Gene Shaving: a New Class of Clustering Methods for Expression Arrays

Trevor Hastie[*], Robert Tibshirani[†], Michael Eisen[‡], Patrick Brown[§], Doug Ross[¶], Uwe Scherf[‖], John Weinstein[**], Ash Alizadeh[††], Louis Staudt[‡‡] and David Botstein

January 14, 2000

Abstract

Through the use of recently developed DNA arrays, it is now possible to obtain accurate, quantitative measurements of the proportional representation of thousands of genes present in a biological sample. DNA arrays have now been used to monitor changes in gene expression during important biological processes (e.g. cellular replication and the response to changes in the environment), and to study variation in gene expression across collections of related samples (e.g. tumor samples from patients with cancer). A major statistical task is to

[*]Depts. of Statistics, and Health, Research & Policy, Sequoia Hall, Stanford Univ., CA 94305. hastie@stat.stanford.edu

[†]Depts. of Health, Research & Policy, and Statistics, Stanford Univ, tibs@stat.stanford.edu

[‡]Life Sciences Division, Lawrence Orlando Berkeley National Labs & Dept. of Molecular. and Cell Biology, University of California. Berk.; eisen@genome.stanford.edu;

[§]Department of Biochemistry, Stanford University;pbrown@cmgm.stanford.edu

[¶]Department of Genetics, Stanford University;dross@cmgm.stanford.edu

[‖]Laboratory of molecular pharmacology, NCI, Bethesda MD 20892

[**]Laboratory of molecular pharmacology, NCI, Bethesda MD 20892; weinstein@dtpax2.NCIFCRF.GOV

[††]Department of Biochemistry, Stanford University; arasha@genome.stanford.edu

[‡‡]Metabolism Branch, DCS, NCI; lstaudt@box-l.nih.gov

Department of Genetics, Stanford University;botstein@genome.stanford.edu

understand the structure of the data from such studies, which often consist of measurements of thousands of genes in tens of conditions. A variety of clustering techniques have been applied to such data, and proven to be useful tools for identifying biologically relevant groupings of genes and samples. In the paper, we describe a new method called gene shaving which tries to identify subsets of genes with coherent expression patterns with large variation across conditions. The technique differs from hierarchical and other clustering methods, described for example in Weinstein et al. (1997) and Eisen et al. (1998). The technique can be unsupervised(that is, treat the genes and samples as unlabeled), or partially or fully supervised by known properties of the genes or samples. We illustrate the method on data from a large survey of gene expression in a diverse collection of human tumor cell lines.

1 Introduction

Virtually all living cells contain chromosomes, large pieces of DNA containing hundreds or thousands of genes, each of which specifies the composition and structure of a single protein. Proteins (polymers of amino acids), are the workhorse molecules of the cell, responsible, for example, for cellular structure, producing energy and important biomolecules (like DNA and proteins), and for reproducing the cells chromosomes. To first approximation, every cell in an organism has the same set of chromosomes, and thus contain the same repertoire of proteins. However, such cells can have remarkably distinct properties (consider the different cells in the human body, virtually all of which contain identical double copies of each of the 23 human chromosomes), which are predominantly the result of differences in the abundance, distribution and state of the cells proteins. One of the seminal discoveries in the early days of molecular biology was that these changes in protein abundance were determined in part by changes the levels of messenger RNA (mRNA), small and relatively unstable nucleic acid polymers that shuttle information from chromosomes to the cellular machines that synthesize new proteins (see Figure 1). Thus, there is a logical connection between the state of a cell and the details of its protein and mRNA composition.

While it remains difficult to measure the abundance of all of a cells proteins, the recently developed DNA array makes it possible to quickly, efficiently and accurately measure the relative representation of each mRNA

species in the total cellular mRNA population. DNA array technologies and their applications have been recently reviewed in great detail (see Nature Genetics supplement "The Chipping Forecast", January 1999), and will not be discussed massive volumes of data produced by such experiments.

For our purposes, a DNA array experiment consists of measurements of the relative representation of a large number of mRNA species (typically thousands or tens of thousands) in a set of related biological samples (e.g. time-points taken during a biological process or clinical samples taken from different patients). For experimental reasons each experimental sample is compared to a common reference sample and the result for each mRNA species (gene) is the ratio of the relative abundance of the gene in the experimental sample compared to the reference. The results of such experiments are naturally represented in a table, with each row representing a gene, each column a sample, and each cells the usually log-transformed expression ratio of the appropriate gene in the appropriate sample.

The process is shown in Figure 2. Tens of thousands of DNA samples, each corresponding to a specific gene, are spotted in an ordered array on a glass slide A target sample and a reference sample are labeled with red and green dyes, and each are hybridized with the DNA on the slide. Using a fluorescent microscope the log (red/green) intensities of RNA hybridizing at each site is measured. The result is a few thousand numbers, typically ranging from say -4 to 4, measuring the expression level of each gene in the experimental sample relative to the reference sample. Positive values indicate higher expression in the target versus the reference, and vice versa for negative values. Henceforth when we refer to expression measurement, we are referring to these log ratios.

The left panel in Figure 3 shows the dataset used in our study. It consists of expression measurements on 6830 genes for a set of 64 human cancer tumors. The right panel shows an exploded view of a subset of the array. The columns are labeled according to the pathology of the sample, and the row labels identify the genes. The order of rows and columns is arbitrary. A full description of these data appears in Ross et al. (1999) and **?**.

Recently some authors have explored the use of clustering methods to arrange the genes in some natural order, with similar genes placed close together. See Weinstein et al. (1997) and Eisen et al. (1998) for recent developments, and Tibshirani et al. (1999) for an overview. There are two major approaches to clustering — bottom up and top-down. In Figure 4 we have applied hierarchical clustering to the genes and cell lines separately.

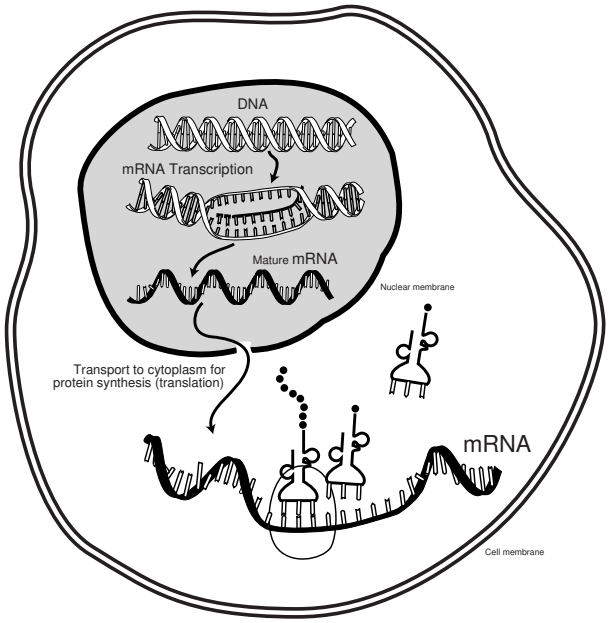

Figure 1: *The process of cell replication*

Each clustering produces a (non-unique) ordering, one that ensures that the branches of the corresponding dendogram do not cross. The figure displays the original data, with rows and columns ordered accordingly.

Some structure is evident in figure 4, and this method can be used to help understand the way in which the genes and cell lines organize. In particular, the column ordering separates the cancer classes extremely well, bearing in mind this is an unsupervised clustering technique. However, finer structure is lost with such a global method. For example suppose the expression of some subset of genes divides the cell lines in an informative (e.g. low for Leukemia and high for others), while another subset of genes divides the cell line a different way. Then methods such as two-way hierarchical clustering, which seek a single reordering of the cell lines for all genes based on a global measure of similarity, cannot find such structure.

In this paper we describe a new method called *gene shaving* for extracting coherent and typically small clusters of genes that vary as much as possible across the cell lines. Figures 5 and 6 show eight gene clusters for the human tumor data, found using shaving.

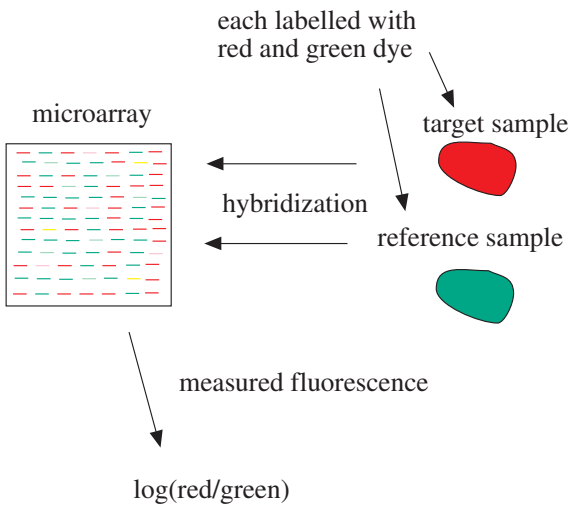

Figure 2: *Schematic of gene expression array process. The nucleotide sequences for a few thousand genes are printed on a glass slide. A target sample and a reference sample are labeled with red and green dyes, and each are hybridized with the DNA on the slide. Through fluorescopy, the log (red/green) intensities of RNA hybridizing at each site is measured.*

In figures 5 and 6 the cell lines have been ordered by values of the average gene expression; this average gene is a good representative of the cluster, since all the members are so similar. Although the cancer classes were not used in the shaving process, the resulting ordering is quite successful at grouping together some of the classes. Since the biological properties of cell lines can differ in many substantially independent axes, we expect that different sets of genes related to different features that vary among genes, will define different orderings of the cell lines. The variance measures at the top of each cluster are discussed in more detail in section 5. The gene names are shown at the left of each rectangle are internal codes for the actual (long) gene names. It turns out that most of the genes are uncharacterized, illustrating the potential for this technique to discover new coherent patterns of expression.

The clusters are all of different sizes. We use an automatic method for choosing the size of the clusters, based on a randomization procedure that protects us from looking too hard in the large sea of genes and finding spurious structure.

The 8 column average genes, one from each cluster, are reasonably uncor-

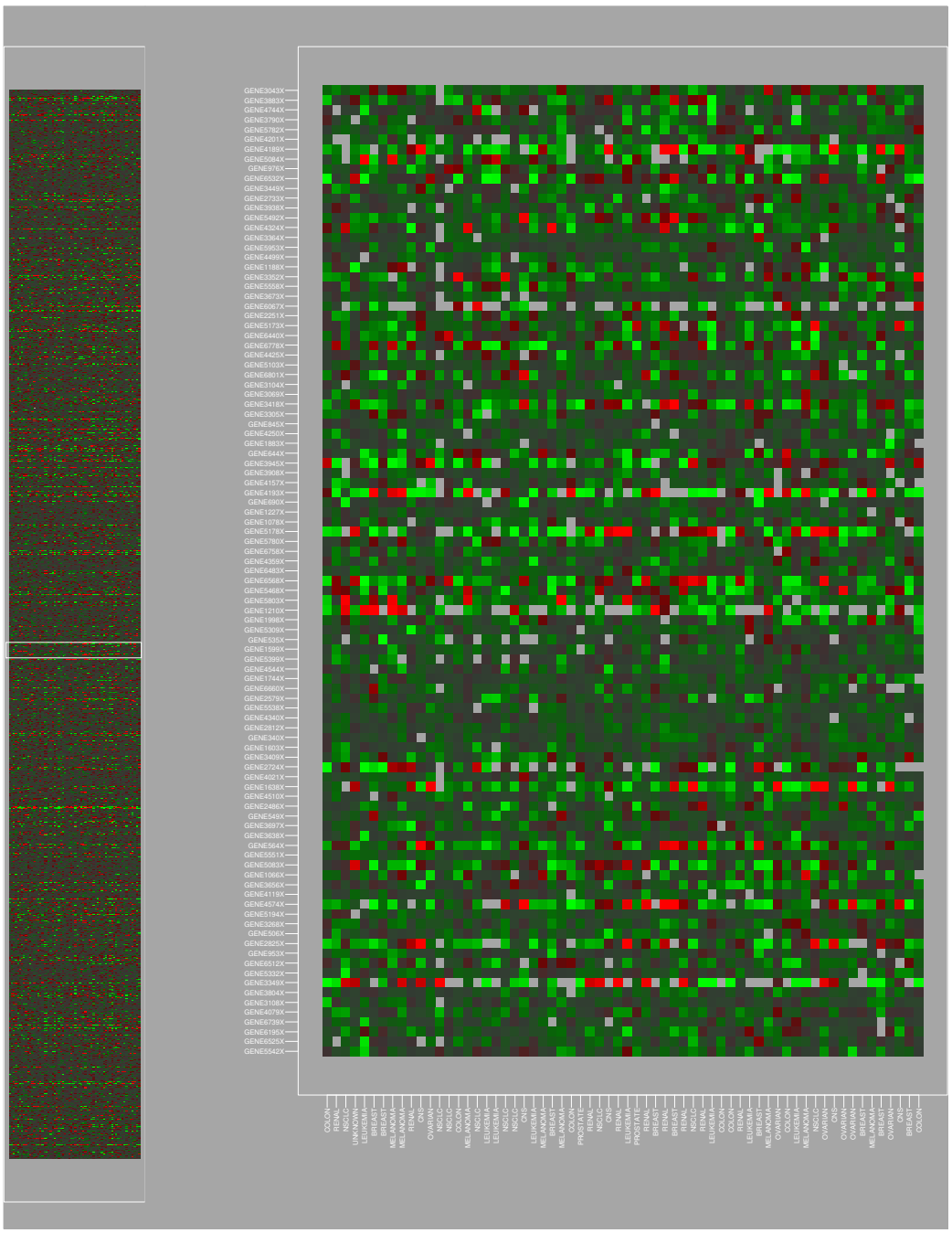

Figure 3: *The left image represents the entire expression array for the human tumor data, in no particular row or column order. The display is a heat map, ranging from bright green (negative, under expressed) to bright red (positive, over expressed). The right panel is an exploded view of a subset of the array.*

