Statistica Sinica **12**(2002), 61-86

PLAID MODELS FOR GENE EXPRESSION DATA

Laura Lazzeroni and Art Owen

Stanford University

Abstract: Motivated by genetic expression data, we introduce plaid models. These are a form of two-sided cluster analysis that allows clusters to overlap. Plaid models also incorporate additive two way ANOVA models within the two-sided clusters. Using these models we find interpretable structure in some yeast expression data, as well as in some nutrition data and some foreign exchange data.

Key words and phrases: Microarrays, SVD, transposable data, unsupervised learning.

1. Introduction

This article introduces the plaid model, a tool for exploratory analysis of multivariate data. The motivating application is the search for interpretable biological structure in gene expression microarray data. Eisen, Spellman, Brown and Botstein (1998) is an early and influential paper advocating the use of cluster methods to identify groups of co-regulated genes from microarray data. We present the model and illustrate it on gene expression and other data. The plaid model allows a gene to be in more than one cluster, or in none at all. It also allows a cluster of genes to be defined with respect to only a subset of samples, not necessarily with respect to all of them. Thus, for example, some yeast genes may belong together in a cluster according to the way they are expressed when the yeast is forming spores, while clustering with other genes under other conditions.

Section 2 introduces the plaid model as a sum of terms called layers, using microarray data as motivation. Section 3 describes our approach to fitting this model to data. Section 4 is devoted to the problem of deciding how many layers to include in a model. Sections 5, 6 and 7 present examples using data on food composition, foreign exchange rates, and gene expression in yeast, respectively. Our main interest is in the microarray application, but the other examples give us insight into how the model works. Section 8 compares the plaid model to others in the literature. Section 9 presents our conclusions.

2. Plaid Model

DNA microarrays allow the measurement of expression levels for a large number of genes, perhaps all genes of an organism, within a number of different experimental samples. The samples may correspond to different toxins or time points. In other cases, the samples may have come from different organs, from tumors or healthy tissue, or from different individuals. The data take the form of a large matrix Y_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, p$, where i indexes n genes and j indexes p samples. The value Y_{ij} measures the strength with which gene i is expressed in sample j. The number np of data values can be very large, over $500,000$ with present technology, and continues to increase with time. Simply visualizing such a volume of data is challenging, and extracting biologically relevant knowledge is harder still.

A natural starting point is to form a color image of the data on an n by p grid, with each cell colored according to the value of Y_{ij}. Figure 1 shows one such image described in Section 7. The ordering of the rows and sometimes of the columns in such an image can be arbitrary. It is natural then to consider ways of reordering the rows and columns in order to group together similar rows and similar columns, thus forming an image with blocks of similar color. Bertin (1983) uses the term "reorderable matrix" for data of this type and gives examples of reordering. That text contains a photograph of an old manual device for reordering matrices. The larger Y_{ij} values are represented by dark beads, and the user can lift and permute rows or columns of beads until a nearly block diagonal pattern emerges. The rows in Figure 1 were ordered after running a hierarchical clustering on the genes.

An ideal reordering of the array would produce an image with some number K of rectangular blocks on the diagonal. Each block would be nearly uniformly colored, and the part of the image outside of these diagonal blocks would be of a neutral background color. This ideal corresponds to the existence of K mutually exclusive and exhaustive clusters of genes, and a corresponding K-way partition of the samples. Every gene in gene-block k is expressed within, and only within, those samples in sample-block k. Algebraically, this ideal corresponds to the representation

$$Y_{ij} \doteq \mu_0 + \sum_{k=1}^{K} \mu_k \rho_{ik} \kappa_{jk}, \tag{1}$$

where μ_0 is a background color, μ_k describes the color in block k, ρ_{ik} is 1 if gene i is in the k'th gene-block (zero otherwise), and κ_{jk} is 1 if sample j is in the k'th sample-block (zero otherwise). The conditions that every gene and every sample be in exactly one cluster are then $\sum_k \rho_{ik} = 1$ for all i, and $\sum_k \kappa_{jk} = 1$ for all j, respectively.

