for QTL discovery and systems genetics', *Mammalian Genome* **18**(6), 473–481.

Schadt, E., Monks, S., Drake, T., Lusis, A., Che, N., Colinayo, V., Ruff, T., Milligan, S., Lamb, J., Cavet, G. et al. (2003), 'Genetics of gene expression surveyed in maize, mouse and man', *Nature* **422**(6929), 297–302.

Segal, E., Battle, A. & Koller, D. (2003), 'Decomposing gene expression into cellular processes', *Pac Symp Biocomput* pp. 89–100.
URL: *http://view.ncbi.nlm.nih.gov/pubmed/12603020*

Shabalin, A., Tjelmeland, H., Fan, C., Perou, C. & Nobel, A. (2008), 'Merging two gene-expression studies via cross-platform normalization', *Bioinformatics* **24**(9), 1154.

Shabalin, A., Weigman, V., Perou, C. & Nobel, A. (2009), 'Finding large average submatrices in high dimensional data', *The Annals of Applied Statistics* **3**(3), 985–1012.

Shen, R., Ghosh, D. & Chinnaiyan, A. (2004), 'Prognostic meta-signature of breast cancer developed by two-stage mixture modeling of microarray data', *BMC Genomics* **5**(1), 94.

Sorlie, T., Perou, C. M., Tibshirani, R., Aas, T., Geisler, S., Johnsen, H., Hastie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S., Thorsen, T., Quist, H., Matese, J. C., Brown, P. O., Botstein, D., Lonning, P. E. & Borresen-Dale, A.-L. (2001), 'Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications', *Proceedings of the National Academy of Sciences* **98**(19), 10869–10874.
URL: *http://www.pnas.org/cgi/content/abstract/98/19/10869*

Sorlie, T., Tibshirani, R., Parker, J., Hastie, T., Marron, J. S., Nobel, A., Deng, S., Johnsen, H., Pesich, R., Geisler, S., Demeter, J., Perou, C. M., Lonning, P. E., Brown, P. O., Borresen-Dale, A.-L. & Botstein, D. (2003), 'Repeated observation of breast tumor subtypes in independent gene expression data sets', *Proceedings of the National Academy of Sciences* **100**(14), 8418–8423.
URL: *http://www.pnas.org/cgi/content/abstract/100/14/8418*

Stewart, G. (1991), 'Perturbation theory for the singular value decomposition', *SVD and Signal Processing, II: Algorithms, Analysis and Applications* pp. 99–109.

Storey, J. & Tibshirani, R. (2003), 'Statistical significance for genomewide studies', *Proceedings of the National Academy of Sciences* **100**(16), 9440–9445.

Szatkiewicz, J., Beane, G., Ding, Y., Hutchins, L., Pardo-Manuel de Villena, F. & Churchill, G. (2008), 'An imputed genotype resource for the laboratory mouse', *Mammalian Genome* **19**(3), 199–208.

Tagkopoulos, I., Slavov, N. & Kung, S. (2005), 'Multi-class biclustering and classification based on modeling of gene regulatory networks', *Bioinformatics and Bioengineering, 2005. BIBE 2005. Fifth IEEE Symposium on* pp. 89–96.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S. & Golub, T. R. (1999), 'Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation', *Proceedings of the National Academy of Sciences* **96**(6), 2907–2912.
URL: *http://www.pnas.org/cgi/content/abstract/96/6/2907*

Tanay, A., Sharan, R. & Shamir, R. (2002), 'Discovering statistically significant biclusters in gene expression data', *Bioinformatics* **18**(Suppl 1), S136–S144.

TCGA: The Cancer Genome Atlas (n.d.), http://cancergenome.nih.gov/.

Tibshirani, R., Hastie, T., Narasimhan, B. & Chu, G. (2002), 'Diagnosis of multiple cancer types by shrunken centroids of gene expression', *Proceedings of the National Academy of Sciences* **99**(10), 6567.