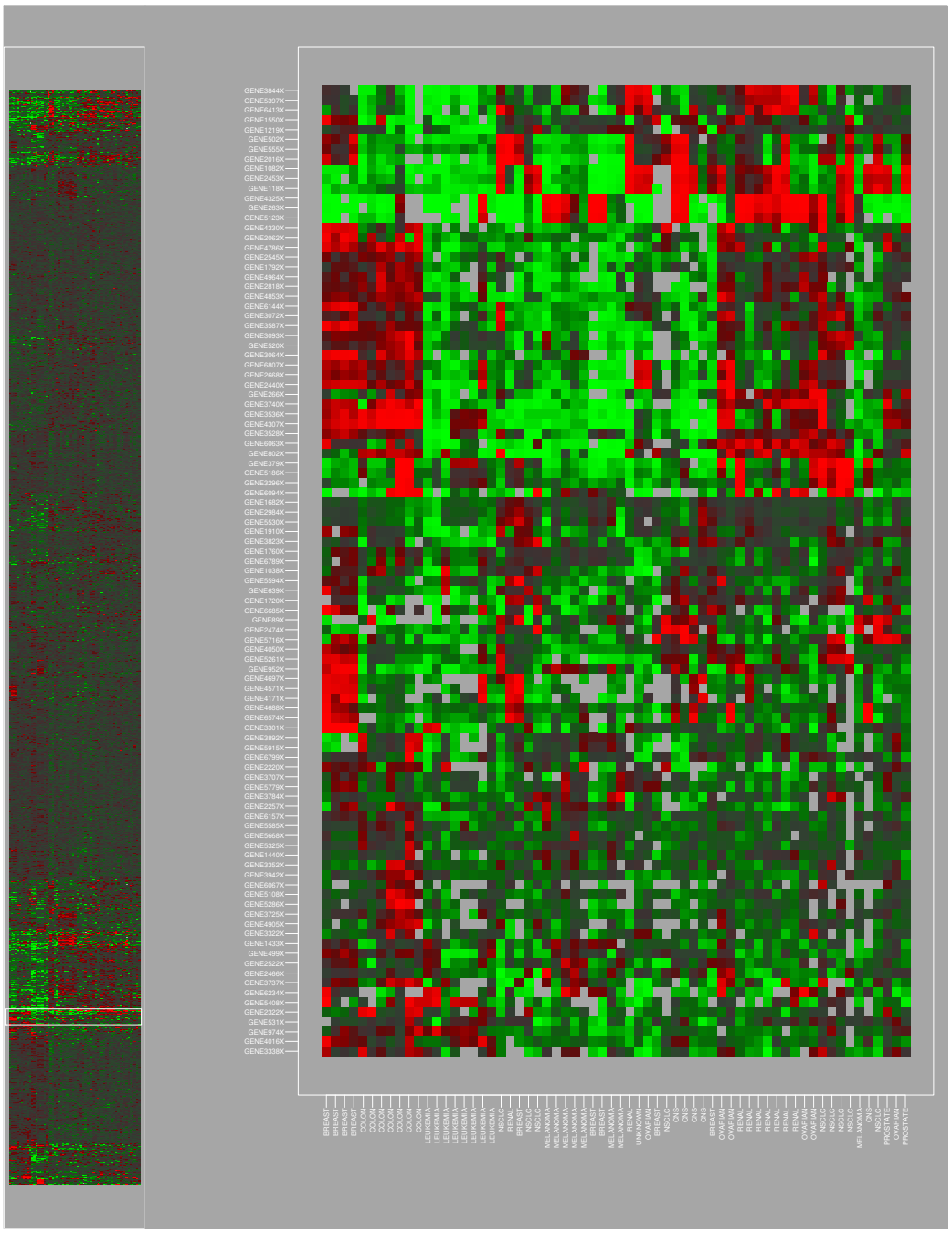

Figure 4: *The human tumor expression array, with rows and columns ordered according to a hierarchical clustering applied separately to the rows and columns. The clustering of the columns has done a good job in separating the cancer classes, albeit in a very high-dimensional space.*

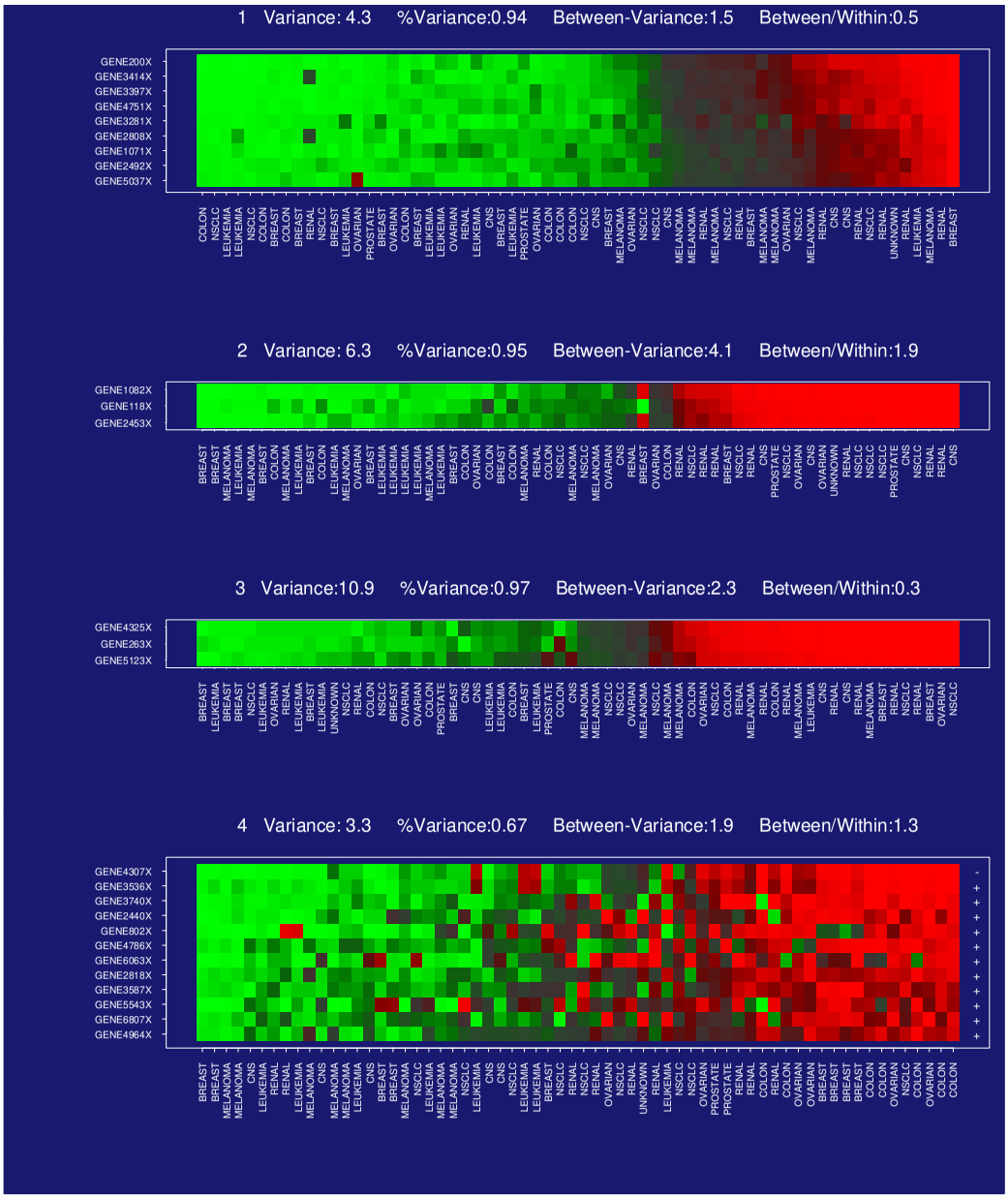

Figure 5: *The first four gene clusters found for the human tumor data. Each is a collection of genes showing similar and strong (high variance) expression behavior. The shaving process determines a sign (+ or −) for each member of the cluster, and the column order is determined by the order of the signed mean gene. The default sign is +, if not indicated.*

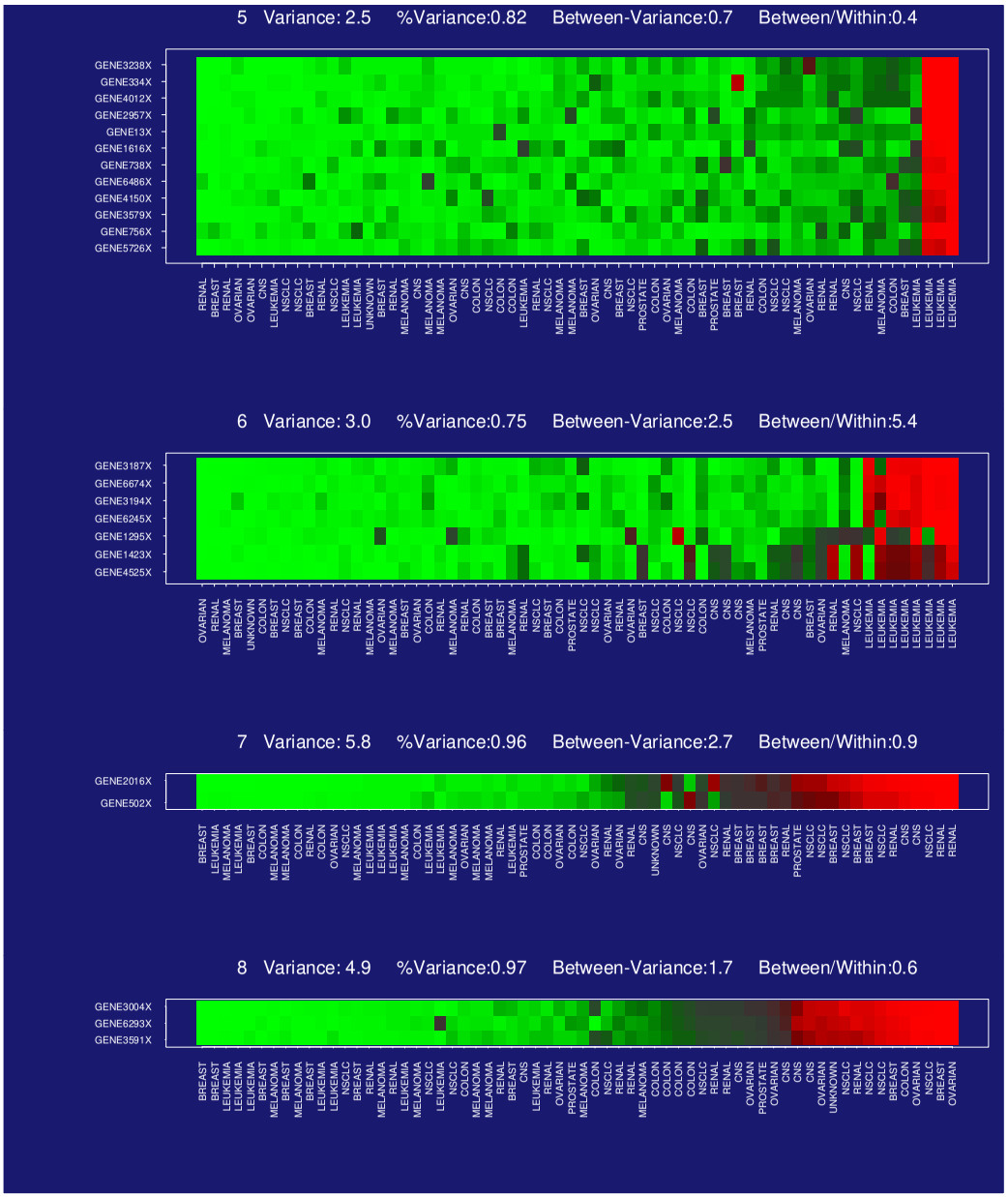

Figure 6: *Gene clusters five to eight; see figure 5. Four measures are reported above each cluster: the variance of the signed mean, the percent variance explained by the signed mean, the between-variance of the signed mean (using cancer classes), and the between/within variance ratio. The latter two use cancer class information that was not used in deriving the clusters.*

related; see figure 9. This is another aspect of the shaving process — it seeks different clusters, where difference is measured by correlation of the cluster mean. Figure 7 shows the results of a hierarchical clustering applied to the eight column average genes. The ordering is as impressive as that obtained for the original array (figure 4). The difference is that there the dimension was 6830, and here it is only 8. This suggests that together the eight clusters with a total of 51 genes contains most of the classification information, relating to tumor of origin, in the data.

This paper is organized as follows. In section 2 we describe principal component gene shaving. Section 3 outlines the gap test for choosing the cluster size, and other methods for isolating clusters are described in section 4. We discuss supervised shaving in section 5. Section 6 shows the application of gene shaving to a problem of stratifying patients having diffuse large cell lymphoma. There we find just two genes that correlate with dramatic differences in patient survival. Finally we propose some further generalizations in section 7.