It has long been recognized (see Needham (1965)) that such an ideal reordering will seldom exist in real data. It is more likely that the blocks will overlap in some places. That is, we may need to allow $\sum_k \rho_{ik} \geq 2$ for some i, or $\sum_k \kappa_{jk} \geq 2$

for some j. Similarly there may be some genes or samples that do not fit well into any cluster. In clustering there is often a miscellaneous (or "ragbag") cluster for items that do not belong to any well defined cluster. This corresponds to $\sum_k \rho_{ik} = 0$ for some i, or $\sum_k \kappa_{jk} = 0$ for some j, assuming that the ragbag cluster is close to the background level.

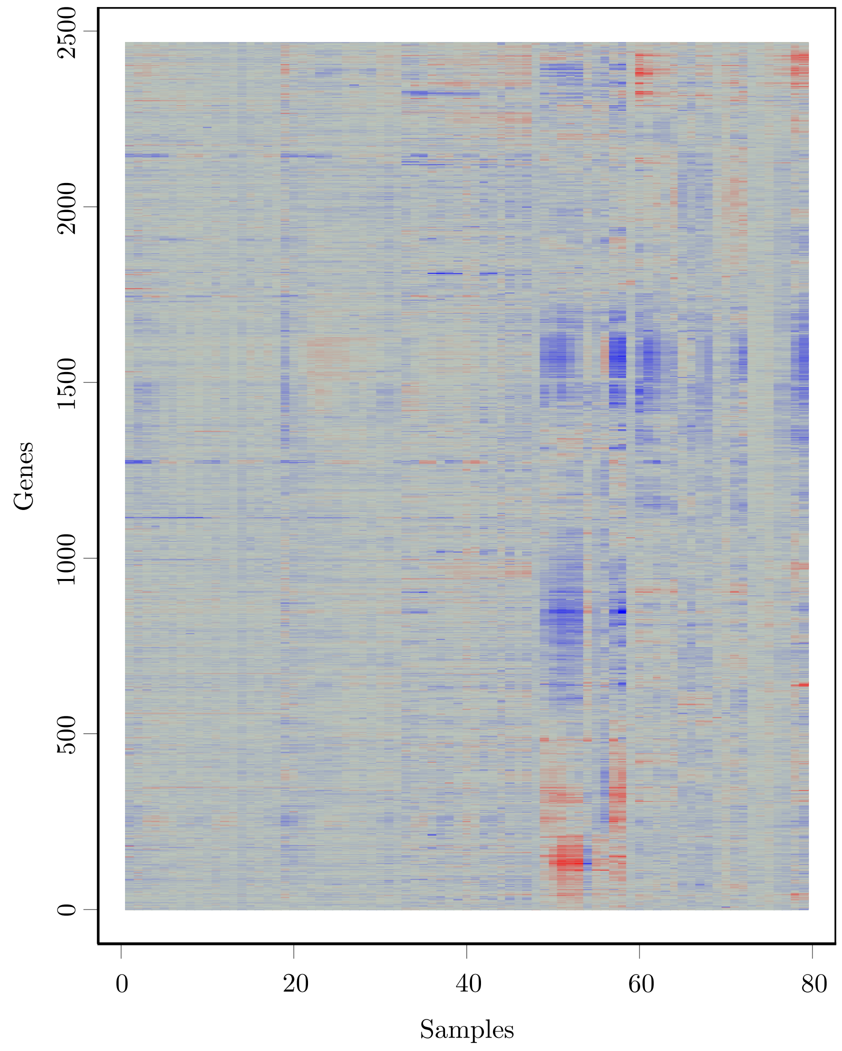

Figure 1. Yeast expression data.