Tibshirani, R., Walther, G. & Hastie, T. (2001), 'Estimating the number of clusters in a data set via the gap statistic', *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **63**(2), 411–423.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. & Altman, R. (2001*a*), 'Missing value estimation methods for DNA microarrays', *Bioinformatics* **17**(6), 520.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. & Altman, R. (2001*b*), 'Missing value estimation methods for DNA microarrays', *Bioinformatics* **17**(6), 520–525.

Tseng, G. (2007), 'Penalized and weighted K-means for clustering with scattered objects and prior information in high-throughput biological data', *Bioinformatics* **23**(17), 2247.

Tseng, G. & Wong, W. (2005), 'Tight Clustering: A Resampling-Based Approach for Identifying Stable and Tight Patterns in Data', *Biometrics* **61**(1), 10–16.

Turner, H., Bailey, T. & Krzanowski, W. (2005), 'Improved biclustering of microarray data demonstrated through systematic performance tests', *Computational Statistics and Data Analysis* **48**(2), 235–254.

Turner, H., Bailey, T., Krzanowski, W. & Hemingway, C. (2005), 'Biclustering models for structured microarray data', *Computational Biology and Bioinformatics, IEEE/ACM Transactions on* **2**(4), 316–329.

Upper, D. (1974), 'The unsuccessful self-treatment of a case of "Writer's Block"', *Journal of Applied Behavior Analysis* **7**(3), 497.

van 't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A. M., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R. & Friend, S. H. (2002), 'Gene expression profiling predicts clinical outcome of breast cancer', *Nature* **415**(6871), 530–536.

Wachter, K. (1978), 'The strong limits of random matrix spectra for sample matrices of independent elements', *The Annals of Probability* **6**(1), 1–18.

Wall, M., Dyck, P. & Brettin, T. (2001), 'SVDMAN–singular value decomposition analysis of microarray data', *Bioinformatics* **17**(6), 566.

Wang, H., Wang, W., Yang, J. & Yu, P. (2002), 'Clustering by pattern similarity in large data sets', *Proceedings of the 2002 ACM SIGMOD international conference on Management of data* pp. 394–405.

Wang, J., Williams, R. & Manly, K. (2003), 'WebQTL web-based complex trait analysis', *NeuroInformatics* **1**(4), 299–308.

Wang, X., Korstanje, R., Higgins, D. & Paigen, B. (2004), 'Haplotype Analysis in Multiple Crosses to Identify a QTL Gene', *Genome Research* **14**(9), 1767.

Wedin, P. (1972), 'Perturbation bounds in connection with singular value decomposition', *BIT Numerical Mathematics* **12**(1), 99–111.

Weigelt, B., Hu, Z., He, X., Livasy, C., Carey, L., Ewend, M., Glas, A., Perou, C. & van't Veer, L. (2005), 'Molecular portraits and 70-gene prognosis signature are preserved throughout the metastatic process of breast cancer', *Cancer Research* **65**(20), 9155–9158.

Weinstein, J. N., Myers, T. G., O'Connor, P. M., Friend, S. H., Fornace, Albert J., J., Kohn, K. W., Fojo, T., Bates, S. E., Rubinstein, L. V., Anderson, N. L., Buolamwini, J. K., van Osdol, W. W., Monks, A. P., Scudiero, D. A., Sausville, E. A., Zaharevitz, D. W., Bunow, B.,

Viswanadhan, V. N., Johnson, G. S., Wittes, R. E. & Paull, K. D. (1997), 'An information-intensive approach to the molecular pharmacology of cancer', *Science* **275**(5298), 343–349. **URL:** *http://www.sciencemag.org/cgi/content/abstract/275/5298/343*

Wongsawat, Y., Rao, K. & Oraintara, S. (n.d.), 'Multichannel SVD-based image de-noising'.

Yang, H., Bell, T., Churchill, G. & de Villena, F. (2007), 'On the subspecific origin of the laboratory mouse', *Nat. Genet* **39**, 1100–1107.

Yang, Y., Dudoit, S., Luu, P., Lin, D., Peng, V., Ngai, J. & Speed, T. (2002), 'Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation', *Nucleic Acids Research* **30**(4), e15.