2 Principal Component Shaving

In this section we describe in detail our technique for producing the clusters in figures 5 and 6.

The gene expression table is an $N \times p$ matrix of real-valued measurements $X = x_{ij}$. The rows are genes, the columns are cell lines, and x_{ij} is the measured (log) expression, relative to a baseline. Typically there are missing entries in X. We use a technique described in Hastie et al. (1999) for imputing missing expression values, an iterative SVD algorithm; our analysis here assumes that X has no missing values.

The sequence of operations 1–5 gives the first cluster of rows — the first block of genes in figure 5. Step 6 orthogonalizes the data to encourage discovery of a different (uncorrelated) second cluster. After orthogonalization and application of Steps 1–5, we obtained the second block of genes in figure 5. The remaining blocks were found in a similar fashion. Note that although we work with the orthogonalized matrix in the shaving process for the second and subsequent clusters, the derived clusters and their averages involve the original genes.

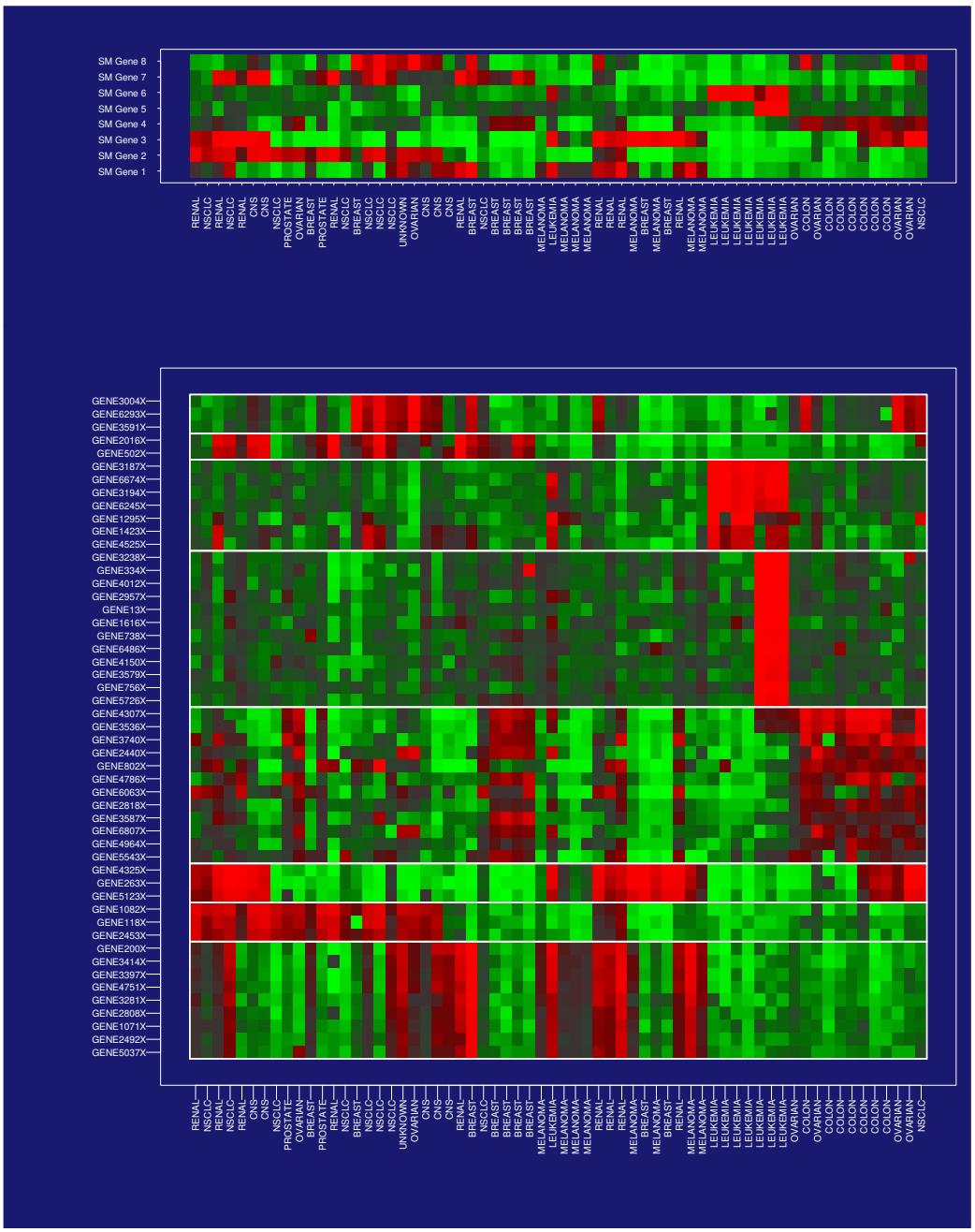

Figure 7: *The top panel shows the eight signed-mean genes together, and ordered by a hierarchical clustering in this eight-dimensional space. The lower panel is similar, except here we show all the genes in each cluster, 51 in all. The cancer class labels of the cell lines shows that this ordering is quite successful at grouping the cancers.*

Principal Component Shaving Algorithm

1. Start with the entire expression table X, each row centered to have zero mean.
2. Compute the leading principal component of the rows of X.
3. Shave off the proportion α (typically 10%) of the rows having smallest inner-product with the leading principal component.
4. Repeat steps 2 and 3 until only one gene remains.
5. This produces a sequence of nested gene clusters $S_N \supset S_{k_1} \supset S_{k_2} \supset \cdots \supset S_1$ where S_k denotes a cluster of k genes. Estimate the optimal cluster size $\hat{k}$ using the gap statistic described in section 3.
6. Orthogonalize each row of X with respect to $\bar{x}$, the average gene in $S_{\hat{k}}$.
7. Repeat steps 1-5 above with the orthogonalized data, to find the second optimal cluster. This process is continued until a maximum of M clusters are found, with M chosen apriori.

2.1 Motivation and Details

In order to motivate and develop this procedure, we need to state a criterion that defines an "optimal" cluster of genes. Intuitively we favor subsets of genes that all behave in a similar manner (coherence), and all show large variance across the cell lines. One measure that captures this notion is the variance of the cluster average.

The average of the k genes in a cluster S_k for cell-line j is $\bar{x}_j = (1/k)\sum_{i\in S_k} x_{ij}$. The variance of $\bar{x}_j$ across cell lines is $(1/p)\sum_{j=1}^{p}(\bar{x}_j - \bar{x})^2$ where $\bar{x} = (1/p)\sum_{j=1}^{p}\bar{x}_j = (1/kp)\sum_{ij} x_{ij}$. This measure combines both variance and covariance of the genes in the subset, since the variance of an average of a set of variables increases with positive correlation between the members of the set.

We can now state more precisely the goal of shaving:

> *Given an expression array, we seek a sequence of nested gene clusters S_k of size k. S_k has the property that the variance of the cluster mean is maximum over all clusters of size k.*

We are still left to decide which set S_k in this sequence to select as our cluster. Later we define a purity measures for a cluster, and a permutation test for providing a cutoff that protects us from finding spurious structure.

How do we find a subset of rows S_k to maximize the between column variance? Since a direct search requires evaluation of $O\binom{N}{k}$ subsets, this is not feasible except for very small k (≤ 4). So we need to resort to approximations. One simple approach is to use a greedy bottom-up strategy. Start with a maximal-variance gene. By exhaustive search, cluster and average it with that gene that causes the least decrease in variance. This can be repeated until the desired cluster size k is reached. Simple updating formulas allow simplifications in the computations. While this strategy works well for small clusters, it tends to be too short-sighted for larger clusters (see figure 10). We need an approach that can directly isolate a fairly large set of genes to average.

We can rephrase the search in terms of a set of gene weights w_i, where $w_i = 1/k$ if $i \in S_k$, else $w_i = 0$. We then seek a weighted average of the genes with maximal variance. It is convenient to introduce a generalized version of this problem that is easy to solve. We allow arbitrary gene weights a_i, positive or negative, with a size constraint $\sum_{i=1}^{N} a_i^2 = 1$. For the generalized problem we seek a linear combination $z_j = \sum_{i=1}^{N} a_i x_{ij}$ such that the variance of z_j is maximal. The solution to this problem is given by the largest principal component of the rows of X, a symmetric eigenvector problem that can be solved exactly and easily. Many of the loadings a_i in the principal component solution tend to have $|a_i| \approx 0$. We set the smallest $100\alpha\%$ of them to zero, reduce the set, and recompute the principal component solution for the reduced subset. This produces a sequence of subsets S_ℓ, decreasing in size from N down to 1, and for each we have a linear combination with maximal variance. We have found that the sequence of subsets produced by this procedure tend to provide good solutions to the more difficult original problem.

The following remarks add support to this shaving procedure:

- Although the normalizations of the two sets of weights are different, a trivial modification fixes that. We can easily change our original weights to $w_i^* = \sqrt{k} w_i$ ($1/\sqrt{k}$ else 0), and hence $\sum w_i^{*2} = 1$. In fact, this is a reasonable modification for eliminating the cluster size bias (σ^2/k) inherent in the variance of averages of variables.

- The principal component weights can be positive or negative. This

leads us to generalize our original criterion and allow weights $w_i = \pm 1/\sqrt{k}$. Genes can belong to a cluster if they are strongly positively or negatively correlated with the members of the cluster. When we average a cluster, we flip the signs accordingly, and refer to the average as a "signed-mean gene". Mostly the signs are the same, but where not we include a sign column in the figures.

- Empirical evidence (figure 8) shows that by the time the clusters have reached their final size k, their principal component weights have $|a_i| \approx 1/\sqrt{k}$.

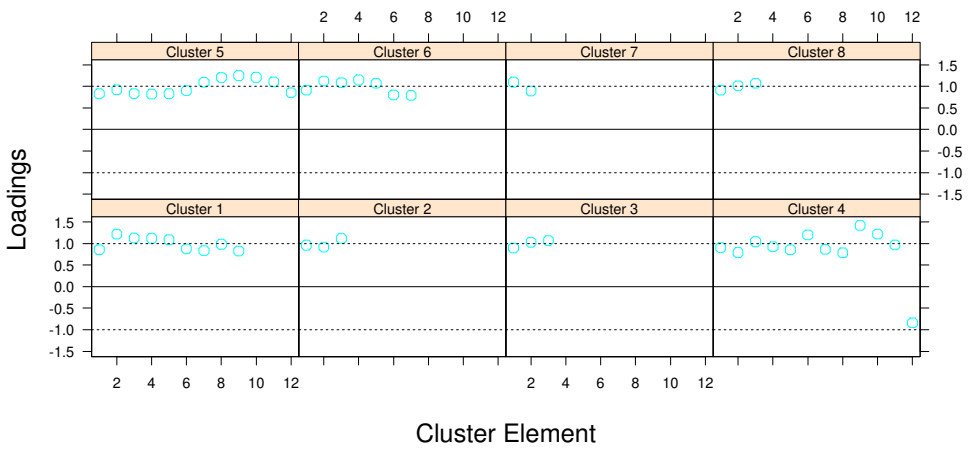

Figure 8: *The final principal component loadings for the eight gene clusters found by shaving. The loadings have been multiplied by $\sqrt{k}$, so that each can be compared to ± 1, which are the horizontal lines in the plots.*

2.2 Shaving and the SVD

The shaving process requires repeated computation of the largest principal component of a large set of variables. If naively implemented, this requires the construction of a $N \times N$ sample covariance matrix Σ of the genes, and

the computation of its largest eigenvector. We can avoid the computational burden by working in the dual space.

Denote the singular value decomposition (SVD) of the $N \times p$ row-centered matrix $\tilde{X}$ by $\tilde{X} = PDQ^T$, where row-centered means that each row averages zero. Here P, D and Q are $N \times p$ orthogonal, $p \times p$ diagonal, and $p \times p$ orthogonal respectively, and we have assumed $N >> p$. D is a diagonal matrix of singular values d_j, ordered from largest to smallest.

If Σ is the $N \times N$ covariance matrix of the genes, then $N\Sigma = \tilde{X}\tilde{X}^T = PD^2P^T$, and we see that the SVD gives the eigenvectors of Σ, and in particular the largest eigenvector $p_1 = \{p_{i1}\}$ is the first column of P. However, $\tilde{X}^T\tilde{X} = QD^2Q^T$ and $\tilde{X}Q = PD$, so we can use the eigen-decomposition of the $p \times p$ matrix $M = \tilde{X}^T\tilde{X}$ to obtain the eigenvectors of Σ. Note that when genes are shaved off, we can update M.

Since we require only the largest eigenvector, we need not compute the entire eigen-decomposition of a $p \times p$ matrix (which can be expensive for a large number of cell lines p); the power method $v_\ell \leftarrow Mv_{\ell-1}/||Mv_{\ell-1}||$ converges to q_1 at rate d_2^2/d_1^2.

The shaving process removes those genes with small loadings p_{i1} on the largest principal component. In addition to appealing to intuition, we can give some justification for this. Suppose we have the largest principal component for a set of N standardized variables x_i, with loadings p_{i1}. How do we identify that variable to delete, such that the variance of the largest principal component on the reduced set is maximum. If we assume to first order that the loadings of the remaining genes will not change, but just be renormalized, it is easy to show that we should eliminate the variable with smallest p_{i1}. Since initially many of these loadings are very small, we shave off more than one, and being small the approximation is reasonably accurate.