If we remove the constraints $\sum_k \rho_{ik} = 1$ and $\sum_k \kappa_{jk} = 1$ from equation (1) we obtain a model which represents the data as a sum of possibly overlapping constant layers that do not have to cover the whole array.

In model (1) a layer describes a response μ_k that is shared by all genes in the layer, for all samples in the layer. It would also be biologically interesting to identify a subset of genes that have identical responses to a subset of conditions, or to identify a subset of conditions that produce identical responses across a subset of genes. The following models support one or the other or both of these notions

$$Y_{ij} \doteq \mu_0 + \sum_{k=1}^{K} (\mu_k + \alpha_{ik})\, \rho_{ik}\kappa_{jk}, \tag{2}$$

$$Y_{ij} \doteq \mu_0 + \sum_{k=1}^{K} (\mu_k + \beta_{jk})\, \rho_{ik}\kappa_{jk}, \tag{3}$$

$$Y_{ij} \doteq \mu_0 + \sum_{k=1}^{K} (\mu_k + \alpha_{ik} + \beta_{jk})\, \rho_{ik}\kappa_{jk}, \tag{4}$$

where each $\rho_{ik} \in \{0,1\}$, each $\kappa_{jk} \in \{0,1\}$ and, if α_{ik} is used, $\sum_i \rho_{ik}\alpha_{ik} = 0$ avoids overparameterization, with a similar condition on β_{jk}. The name "plaid model" describes the appearance of a color image plot of $\mu_k + \alpha_{ik} + \beta_{jk}$.

Each model (1) to (4) approximates the image by a sum of layers. We use the notation θ_{ijk} to represent μ_k, $\mu_k + \alpha_{ik}$, $\mu_k + \beta_{jk}$, or $\mu_k + \alpha_{ik} + \beta_{jk}$, as needed. We get a little more generality by mixing layer types, so that α_{ik} or β_{jk} might appear in some but not all θ_{ijk}. The model may then be written as a sum of layers,

$$Y_{ij} \doteq \sum_{k=0}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk}, \tag{5}$$

where θ_{ij0} describes the background layer. In some settings it might make sense to have a background layer with α_{i0} and/or β_{j0} terms.

We conclude this section by describing some interpretations of the layers. If $\rho_{ik} = 1$ for all i, but κ_{jk} is not 1 for all j, then layer k describes a cluster of samples. A converse description applies for a cluster of genes. If the layer for a cluster of genes contains a term β_{jk}, then that cluster of genes is a set of p-vectors centered near the vector $(\mu_k + \beta_{1k}, \ldots, \mu_k + \beta_{pk})$. If that layer also contains a term α_{ik}, then the genes cluster along a line segment through this center.

Each layer may represent the presence of a particular set of biological processes or conditions. The values of α_{ik} and β_{jk} provide orderings of the effects of layer k upon the genes and samples. Genes with larger values of $|\mu_k + \alpha_{ik}|$ are more greatly affected under the conditions of layer k than other genes within the layer. These effects are also greater for samples with larger values of $|\mu_k + \beta_{jk}|$. If $\mu_k + \alpha_{ik}$ is positive for one gene i and negative for another, then the first gene is upregulated and the second gene is downregulated within layer k.

A layer may contain some but not all genes, and some but not all samples. This can be interpreted as a group of genes that express themselves similarly within the given set of samples. Such layers combine gene clustering with variable selection on the samples. To avoid repetition, we will not describe converses to all of the features of the plaid model.

These interpretations are most straightforward if there is only one layer, or if multiple layers do not overlap significantly. Where layers overlap, the interpretations for layer k apply to the values of Y_{ij}, after first subtracting the other layers.