Principal components and the SVD simultaneously provide $\min(p-1, N)$ uncorrelated linear combinations of genes, with an ordering in terms of variance. These *eigen-genes* themselves are an interesting set of summaries of an expression array (Alter 1999), although they are global in nature (they do not identify small subsets of interesting genes). Having found the first gene cluster $S_{k_1}^1$, with its mean-gene $\bar{x}_j^1$, we would like to find the next cluster that is different from the first. We could remove the genes in $S_{k_1}^1$ from S_N, and repeat the process, but there is some chance that fairly strong genes shaved off in the first round will vote too strongly again. Instead, we attempt to discourage correlation of subsequent clusters with the mean gene $\bar{x}_j^1$ by adjusting M. Let $H_1^\perp$ be the projection operator in R^p onto the $p-1$ dimen-

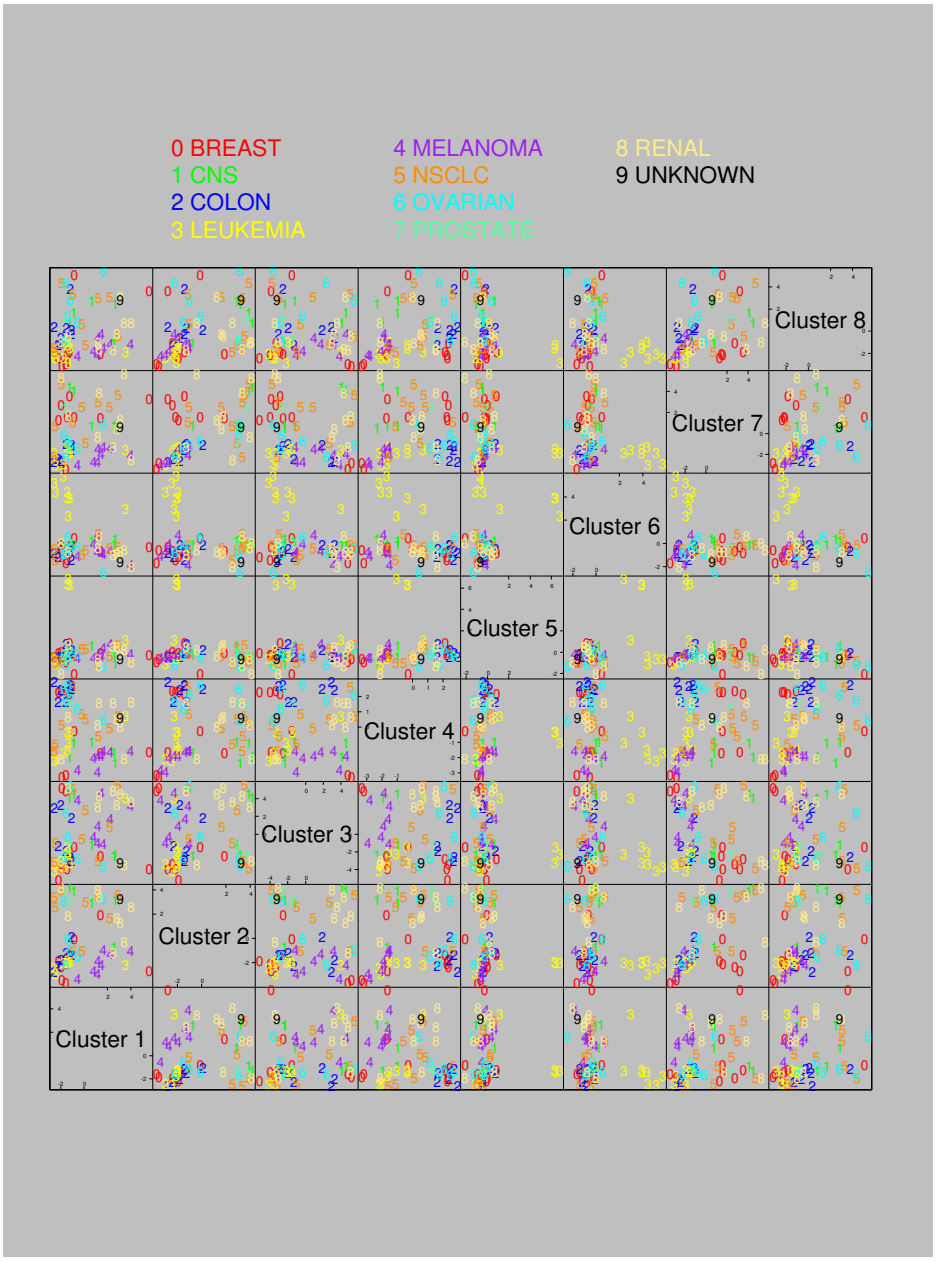

Figure 9: *Pairwise scatterplot of the signed means for all eight clusters. The shaving process avoids strong correlations between clusters by orthogonalization. The cancer classes have been color coded. Note that cluster 6 alone does a good job classifying Leukemia, and clusters 3 and 4 Melanomas.*

sional subspace orthogonal to $\bar{x}_i^1$. We then replace M by $M_{\cdot 1} = H_1^{\perp} M H_1^{\perp}$, and base our next shaving sequence on it. Note that there is a corresponding adjustment for the entire expression array: $\tilde{X}_{\cdot 1} = \tilde{X} H_1^{\perp}$. Having identified the subset $S_{k_2}^2$, we compute $\bar{x}_j^2$ as before, by averaging the (original) genes in the set. This process is repeated; each time we adjust M and $\tilde{X}$ to be orthogonal to *all* the $\bar{x}_j^{\ell},\ \ell = 1, \ldots, k$ found so far.

This procedure works exactly for principal components. For example, if $H_1^{\perp} = I - p_1 p_1^T$, then the leading eigenvector of $M_{\cdot 1}$ can be shown to be p_2, which is the second leading eigenvector of M, subject to being orthogonal to p_1. This process can be repeated; each time we make sure $M_{\cdot 12 \ldots j}$ is orthogonalized with respect to $\bar{x}^k,\ k = 1, \ldots . j$.

3 The gap estimate of cluster size

Each shaving sequence produces a nested set of gene clusters S_k, starting with the entire expression matrix and then proceeding down to some minimum cluster size (typically one). After one of these clusters is chosen, we orthogonalize the rows of the original data matrix with respect to the column mean of this chosen cluster, and then repeat the entire shaving procedure. If there are p columns in the data matrix, we can obtain p shaving sequences in this manner.

If we applied this procedure to null data, in which the rows and columns were independent of each other, we could still find some interesting looking patterns in the resulting blocks. Hence we need to calibrate this process so that we can differentiate real patterns from spurious ones. Even in the case of real structure, it is unlikely that a distinct set of genes is correct for a cluster, and the rest not. More likely there is a graduation of membership eligibility, and we have to decided where to draw the line. Here we describe a procedure based on randomization that helps us select a reasonable cluster size.

Our method requires a quality measure for a cluster. We favor both high variances clusters, and high coherence between members of the cluster. Since the generation of the cluster sequence was driven strongly by the former, we focus on the latter in selecting a good cluster. By analogy with ANOVA for

grouped data, we define the following measures of variance for a cluster S_k:

$$V_W = \frac{1}{p}\sum_{j=1}^{p}\left[\frac{1}{k}\sum_{i\in S_k}(x_{ij}-\bar{x}_j)^2\right] \quad \text{Within Variance} \tag{1}$$

$$V_B = \frac{1}{p}\sum_{j=1}^{p}(\bar{x}_j-\bar{x})^2 \quad \text{Between Variance} \tag{2}$$

$$V_T = \frac{1}{kp}\sum_{i\in S_k}\sum_{j=1}^{p}(x_{ij}-\bar{x})^2 \quad \text{Total Variance}$$

$$= V_W + V_B \tag{3}$$

The between variance is the variance of the (signed) mean gene. The within variance measures the variability of each gene about the cluster average, averaged also over cell lines. Since this can be small if the overall variance is small, a more pertinent measure is the *between-to-within* variance ratio V_B/V_W, or alternatively the *percent variance explained*

$$R^2 = 100\frac{V_B}{V_T} = \frac{\frac{V_B}{V_W}}{1+\frac{V_B}{V_W}} \tag{4}$$

A large value of R^2 implies a tight cluster of coherent genes. This is the quality measure we use to select a cluster from the shaving sequence S_k.

Let S_k index the clusters of a given shaving sequence (with k being the number of genes). Let D_k be the R^2 measure for the kth member of sequence. We wish to know whether D_k is larger than we would expect by chance, if the rows and columns of the data were independent.

Let X^{*b} be a permuted data matrix, obtained by permuting the elements within each row of X. We form B such matrices, indexed by $b = 1, 2, \ldots B$. Let D_k^{*b} be the R^2 measure for cluster S_k^{*b}. Denote by $\bar{D}_k^*$ the average of D_k^{*b} over b. The *Gap* function is defined by

$$\text{Gap}(k) = D_k - \bar{D}_k^* \tag{5}$$

We then select as the optimal number of genes that value of k producing the largest gap:

$$\hat{k} = \text{argmax}_k\ \text{Gap}(k) \tag{6}$$

The idea is that at the value $\hat{k}$, the observed variance is the most ahead of expected. Multiple clusters are produced for the randomized data just like

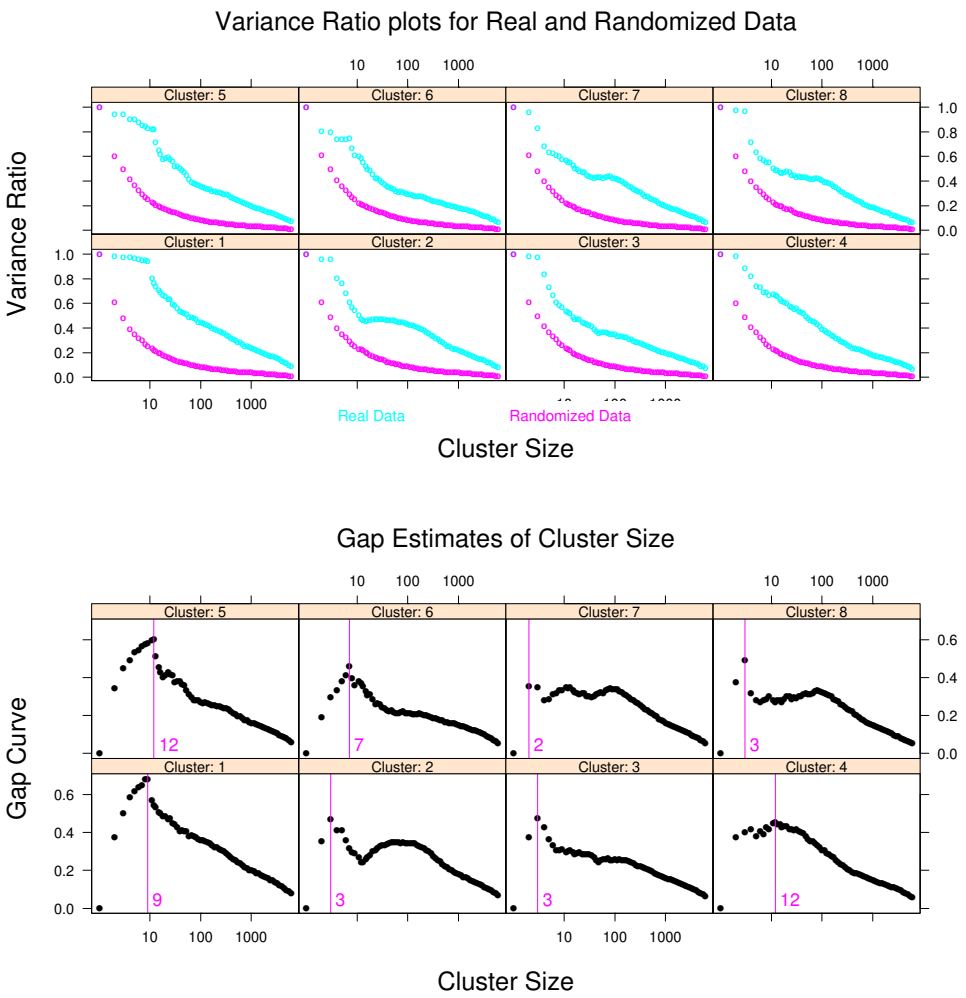

Figure 10: *The top panel displays the percent variance explained by each cluster, both for the original human tumor data, and an average over randomized versions. Notice that for the real data, in many cases the ratio remains near 1 for a while, before dropping off, indicating a coherent cluster. The lower panel shows the gap curve, which highlights the difference between the pair of curves.*

for the original data, and the gap test is used repeatedly to select the cluster size at each stage.

For the human tumor data, the maxima occur between 2 and 12 genes. Figure 10 shows the percent-variance curves D_k for both the original and randomized tumor data as a function of size, and the gap curves used to select the specific cluster sizes in figures 5 and 6.

As a final point, one might consider using the R^2 itself as a criterion for shaving, rather than variance; i.e. look for subsets of genes with large R^2. It is simple to show that principal component shaving applied to the *standardized genes* — each row of X is standardized to have unit variance — tries to achieve this goal. The clusters found were a coherent sets of standardized genes, but did not necessarily have interesting (large) variance on the original scale.