3. Estimation

Suppose we seek a plaid model with a small value of

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\Big(Y_{ij}-\theta_{ij0}-\sum_{k=1}^{K}\theta_{ijk}\rho_{ij}\kappa_{jk}\Big)^2. \tag{6}$$

For each layer k, there are $(2^n-1)(2^p-1)$ ways to select the participating genes and conditions. Even for modestly large n and p, it is impossible to investigate all of these selections, and so there is no assurance of finding the best fitting model for a given number K of layers. Gordon (1996) notes that many clustering problems are NP-hard, and we cannot expect the present problem to be simpler. For an up-to-date survey of optimization issues in clustering, see Hansen and Jaumard (1997). Even though an optimal fit is likely to be beyond our ability, we may still find that a numerical search provides an interpretable layer.

To simplify matters, suppose we have $K-1$ layers and are seeking the K'th layer to minimize the sum of squared errors. Let

$$Q=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(Z_{ij}-\theta_{ijK}\rho_{iK}\kappa_{jK}\right)^2, \tag{7}$$

where

$$Z_{ij}=Z_{ij}^{K-1}=Y_{ij}-\theta_{ij0}-\sum_{k=1}^{K-1}\theta_{ijk}\rho_{ij}\kappa_{jk} \tag{8}$$

is the residual from the first $K-1$ layers.

We adopt an iterative approach with each cycle updating θ values, ρ values and κ values in turn. Let $\theta^{(s)}$ denote all θ_{iK} values at iteration s. Similarly let $\rho^{(s)}$ and $\kappa^{(s)}$ represent all ρ_{iK} and κ_{jK} values at iteration s. After selecting starting values $\rho^{(0)}$ and $\kappa^{(0)}$ as described below, we follow S full update iterations. For $s=1,\ldots,S$, at iteration s, $\theta^{(s)}$ is updated from $\rho^{(s-1)}$ and $\kappa^{(s-1)}$, then $\rho^{(s)}$ is updated from $\theta^{(s)}$ and $\kappa^{(s-1)}$, and finally $\kappa^{(s)}$ is updated from $\theta^{(s)}$ and $\rho^{(s-1)}$. A reasonable alternative is to update θ values, then ρ values, then κ values, using at each stage the most recent values of the other variables. Instead, we opted to treat the genes and samples symmetrically in this iteration. The ρ and κ updates would be the same if they were done in the opposite order. The final update only changes $\theta^{(S+1)}$, so that the final layer values are $\rho^{(S)}$, $\kappa^{(S)}$, and $\theta^{(S+1)}$.

It is convenient to consider ρ and κ values in a continuous range, only forcing them to takes values 0 or 1 in the last one or several iterations. At intermediate stages, the values of θ_{ijK} describe a "fuzzy analysis of variance" in which ρ_{iK} and κ_{jK} are not necessarily 0 or 1. Replacing binary or integer variables by continuous ones is a very common device in integer programming (Wolsey (1998)), where it is known as relaxation. In what follows we drop the subscript K to simplify the presentation.

3.1. Updating θ_{ij}

To update the θ_{ij} values, given ρ_i and κ_j we minimize

$$Q = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(Z_{ij} - (\mu + \alpha_i + \beta_j)\,\rho_i\kappa_j\right)^2 \tag{9}$$

subject to identifying conditions

$$0 = \sum_{i=1}^{n}\rho_i^2\alpha_i = \sum_{j=1}^{p}\kappa_j^2\beta_j. \tag{10}$$

Straightforward Lagrange multiplier arguments show that

$$\mu = \frac{\sum_i\sum_j \rho_i\kappa_j Z_{ij}}{\left(\sum_i \rho_i^2\right)\left(\sum_j \kappa_j^2\right)}, \tag{11}$$

$$\alpha_i = \frac{\sum_j \left(Z_{ij} - \mu\rho_i\kappa_j\right)\kappa_j}{\rho_i\sum_j \kappa_j^2}, \tag{12}$$

$$\beta_j = \frac{\sum_i \left(Z_{ij} - \mu\rho_i\kappa_j\right)\rho_i}{\kappa_j\sum_i \rho_i^2}. \tag{13}$$

The update (11) for μ above is the same whether or not the K'th layer includes α_i or β_j, and updates (12) and (13) for α_i and β_j, respectively, are the same whether or not the other is included in the layer. If ρ_i is near zero, so observation i is effectively absent, then α_i is taken to be zero, and similarly when κ_j is close to zero, β_j is taken to be zero.

3.2. Updating ρ_i and κ_j

Given values for θ_{ij} and κ_j, the values for ρ_i that minimize Q are

$$\rho_i = \frac{\sum_j \theta_{ij}\kappa_j Z_{ij}}{\sum_j \theta_{ij}^2\kappa_j^2}, \tag{14}$$

and similarly, given θ_{ij} and ρ_i, the minimizing values for κ_j are

$$\kappa_j = \frac{\sum_i \theta_{ij}\rho_i Z_{ij}}{\sum_i \theta_{ij}^2 \rho_i^2}. \tag{15}$$

The quantities ρ_i and α_i pertaining to gene i are updated only with data from that gene. This makes the updates particularly fast.

We do not allow the values ρ_i and κ_j to move too quickly towards 0 or 1, as this might "lock in" a suboptimal initial condition. Instead, at iteration s, ρ_i and κ_j are replaced by $0.5 + s/(2S)$ if they are larger than 0.5, and by $0.5 - s/(2S)$ otherwise.

3.3. Starting values

In order to search in the residuals $Z_{ij} = Z_{ij}^{K-1}$ for the K'th layer of the model, we need starting values of ρ_i and κ_j. We have considered starting values all equal to 0.5, and starting values randomly generated near 0.5. The most successful starting values have been found as follows: fix $\theta_{ijK} = 1$ for all i and j, and perform several iterations updating ρ and κ values only. The ρ and κ vectors then approach multiples of the singular vectors u_1 and v_1, respectively, of the matrix Z corresponding to the largest singular value λ_1. The matrix $\lambda_1 u_1 v_1'$ is the closest rank one approximation to Z as measured by the sum of squared matrix entry errors. The iteration used to get ρ and κ is known as the the power method. The starting values are obtained by replacing the singular vectors ρ and κ by their absolute values, scaled so that they sum to $n/2$ and $p/2$ respectively.

The search for the largest singular values itself needs starting ρ and κ values. The search can fail if the initial ρ (respectively κ) is orthogonal to u_1 (respectively v_1). We start the iteration with each ρ_i and κ_j equal to 0.5 plus very small random numbers, to reduce the likelihood of such a failure.

3.4. Further issues

For a given value of k, the values of μ_k, α_{ik}, and β_{jk} are easy to optimize with all other parameters in the model fixed. For a set of K layers, it is simple to re-estimate all of the θ_{ijk}, by cycling through $k = 1, \ldots, K$ in turn, several times. These backfitting cycles conduct a partial re-optimization, updating all of the θ_{ijk} parameters but not the ρ and κ parameters. They tend to be extremely fast, especially if the layers are small. We typically run backfitting after each new layer has been added to the model.

Deciding how large K should be is the subject of Section 4 below.

In the basic algorithm ρ_i tends to approach 1 instead of 0 if including gene i in the layer reduces the total sum of squared errors. This can happen because the gene fits the layer well, or because the gene has a very large residual, a small proportion of which is explained by the layer. The algorithm has an option to trim away such genes. Under this option, any gene whose sum of squared residuals is not reduced by a user specified proportion is released from the layer (ρ_i set to 0), possibly to be included in some later layer.

A variant on the algorithm updates ρ_i and κ_j by $0.5 \pm \Delta_s$ where $\Delta_s = \min(s/(2(S-T)), 0.5)$ for some nonnegative integer $T < S$. The effect is that, of S steps, the final T of them pick ρ_i and κ_j values in $\{0,1\}$. We use 6 iterations to find the starting values, then 10 iterations increasing Δ_s to 0.5, and finally 3 iterations where ρ_i and κ_j are placed in the set $\{0,1\}$. The algorithm does not appear to be very sensitive to these choices.