4 Other methods for clustering

Here we compare PC shaving with more conventional clustering methods, such as hierarchical clustering, and also consider other top-down and bottom-up clustering approaches.

4.1 Shaving vs two-way hierarchical clustering

Here we show by a simple example that two-way hierarchical clustering is hampered by its global view in discovering meaningful subsets of genes.

We generated data with $N = 1000$, $p = 60$, $z_{ij} \sim N(0,1)$ and

$$x_{ij} = z_{ij} + b_{ij} \tag{7}$$

where for $1 \leq i \leq 10$

$$b_{ij} = \begin{cases} -1 & \text{if} \quad j \leq 30 \\ +1 & \text{if} \quad j > 30 \end{cases} \tag{8}$$

for $11 \leq i \leq 20$

$$b_{ij} = \begin{cases} -1 & \text{if} \quad j \text{ odd} \\ +1 & \text{if} \quad j \text{ even} \end{cases} \tag{9}$$

and $b_{ij} = 0$ for $i > 20$.

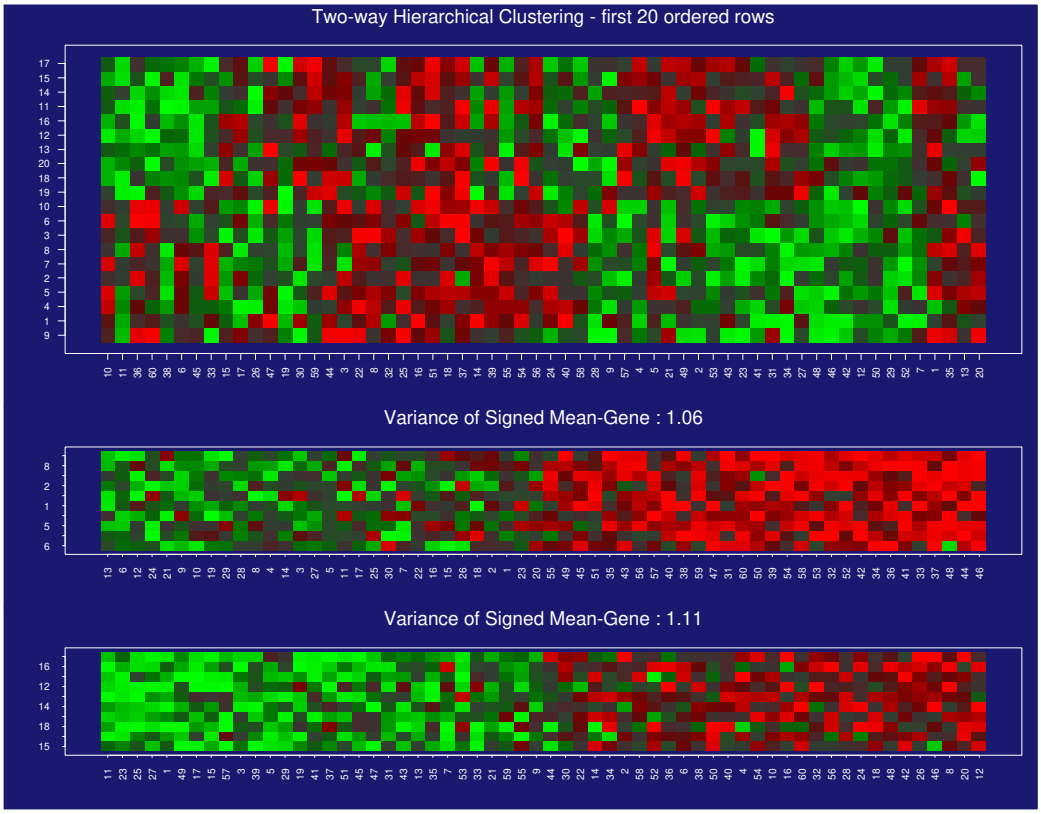

Figure 11: *The top panel shows the results of two-way hierarchical clustering applied to the simulated data. We show the first twenty rows, which were correctly ordered. The column ordering is not of much use. The lower two panels shows the results of shaving, which has recovered the correct genes in each cluster. Each cluster has a column ordering that corresponds to each of the effects* b_1 *and* b_2.

Thus there are two orthogonal trends, each involving 10 rows, superimposed on Gaussian noise. The top panel in figure 11 shows the result of two-way hierarchical clustering applied to these simulated data. The rows 1–20 were clustered first, and hence we have displayed just those rows. However the underlying structure is not apparent in the estimated column ordering. The reason is that a single column ordering is trying to strike a compromise between the two competing column orderings: $1, 2, \ldots 30$ vs $31, 32, \ldots 60$, and $1, 3, 5, \cdots 59$ vs $2, 4, 6, \cdots 60$.

The result from principal component shaving is shown in the lower two panels of figure 11. Each cluster contains the correct genes, and the column order of each cluster is correct as well.

4.2 Other approaches for generating small clusters

Here we collect together and compare several other different approaches one might consider for identifying small influential and coherent clusters. Each method delivers a sequence of clusters S_k, as does principal component shaving. They are

- Bottom-up agglomeration, as described in section 2.1; genes are combined one at a time in a greedy fashion to maximize variance.
- Top-down mean shaving to maximize cluster-mean variance. Starting with the overall mean, we can drop genes one at a time to maximize the variance of the remainder. Simple updating algorithms ease the computational burden. The details are given in the boxed algorithm below.
- Principal component *thresholding*. Rather than repeatedly computing the largest principal component on the decreasing subsets, we can simply compute the first principal component of all of the genes, and then generate the entire sequence by thresholding the gene loadings.

Figure 12 summarizes the differences when searching for a cluster in the human tumor data. We see that while the bottom up approach finds higher variance small clusters, after size five it is dominated by PC shaving. Furthermore, we see from the percent variance plot that there is less coherence than for PC shaving. Thresholding a single principal component seems to be dominated by both.

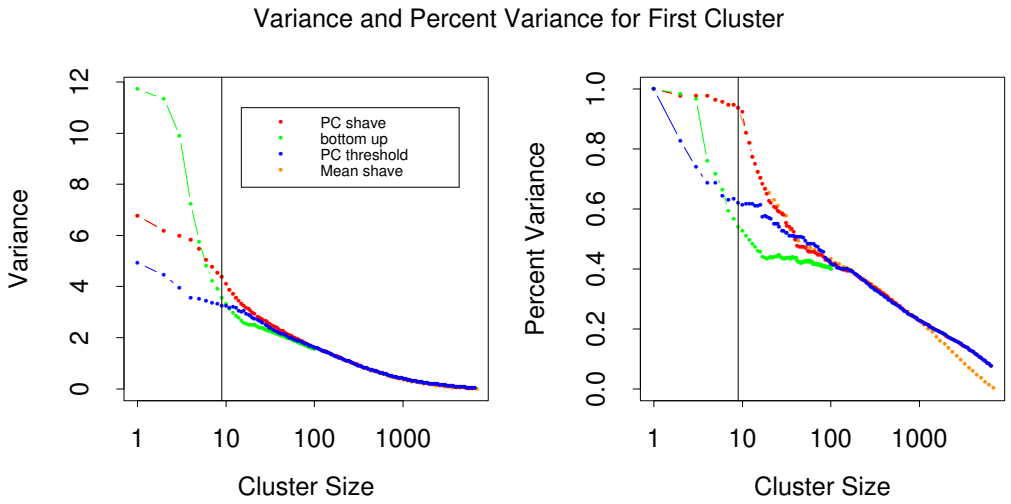

Figure 12: *Variance curves for the first cluster found using four different techniques on the human tumor data. The left panel shows the variance as a function of size for the four methods: principal component shaving, bottom up agglomeration, principal component thresholding, and top-down mean shaving. The right panel shows to percent variance explained by the cluster mean. In this example, PC shaving and mean shaving are almost indistinguishable at small cluster sizes.*

The most promising competitor is the top-down mean shaving. Although efficient updating formulas can be used to determine the best gene to remove with top down-mean shaving, the technique still requires $O(N^2p)$ operations if started at the full set of genes. However, by shaving off a fraction of genes rather than one at a time, one can dramatically increase the speed.

Our experience is that although it is simpler and faster than PC shaving, mean shaving is not as reliable. The left side of Table 1 shows the results of a further 100 simulations from the model (7). The "# correct" refers to the number of times either rows 1–10 or 11-20 are chosen. All methods are successful at finding the second block, if the first block has been isolated. Principal component shaving easily outperforms the other methods in recovering the underlying structure.

Mean Shaving Algorithm

1. Start with the entire expression array X, each row (indexed in S_N) centered to have zero mean. Compute the variance V_i of each gene x_{ij}, the mean gene $\bar{x}_j$ for S_N and its variance V.

2. For the current set S_k compute the *jack-knifed* variances

$$V_{-i} = \frac{k}{k-1}V - \left[\frac{2k}{(k-1)^2}\mathrm{Cov}(\bar{x}_j, x_{ij}) - \frac{V_i}{(k-1)^2}\right]; \qquad (10)$$

   this is an exact formula for the variances of the mean of the set S^i_{k-1} obtained by leaving out the ith gene from S_k.

3. Shave off the proportion α of genes from S_k having smallest values for V_{-i}.

4. Repeat steps 2 and 3 until 1 gene remains.

4.3 An underlying model for gene shaving

By examining models of the kind (7) a bit more closely, we can get some insight into these differences in performance. Suppose

$$X = Z + \sum_{k=1}^{K} w_k b_k^T, \qquad (11)$$

where Z is a $N \times p$ matrix of independent Gaussian elements z_{ij} with mean 0 and variance σ^2. Each w_k is an N-vector of weights, with only a small number n_k of non-zero elements (ones). Denote by C_k the indices of the rows

Table 1: *Results of simulation study*

Method	# correct/100	Average Variance (s.e)
Principal component shaving	100	1.16 (0.01)
Principal component thresholding	64	1.06 (0.02)
Column mean shaving	63	1.10 (0.01)

in the kth block, i.e. those having a one in vector w_k. Each b_k is a p vector, and we further assume that the collection of b_k's are orthogonal: $\langle b_j, b_k \rangle = 0$, $\forall j \neq k$. These models are meant to mimic a reality where there are several different clusters of genes operating in a sea of randomness.

The column mean is $\bar{x} = \sum_i z_i/N + \sum_k (n_k/N) b_k$ (using an obvious vector notation for rows or genes), and by the strong law of large numbers $\bar{x} \approx \sum_k (n_k/N) b_{kj}$. The inner product with each row x_i is $\langle \bar{x}, x_i \rangle \approx ||z_i||^2/N + \sum_k (n_k/N) I(i \in C_k) ||b_k||^2$. If $||b_k||^2/p \approx \sigma^2$, the second term will dominate if the n_k are larger than about 2 or 3. In our example $K = 2$, $p = 60$, $N = 1000$, $n_1 = n_2 = 10$, $\sigma^2 = 1$, and $||b_1||/p = ||b_2||/p = 1$. Mean shaving will fairly easily trim off all those genes not in any C_k. This was true for all 100 experiments. At that point, competition between the "loaded" genes becomes fierce. Subsequent shaves slowly change the mean model resulting in one of the b_k dominating, but because of the noise some of its carriers will be lost.

For the principal component based methods

$$X^T X/N \approx I \cdot \sigma^2 + \sum (n_k/N) b_k b_k^T \tag{12}$$

(here we work with the smaller outer product matrix— see the SVD section 2.2.) This has first K eigenvectors b_k, ordered according to the eigenvalues $\sigma^2 + (n_k/N)||b_k||^2$. If all the $(n_k/N)||b_k||^2$ are exactly the same, the eigenvectors are all equivalent and any vector in their subspace can emerge as the "largest". In practice, with noise present, they will be about the same, but overlap will still exist. This explains why the PC thresholding method can mix genes belonging to different loaded clusters. In all 100 experiments, PC thresholding always delivered such a mix in its chosen 10 genes.

PC shaving on the other hand arrives at the balanced subset $\cup_k C_k$, upsets the balance by trimming off some genes, and then recomputes the principal components. Now one of the b_k is likely to dominate, and all its members will then be recovered in subsequent shaves.

5 Supervised shaving

The methods discussed so far have not used information about the columns to "supervise" the shaving of rows. Here we generalize principal component and mean shaving to incorporate full or partial supervision.