In another variant of the algorithm, the updates of ρ_i and κ_j are as in (14) and (15), except that the quantity $\sum_{i=1}^{n}\sum_{j=1}^{p}\rho_{ik}^2\kappa_{jk}^2\theta_{ijk}^2$ is added to the denominator of (14) if there are any α_{ik} terms, and to the denominator of (15) if there are any β_{jk} terms. This builds in a preference for layers with smaller numbers of genes or experiments.

A layer can be easier to interpret if every $\mu + \alpha_i$ and every $\mu + \beta_j$ has the same sign (that of μ). The algorithm has a "unisign" option that builds in a preference of this kind. Under this option, each time ρ_i is updated the algorithm checks whether $\mu + \alpha_i$ and μ are of the same sign. If not, the value of ρ_i is reduced.

4. Regularization

A greedy algorithm that adds one layer at a time requires a stopping rule. We suppose that as each layer is removed from the data, the residual becomes more and more like unstructured noise. We propose a simple rule that will give only a small number of extra layers once the data have been reduced to noise.

First, we measure the size or importance of layer k by the sum of squares $\sigma_k^2 = \sum_{i=1}^{n}\sum_{j=1}^{p}\rho_{ik}\kappa_{jk}\theta_{ijk}^2$. We would like to accept a layer if it is significantly larger than what we would find in noise. The distribution of σ_k^2 on noise is not known. Instead of using that distribution, we expand on a permutation technique (called random 3) in Eisen, Spellman, Brown and Botstein (1998). Let Z_{ij} be the residual matrix in which we search for layer k. For each $r = 1, \ldots, R$, let $\tilde{Z}_{ij}^{(r)}$ be a matrix obtained by randomly permuting every row of Z_{ij} and then randomly permuting every column of the result. All $(n+p)R$ permutations are independent and all are uniformly distributed. This means that, when permuting column entries, each of $n!$ possible permutations is equally probable, and similarly for the $p!$ possible row permutations. Let $\tilde{\sigma}_k^{2,r}$ denote the size of the layer found by the algorithm in the randomized data $\tilde{Z}_{ij}^{(r)}$.

The stopping rule is: if $\sigma_k^2 > \max_{1\le r\le R}\tilde{\sigma}^{2,r}$ and $k < K_{\max}$, add the new layer k to the model, otherwise stop. Here $K_{\max}$ is a prespecified limit on the number of layers in the model.

One way to characterize noise is to say that the data values are independent of row and column labels. The chance of accepting a layer in such noise is $1/(R+1)$. It is reasonable to suppose that the probability of accepting m or more layers from noise is approximately $(R+1)^{-m}$, and that the expected number of layers accepted after the residual has become noise is approximately $1/R$. For instance, when searching in noise with $R = 3$, there is approximately a 3/4 probability of finding 0 noise layers, a 3/16 probability of finding 1 layer, a 3/64 probability of finding 2 layers, and a 1/64 probability of finding more than 2 layers. These approximations would be the exact if subtracting a layer from noise left a residual that was noise.

One might choose $R = 99$ (or 19) to give only a 1% (respectively 5%) chance of accepting a layer in noise. We prefer to work with R between 1 and 4, depending on the size of the data set. Computational costs are proportional to $R + 1$, so this represents a worthwhile speedup, at the expense of slightly raising the expected number of noise layers. By speeding up the algorithm, there is more time to explore variations of the algorithm.

In practice we have seen that this stopping rule sometimes gives a large number of layers. When this happens each real-data layer is always somewhat bigger than the ones fitted to permuted data, but as k increases both σ_k^2 and $\tilde{\sigma}_k^{2,r}$ usually decrease. One interpretation is that such layers are statistically significant even though they may not be practically significant. The data analyst could reasonably delete them from the model, or not bother to interpret them. In other examples, the stopping rule gives a small number of layers.

There is a small risk that this rule will stop too soon because an unusually highly structured random permutation was generated. This risk can be reduced by accepting a layer if at most a of R randomized layers are larger than it. The probability of accepting a layer fit to noise is $(a+1)/R$ and the expected number of layers found in noise is then close to $(a+1)/(R-a)$. We have found the original $a = 0$ stopping rule to be acceptable.

5. Food Example

The first example uses nutritional data from 961 different foods. These data were found at http://www.ntwrks.com/~mikev/chart1.html. For each food, the following were recorded: grams of fat, calories of food energy, grams of carbohydrate, grams of protein, milligrams of cholesterol, grams of saturated fat, and the weight of the food item in grams. Some foods appear in different serving sizes, as for example a piece of cherry pie, or an entire pie. To measure food composition, each of the first six variable values was divided by the weight of the food item, yielding the value X_{ij} for food i and composition variable j. The calorie values have a variance that is over 850 times as large as the saturated fat values. To equalize the variance, the data were centered and scaled, leading to $Y_{ij} = (X_{ij} - \bar{X}_{\cdot j})/S_j$,where $\bar{X}_{\cdot j} = (1/n)\sum_{i=1}^{n} X_{ij}$ and $S_j^2 = (1/n)\sum_{i=1}^{n}(X_{ij} - \bar{X}_{\cdot j})^2$.

By taking out the mean of each column, the background layer corresponds to foods and food measures near the column means. Some foods are unusually rich by some measures. For example, egg yolks are about 18 standard deviations above the mean cholesterol level. A few foods (like salt) have low values in all measures. If we had not subtracted out the columns means, such foods would have been at the background level.

Our algorithm was as follows: we searched for up to 10 layers containing both α_i and β_j components. We used $R = 4$ shuffles in the stopping rule, opted to prefer a common sign for $\mu + \alpha_i$ and for $\mu + \beta_j$ within each layer, and released any row (or column) from a layer if joining the layer did not reduce its sum of squares by at least 51%. All 10 layers were larger than noise. We decided to drop the last 5 layers because they were small. The layer sizes during and after search are shown in Table 1.

Table 1. Layer sizes for the food example. σ_k^2 are the sizes found during greedy training. $\widetilde{\sigma}_{k,r}^2$ are the corresponding sizes found on randomized data. The final two columns show σ_k^2 for $K = 10$ and 5 layers respectively, after backfitting.

Original		**Randomized**				**After backfitting**	
k	σ_k^2	$\widetilde{\sigma}_{k,1}^2$	$\widetilde{\sigma}_{k,2}^2$	$\widetilde{\sigma}_{k,3}^2$	$\widetilde{\sigma}_{k,4}^2$	$K = 10$	$K = 5$
1	1799.02	459.85	809.05	987.79	1057.44	2634.25	1799.02
2	944.44	0.00	356.79	370.94	0.00	1325.25	759.94
3	811.28	417.89	355.38	377.96	393.91	788.88	831.14
4	667.91	256.21	192.84	244.49	369.22	420.74	669.50
5	413.23	198.28	184.64	95.32	225.06	932.80	775.76
6	152.05	104.49	67.86	101.17	75.61	400.03	
7	120.72	50.02	75.30	59.99	79.29	201.04	
8	83.35	57.07	34.07	67.08	65.04	331.64	
9	100.14	35.53	46.33	30.83	28.75	168.63	
10	61.47	23.13	23.14	44.73	35.49	120.48	

Layer 1 contains 180 foods and the variables fat proportion, saturated fat proportion and calories per gram, as shown in Table 2. Because $\mu_1 = 1.54$, these foods are about 1.54 standard deviations above the mean. The values of β_{1j} range from 0.14 standard deviations for fat proportion to -0.09 standard deviations for calories per gram. These are high calorie fatty foods. The 20 foods with the highest α_i are listed in Table 2. This layer also contains more oils, margarines, nuts and some dairy products.