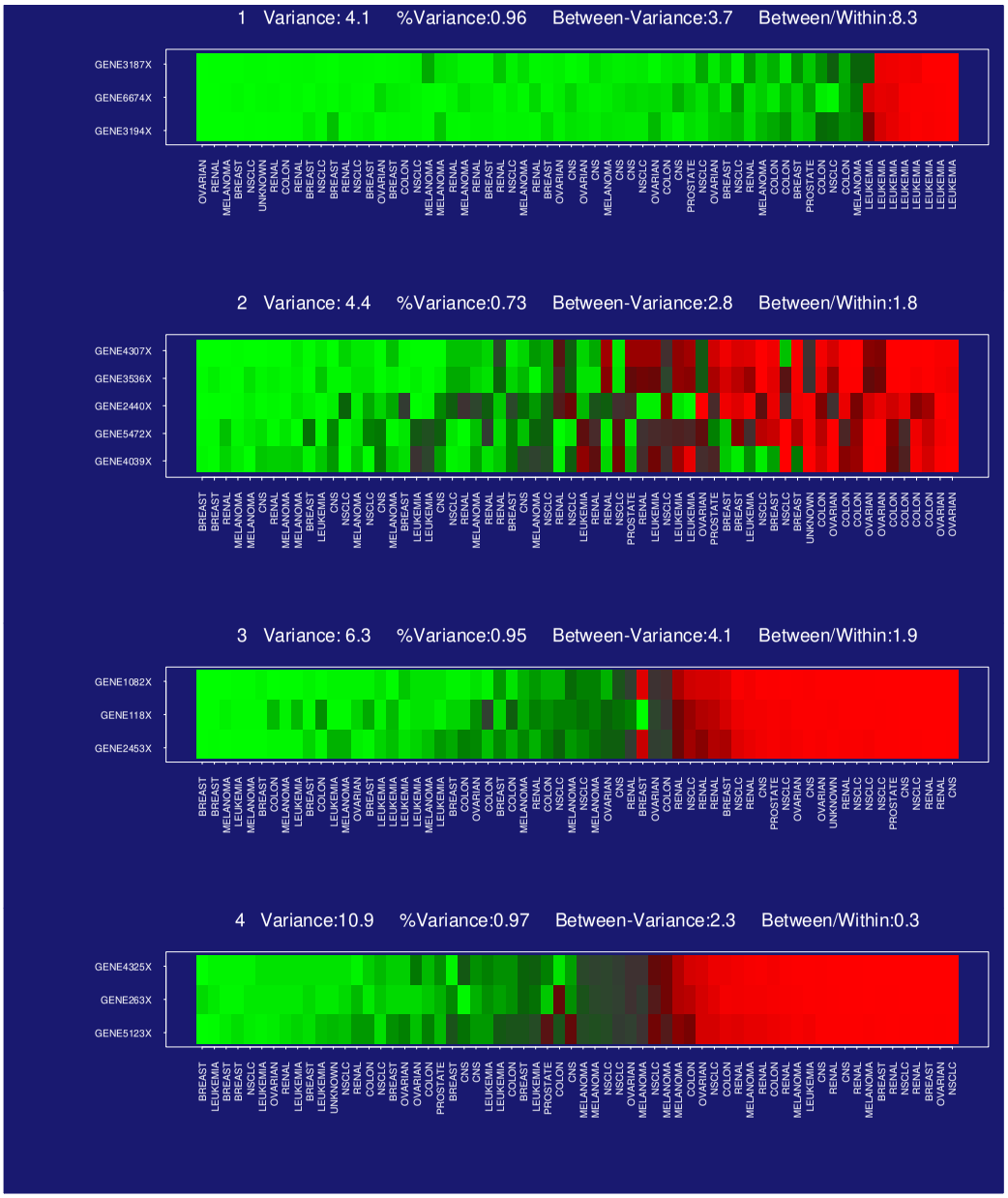

Figure 13: *The first four gene clusters found for the human tumor data, using supervised shaving. As can be expected, the cluster ordering produced by the mean gene produces better class separation than in the unsupervised case.*

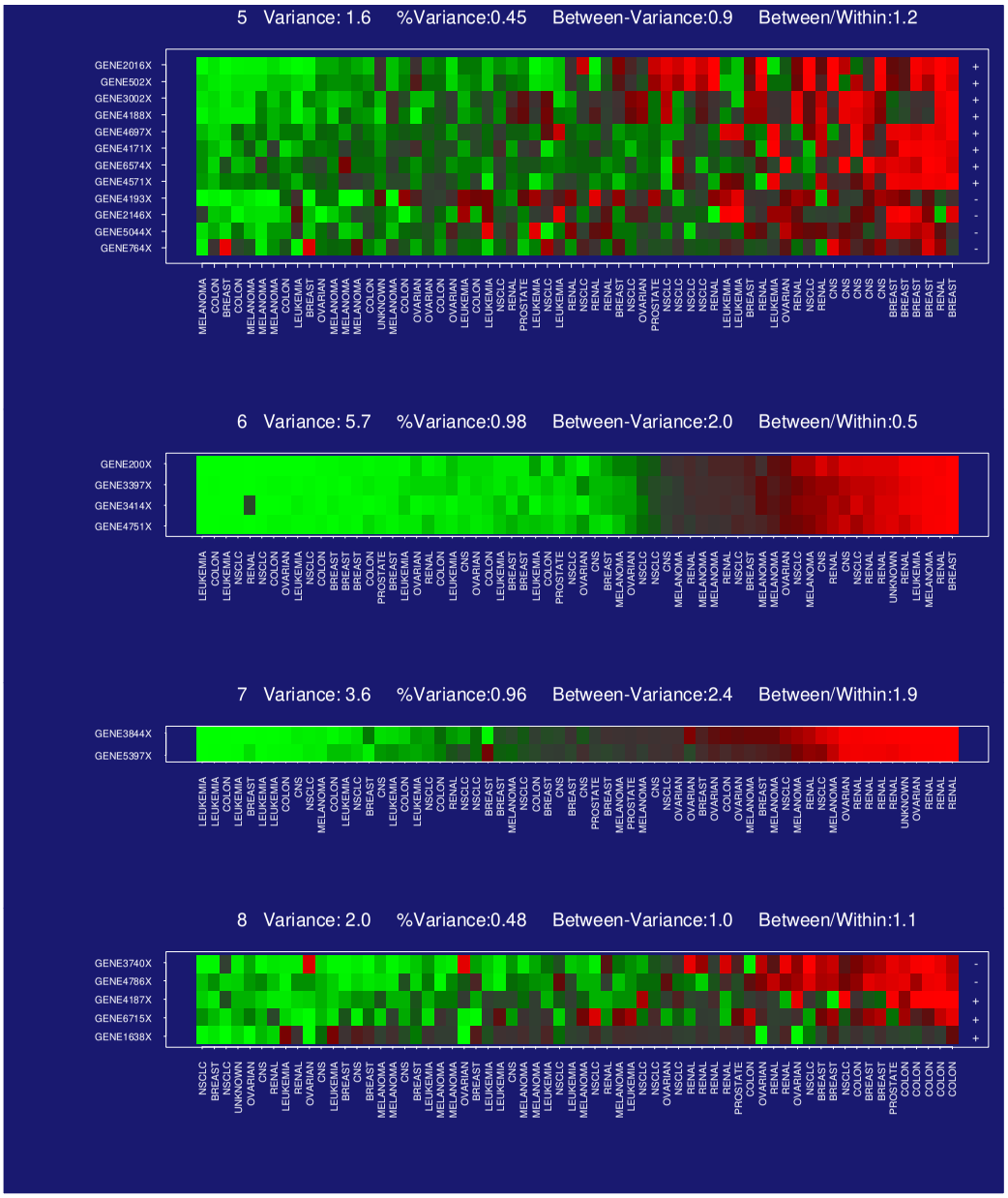

Figure 14: *Gene clusters five to eight found by supervised clustering; see figure 13.*

The human tumor data has a column classification $Y \in \mathcal{C}$, which classifies the tumor according to one of 9 cancer classes (+ one unknown). For a given cluster S_k, we may not only want the cluster mean $\bar{x}_j$ to have large variance, but would also like it to discriminate among the cancer classes. Although the unsupervised methods have automatically tended to produce good discriminators, we can help them do better.

Let $\bar{x}_c = \text{Ave}_{Y_j=c}\bar{x}_j$, $c \in \mathcal{C}$ where $\mathcal{C}$ is the set of classes. $\bar{x}_c$ is the average expression of all genes in the cluster S_k having class c. We use $\text{Var}(\bar{x}_c)$ as a measure of class discrimination. If there are p_c cell lines in class c, then

$$\text{Var}(\bar{x}_c) = \frac{1}{p}\sum_{c\in\mathcal{C}} p_c \bar{x}_c^2, \tag{13}$$

a weighted variance (there is no mean term in (13) since each gene and hence $\bar{x}_j$ average zero, and so the weighted mean does as well). Note that $\text{Var}(\bar{x}_c)$ will be large if the values of $\bar{x}_c$ are spread out, which will occur if $\bar{x}_j$ localizes two or more of the cancer classes.

Figures 13 and 14 show the results of supervised shaving on the human tumor data. These 8 clusters are to be compared with the corresponding 8 found by unsupervised shaving (figures 5 and 6). There are four measures reported at the top of each cluster:

- *Variance:* The variance of the signed cluster mean; in clusters where some genes have a sign-flip, a sign-column is included for identification.
- *% Variance:* The fraction of total variance explained by the signed mean; see (1).
- *Between Class Variance:* as defined above in (13)
- *Between to Within Variance Ratio:* B/W, where $V = B+W$, measures the discriminatory strength of the cluster mean.

The last measure is popular in discriminant analysis, with a high value indicating good separation among some of the classes. Note the distinction between the between/within variances here, defined for the cluster mean gene with respect to the cancer class labels, and the between/within variances discussed in section 3. There each cell line defined a "class", and so the between variance there was in fact the total variance here.

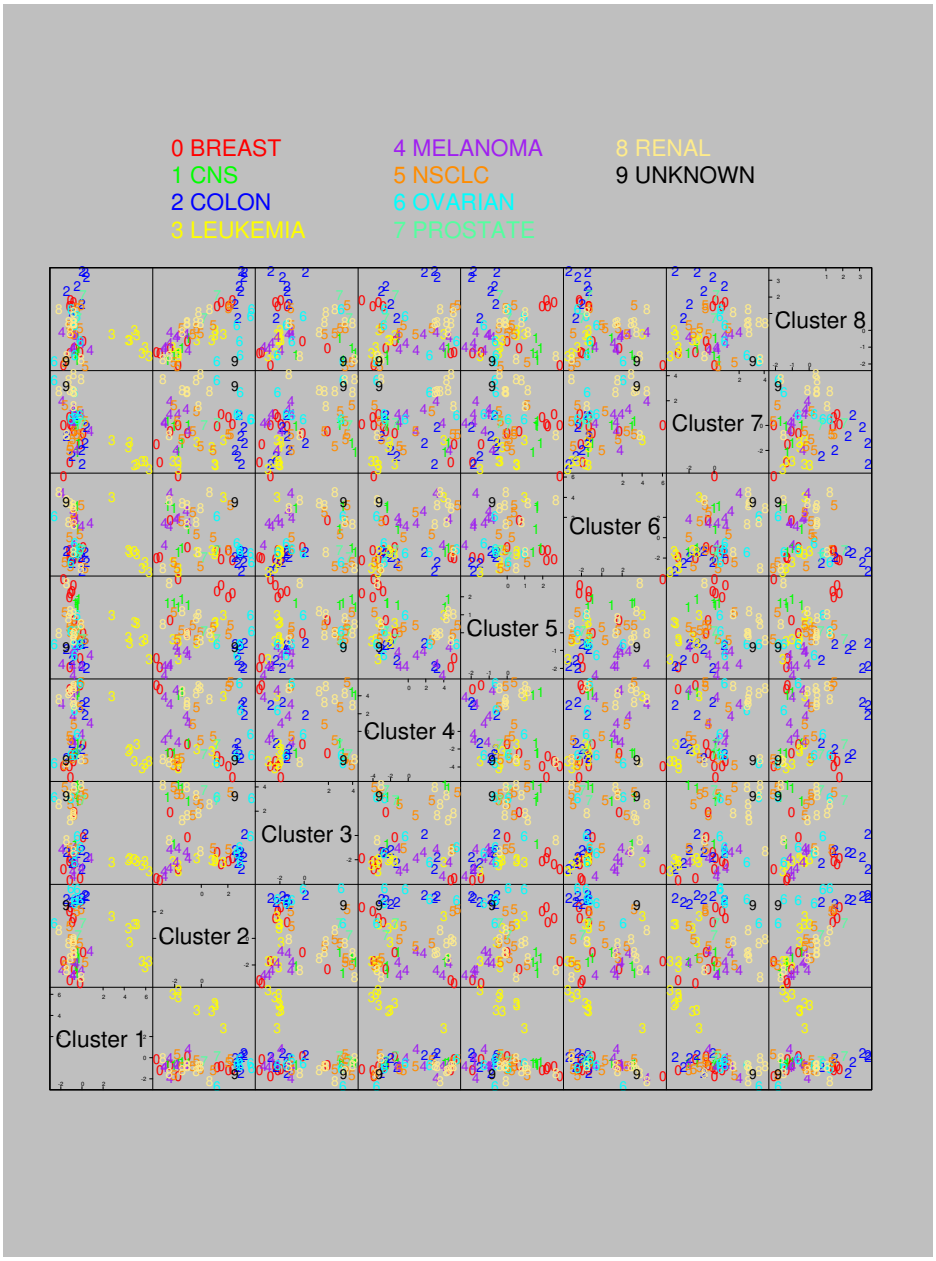

Figure 15: *Pairwise scatterplot of the signed means for all eight clusters found by supervised shaving. The classes are better separated than in the corresponding plot found by unsupervised shaving (figure 9). The error rate of a linear discriminant classifier is 8% using these 8 variables, compared to 30% for unsupervised.*

Overall the clusters here have better B/W ratios, and produce better cancer class separation. Some of the clusters found are identical (3&4 vs 2& 3 earlier), a reflection of the fact that large variance can often be explained by large between variance. Figure 15 gives pairwise plots of the eight mean genes. These show better class separation, and indeed, a simple linear discriminant rule has a training error (9 known classes) of 8%, compared to 30% for the unsupervised case. The classifier also predicts the *unknown* class to be *Ovarian* with posterior probability ≈ 1.

5.1 Computational Details

Let H_B be the regression projection operator such that $H_B\bar{x}$ produces the p vector of fitted class means (i.e. $(H_B\bar{x})_j = \bar{x}_c$ if $Y_j = c$). Then it is easy to see that $\mathrm{Var}(\bar{x}_c) = \bar{x}^T H_B \bar{x}/p$. Now suppose that $\bar{x} = \sum_{i=1}^N w_i x_i$, a weighted sum of genes. Here we have in mind the $1/k$ else zero weights for a cluster mean, or more general weights such as in principal components. Then

$$\begin{aligned} \bar{x}^T H_B \bar{x} &= w^T (X H_B X^T) w \quad (14) \\ &= w^T X^* X^{*T} w \quad (15) \end{aligned}$$

where $X^* = XQ$ and $H_B = QQ^T$ (X^* is a $N \times |\mathcal{C}|$ matrix whose rows are standardized versions of the class means for each gene).

These technicalities reveal a simple modification of PC or mean shaving. The cluster S_k of rows of X with maximal *between-class variance* (of its mean gene) is also the cluster that maximizes the sum of squares of the mean of the rows of X^*. Hence we can apply our same algorithms, such as PC shaving or mean shaving, to X^* to identify such clusters. We mention some technical issues associated with supervised PC shaving.

- *Orthogonalization.* While the shaving algorithm operates on X^*, once the cluster is determined, X is orthogonalized by the cluster mean of the original genes.

- *Gap Test.* We use the identical procedure as in supervised shaving to select the cluster size: a gap test based on the percent variance explained by the cluster mean (of original genes). In other words, supervised shaving delivers a sequence S_k, and we select the size based on cluster purity.

- *Speed.* Since X^* has far fewer columns than X, supervised shaving is faster than unsupervised shaving.

5.2 More general supervised shaving techniques.

There are several ways in which we can generalize the supervised shaving procedure.

One can modify the amount of supervision by using a compromise quadratic form $H_\lambda = (1-\lambda)I + \lambda H_B$ for $\lambda \in [0,1]$ — a procedure we call *partial supervision*. The algorithmic details are unchanged, and this allows more flexibility in striking a balance between supervised and unsupervised clustering. Partial supervision can be particularly useful when (overly) aggressive supervised criteria are used.

In our example Y is a discrete factor, indicating cancer class. In other settings Y might measure some continuous attribute of each cell-line, and a quality measure for a cluster mean $\bar{x}$ might be the strength of its regression on Y. This can be measured using the regression sum-of-squares $\bar{x}^T H_Y \bar{x}$, where $H_Y = y(y^T y)^{-1} y^T$ and $y = (y_1, \ldots, y_p)^T$ is the p vector of realized values of Y. In fact H_B above is exactly of this form; in this case y is a $p \times |\mathcal{C}|$ indicator matrix representing the classes.

This opens the door to a variety of generalizations in a similar vein:

- H_Y can be a projection operator for a more flexible form of regression than linear regression, using a basis expansion in Y. For example, if Y is *time*, then we may require that $\bar{x}$ be smooth in time, with possibly a known periodicity. This can all be achieved using a well chosen Fourier basis expansion of Y, and hence in the construction of H_Y.

- Y can represent several variables, continuous, discrete or both, and H_Y the projection operator that fits a linear regression (main effects and/or interactions). Again $\bar{x}^T H_Y \bar{x}$ measures the association between a cluster mean and this set via the regression sum-of-squares.

All these methods produce an appropriate quadratic form $\bar{x}^T H_Y \bar{x}$ for measuring the strength of $\bar{x}$ relative to Y, by using the regression sum-of-squares with $\bar{x}$ as the response and Y providing the predictors.

We can also reverse their roles, treating y as the response and $\bar{x}$ as the predictor. Suppose we have a log-likelihood $\ell(y; \bar{x}^T \beta)$ that relates y and $\bar{x}$, through a parameter β. Typically $\beta = 0$ indicates no effect for $\bar{x}$ in

the model. The score function is $d\ell/d\beta$, which usually is a sum over the observations of the form $\sum_{j=1}^{p} s_j(\beta)\bar{x}_j$. Letting $s = [s_1(0), s_2(0), \ldots s_p(0)]^T$ be the vector of score components evaluated at $\beta = 0$, this leads to the quadratic form $\bar{x}^T ss^T \bar{x}$. This can be used on its own, or as a penalty in the partially supervised shaving procedure mentioned above.

When fully supervised, this penalty matrix has rank only one, and thus the procedure reduces to simply ranking the genes in order of the size of their inner product with the proportional hazards score vector. Thus there is no role for principal components or top-down shaving in this case. However to encourage coherence within the gene clusters, it is better to use a partially supervised criterion, which does utilize principal components and top down shaving.

This framework can be applied with any linear or generalized model connecting $\bar{x}$ to y. One particularly interesting application is to Cox's proportional hazards model. In that case the score function $d\ell/d\beta$ evaluated at $\beta = 0$ has the form

$$\sum_{k \in R} (\bar{x}_k - z_k) \tag{16}$$

Here R is the set of indices of the individuals who failed, and z_k is the average value of predictors $\bar{x}_j$ for those individuals at risk when individual k failed. The individual score components $s_i(0)$ are obtained by collecting terms in expression (16).

This model-based approach is very general, and allows the user to include $\bar{x}$ as a covariate in a model, along with other prognostic factors that may be available for the samples.

5.3 Aggressive supervision and overfitting

The between-variance is not scale invariant — it responds to both large variance of the cancer-class means, as well as simply large variance. In discriminant analysis the between-to-within variance ratio B/W is a scale invariant measure that more accurately identifies variables with high discriminatory power. Large B/W implies that both the class means of $\bar{x}$ are well separated on average (the between variance is large), and that each class is tightly clustered about its mean (small within variance). This is the last of the four measures that label each cluster in figures 5, 6,13, 14.

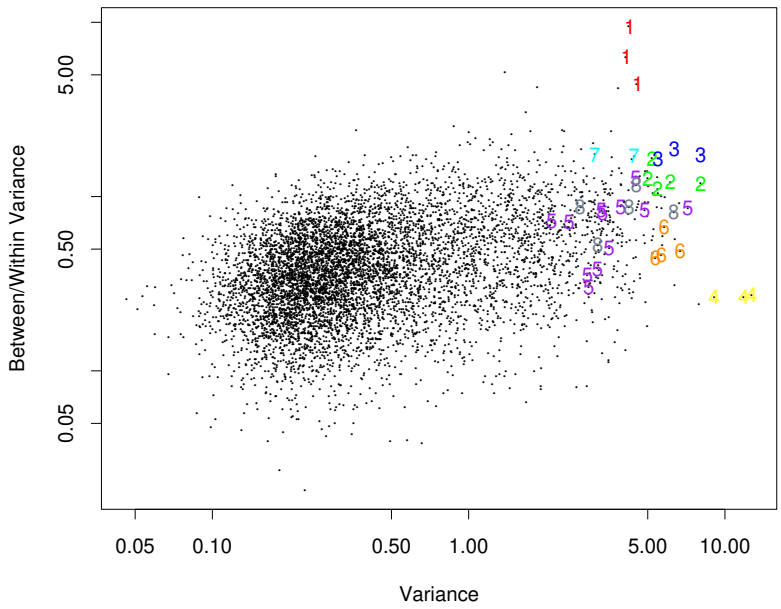

Figure 16: *The B/W ratio for each gene, plot against its variance (both on log scale). The labeled points indicate those genes that belong to one of the supervised clusters 1-8. They all have high variance, and tend to have high B/W ratios.*

Figure 16 displays the B/W ratio for each individual gene, plotted against its variance. The labeled points indicate those genes that are (supervised) cluster members, and they tend to be high on both scores. Recently Golub et al. (1999) used scores approximately equivalent to B/W to screen genes for inclusion in a two class cell-line classifier. They used the average of the top ranking M genes as a classification score (with a cutoff chosen empirically). While our clusters tend to include such genes, they incorporate the additional constraint that the genes in a set are similar and have high variance. These two additional constraints offer some protection from overfitting.

We implemented a mean-shaving algorithm that attempts to directly optimize B/W, a more aggressive supervised strategy than simply using the between variance. Since we can compute exactly one-move adjustments to

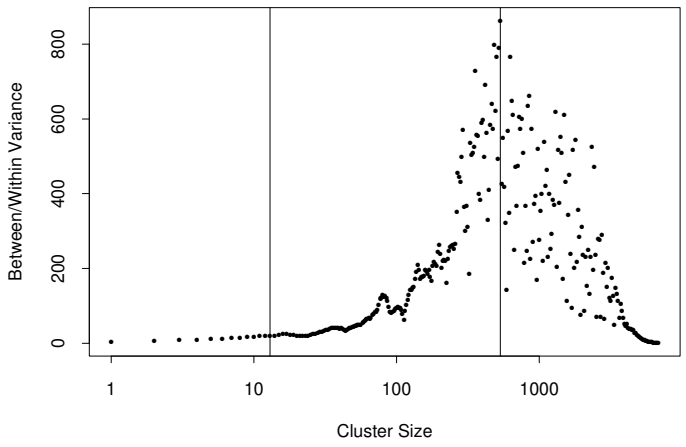

Figure 17: *A plot of the B/W ratios encountered during a column mean shaving to optimize this ratio. The noisy behavior around the peak indicates the sensitivity of the procedure to small changes in the cluster memberships, and is a reflection of the multitude of "good" sets obtainable by greedy search methods. The two vertical lines indicate clusters displayed in figure 18*

both the numerator and denominator, we can compute and rank the one move adjustments for the ratio. Figure 17 shows the sequence of B/W ratios encountered.

There is a lot of variability in this B/W trace, and simply picking the cluster with the largest value does not appear to deliver anything too useful (figure 18, upper plot). The variance and coherence of this set are very poor. It is well known that when the number of predictors (genes) exceeds the number of observations (cell-lines), infinitely many perfect linear separators can be found. Here we are using *equally weighted* linear predictors, which is a strong restriction, but with this many genes it appears that near perfect but most likely spurious separators can be found by greedy search methods. The lower plot in figure 18 shows a much smaller cluster selected from the sequence to have better variance and coherence. The separation suffers, but some classes are nevertheless well separated. These more aggressive supervised clustering techniques deserve more study. We currently favor the

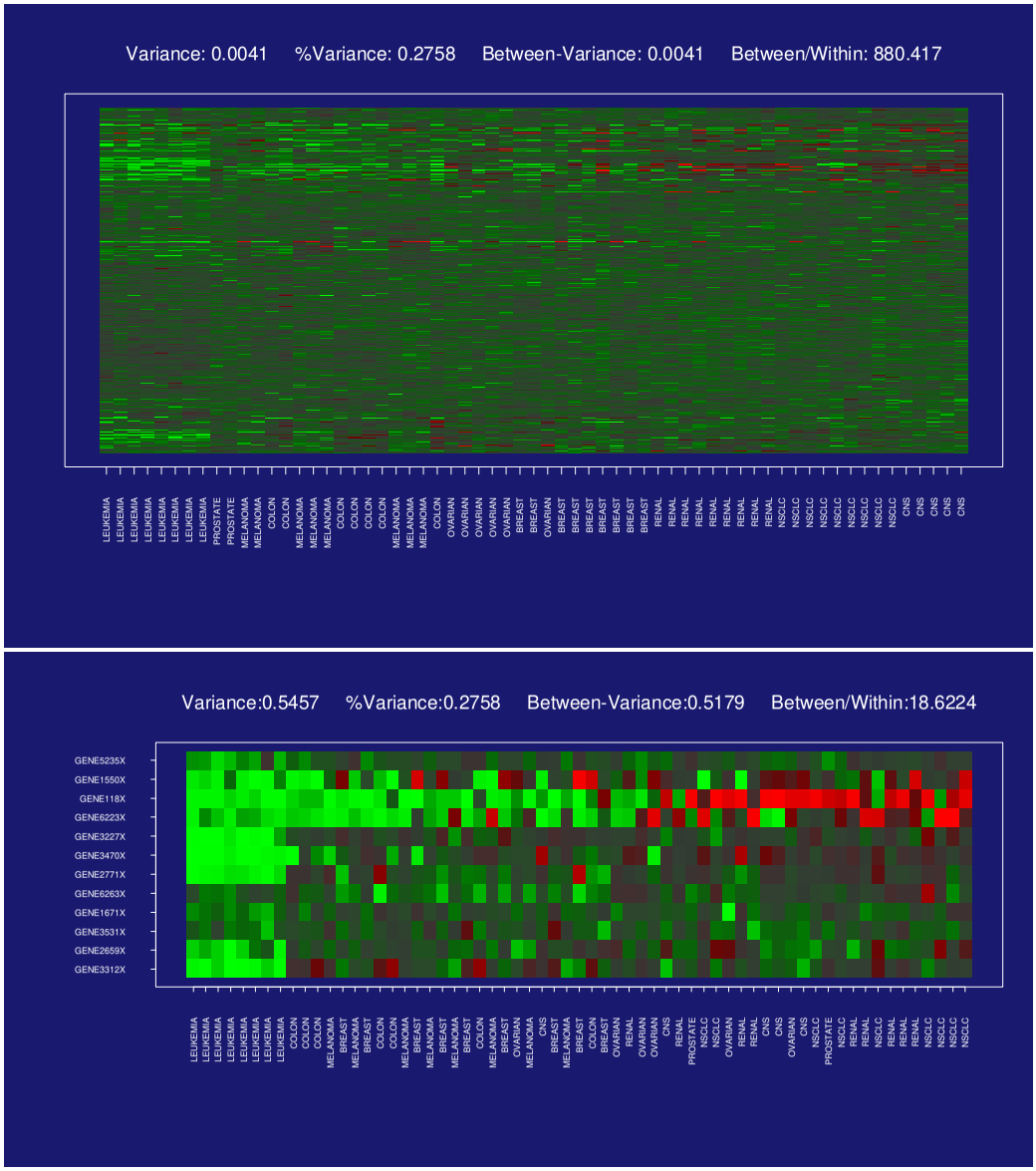

Figure 18: *Two clusters found by the B/W mean shaving algorithm. The upper plot is almost certainly spurious, showing nearly perfect separation using a mean of 500 genes with very little coherence and small variance. The lower plot was selected by eye, for its size, improved variance and coherence. This cluster separates some of the classes well.*

compromise strategy of using between variance for shaving, and report the B/W ratio as a summary measurement.

6 Example: Diffuse large cell lymphoma

We applied unsupervised gene shaving to a dataset consisting of 461 gene expression measurements on 48 patients with Diffuse Large Cell Lymphoma (DLCL). These data are taken from a larger study described in Alizadeh et al. (1999). Four shaving runs were done, creating the four gene blocks shown in figure 19.

Four two-level factors g_1, g_2, g_3, g_4 were created by splitting between the 24th and 25th columns of each gene cluster, thus creating four different low vs high expression groupings. Of the 48 patients, the overall censored survival times as well as the International Prognostic Index (IPI) score were known for 36 of them. We included each factor g_j with IPI (low vs high), in four different proportional hazards models and checked for significance. Of the four models, only g_3 was significant at level 0.05 with IPI in the model. Factor g_3 showed a relative risk of 5.7, P-value=0.0059. (By itself, g_3 showed a relative risk of 2.3, P-value=0.12).

Survival curves are shown in figure 20 and show strong separation between the low and high expression groups of g_3 for the high IPI group. The five genes defining the groups turn out to be two copies of one gene, and three copies of another, the genes being:

```
[1,] "17218|712829|TTG-2=Rhombotin-2=translocatedint(11;14)(p13;q11)
      Tcellacutelymphocyticleukemia=cysteinerichproteinwithLIMmotif"
[2,] "17354|814251|SLAM=signalinglymphocyticactivationmolecule"
```

Patients with low expression of these two genes fare much worse than those with high expression.

To investigate the actual type-I error rate of this finding, we repeated the entire process 100 times on permuted versions of the dataset. This process consisted of gene shaving to produce four blocks, followed by proportional hazards regression with IPI and each of the 4 group variables separately. For each of the 4 group variables we recorded the smallest resulting P-value. Out of 100 datasets, the smallest P-value for g_3, in either IPI group, was

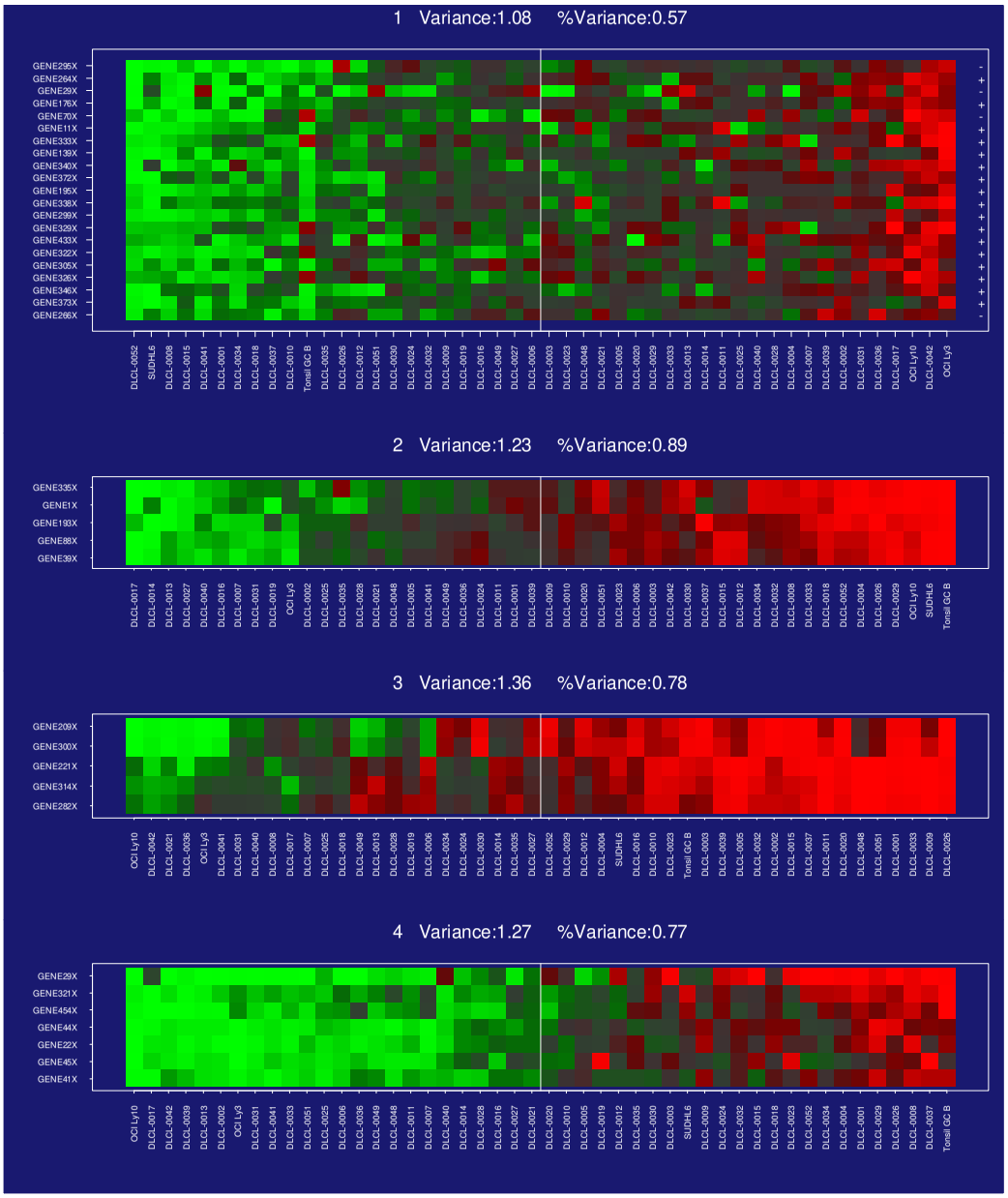

Figure 19: *DLCL Lymphoma example: four gene clusters, showing the gene names and the patient identification for each ordering. The vertical division at the 24-25 boundary was used to create four two-level factors* $g_1, \ldots, g_4$*, which were then related to patient survival.*

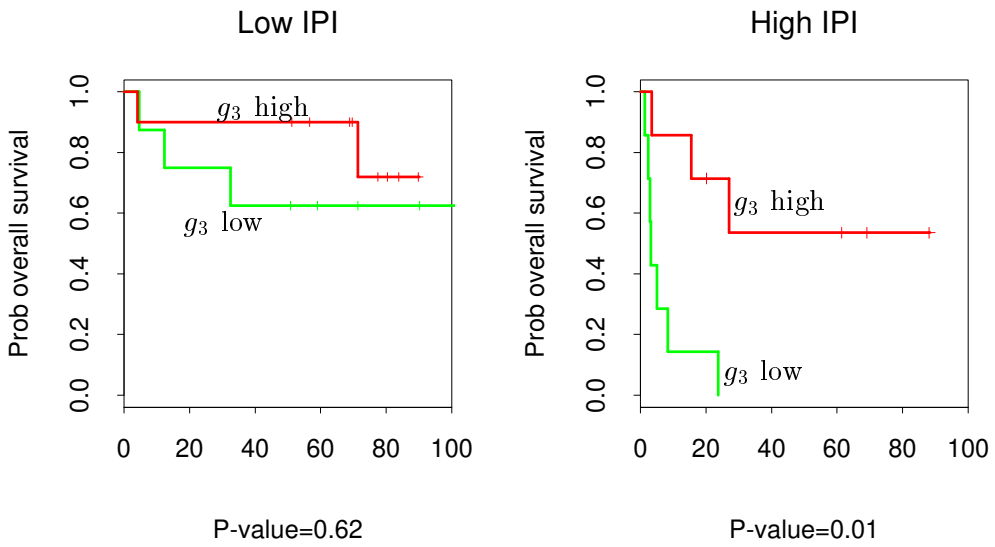

Figure 20: Kaplan-Meier curves for survival, stratified by IPI and g_3

≤ 0.0059 (our value from the original dataset) 3 times. The smallest P-value for g_3 *either* by itself, or stratified by IPI, was $\leq .0059$ five times out of 100. Thus the original test is significant at the 0.05 level. As far as we know, the connection of these genes to DLCL has not been previously suggested, and further investigation is needed to see if this connection is real.

7 Discussion and further generalizations

We have proposed a set of "shaving" methods for isolating interesting sets of genes, from a set of micro-array experiments. The methods may be unsupervised, or may be supervised— i.e. use information available about the samples such as a class label or survival time. Our experiments suggest that one of these methods— principal component shaving— performs the best and is the most promising for real applications.

The proposed shaving methods search for clusters of genes showing both high variation across the samples, and coherence (correlation) across the genes. Both of these aspects are important and cannot be captured by simple clustering of the genes, or thresholding of individual genes based on the variation over samples.

With our model-based approach for supervised shaving, one can incorporate other prognostic factors in the search for interesting gene clusters. If an outcome such as survival time is available for each sample, the method searches for a gene cluster whose column average gene $\bar{x}$ has a significant effect, possibly the presence of other prognostic factors, for predicting the outcome.

The micro-array data x_{ij} we have considered are real-valued expression levels. However other kinds of arrays produce different kinds of data. In particular, some arrays detect the presence of absence of single nucleotide polymorphisms (SNPs), so that the x_{ij} values take on one of $k \geq 2$ unordered values. The shaving methods described can be easily modified to handle this kind of data.

In detail, we construct k data matrices $X_1, X_2 \ldots X_k$, each of size $n \times m$. The ijth element of X_j is 1 if x_{ij} falls in class j, and zero otherwise. Letting Σ_j $j = 1, 2, \ldots k$ be the $n \times n$ covariance matrices of the genes in each X_j, we simply apply principal component shaving, using $\sum_{j=1}^{k} \Sigma_j$ as the variance matrix for the penalty. This can be done unsupervised, or a supervision term can also be added.

Acknowledgments

We would like to thank Mark Segal for helpful discussions, and Nick Fisher and Jerome Friedman for their work on "bump-hunting" which inspired some of the ideas in this paper. The first two authors were supported by grants from The National Science Foundation and the National Institutes of Health.

References

Alizadeh, A., Eisen, M., Davis, R. E., Ma, C., Lossos, I., Rosenwal, A., Boldrick, J., Sabet, H., Tran, T., Yu, X., J., P., Marti, G., Moore, T., Hudsom, J., Lu, L., Lewis, D., Tibshirani, R., Sherlock, G., Chan, W., Greiner, T., Weisenburger, D., Armitage, K., Levy, R. Wilson, W., Greve, M., Byrd, J., Botstein, D., Brown, P. & Staudt, L. (1999), Identification of molecularly and clinically distinct substypes of diffuse large b cell lymphoma by gene expression profiling, Technical report.

Alter, O. (1999), Singular value decomposition for dna microarrays, Technical report. In preparation.

Eisen, M., Spellman, P., Brown, P. & Botstein, D. (1998), 'Cluster analysis and display of genome-wide expression patterns', *Proc. Nat. Acad. Sci* **95**, 14863–14868.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gassenbeek, M., J.P., M., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C. & Lander, E. (1999), 'Molecular classification of cancer: class discovery and class prediction by gene expression monitoring', *Science* **286**, 531–538.

Hastie, T., Alter, O., Sherlock, G., Eisen, M., Tibshirani, R., Botstein, D. & Brown, P. (1999), Imputation of missing values in dna microarrays, Technical report. Working draft.

Ross, D., Scherf, U., Eisen, M., Perou, C., Spellman, P., Iyer1, V., Rees, C., Jeffery, S., Van de Rijn, M., Waltham, M., Alexander, P., Lee, J., Lashkari, D., Shalon, D., Myers, T., Weinstein, J., Botstein, B. & Brown, P. (1999), Systematic variation in gene expression patterns in human cancer cell lines, Technical report, Stanford University.

Tibshirani, R., Hastie, T., Eisen, M., Ross, D., Botstein, D. & Brown, P. (1999), Clustering methods for the analysis of dna microarray data, Technical report, Department of Statistics, Stanford Univ.

Weinstein, J., Myers, T., O'Connor, P., Friend, S., Fornaca Jr, A., Kohn, K., Fojo, T., Bates, S., Rubinstein, L., Anderson, N., Buolamwini, J., van Osdol, W., Monks, A., Scudiero, D., Sausville, E., Zaharevitz, D., Bunow, B., Viswanadhan, V., Johnson, G., Wittes, R. & Paull, K. (1997), 'An information-intensive approach to the molecular pharmacology of cancer', *Science* **275**, 343–9.