Table 2. Top 20 of 180 foods in layer 1.

$\mu = 1.54$	**Layer 1**
α_i	**Food**
2.86	LARD 1 CUP
2.83	LARD 1 TBSP
2.81	BUTTER, SALTED 1/2 CUP
2.81	BUTTER, UNSALTED 1/2 CUP
2.76	BUTTER, SALTED 1 TBSP
2.76	BUTTER, UNSALTED 1 TBSP
2.73	BUTTER, SALTED 1 PAT
2.73	BUTTER, UNSALTED 1 PAT
2.13	FATS, COOKING/VEGETBL SHORTENG1 TBSP
2.11	FATS, COOKING/VEGETBL SHORTENG1 CUP
1.76	SOYBEAN-COTTONSEED OIL, HYDRGN1 TBSP
1.75	SOYBEAN-COTTONSEED OIL, HYDRGN1 CUP
1.73	PEANUT OIL 1 TBSP
1.70	PEANUT OIL 1 CUP
1.62	SOYBEAN OIL, HYDROGENATED 1 TBSP
1.60	SOYBEAN OIL, HYDROGENATED 1 CUP
1.55	OLIVE OIL 1 TBSP
1.53	OLIVE OIL 1 CUP
1.51	CORN OIL 1 TBSP
1.49	CORN OIL 1 CUP
β_j	**Nutritional variable**
0.13	Fat Proportion
-0.04	Saturated Fat Proportion
-0.09	Calories per Gram

Layer 2 is described in Table 3. This layer contains foods that are high in cholesterol and especially high in protein. For protein the value of $\mu + \beta$ is 2.08 standard deviations, and for cholesterol it is 0.53 standard deviations. This layer also contains some more meats, seafoods, nuts and cheeses.

Layer 3, shown in Table 4, contains foods that are low in all of the variables except possibly cholesterol. Most are also low in cholesterol, as indeed are most of the foods not in this layer.

Layer 4 is presented in Table 5. It contains foods that are on average 1.42 standard deviations above the mean in proportion of carbohydrate. The appearance of breakfast cereals near to pure sugar reflects that both have high carbohydrate counts. A data set that broke carbohydrates into fiber, starch and sugar or that included vitamin content would be likely to distinguish these foods.

Table 3. Top 20 of 143 foods in layer 2.

$\mu = 1.31$	**Layer 2**
α_i	**Food**
2.88	GELATIN, DRY 1 ENVELP
1.40	BEEF HEART, BRAISED 3 OZ
1.25	SEAWEED, SPIRULINA, DRIED 1 OZ
1.04	PARMESAN CHEESE, GRATED 1 OZ
1.04	PARMESAN CHEESE, GRATED 1 CUP
1.00	LAMB,CHOPS,ARM,BRAISED,LEAN 1.7 OZ
0.96	SHRIMP, CANNED, DRAINED 3 OZ
0.93	PARMESAN CHEESE, GRATED 1 TBSP
0.79	LAMB,CHOPS,ARM,BRAISED,LEAN+FT2.2 OZ
0.78	PORK SHOULDER, BRAISD, LEAN 2.4 OZ
0.62	PORK CHOP, LOIN, BROIL, LEAN 2.5 OZ
0.61	BEEF, CKD,BTTM ROUND,LEAN ONLY2.8 OZ
0.61	BEEF, CKD,CHUCK BLADE,LEANONLY2.2 OZ
0.57	VEAL RIB, MED FAT, ROASTED 3 OZ
0.54	CHICKEN, ROASTED, DRUMSTICK 1.6 OZ
0.53	CHICKEN, FRIED, FLOUR, BREAST 3.5 OZ
0.53	BUTTERMILK, DRIED 1 CUP
0.50	YEAST, BAKERS, DRY, ACTIVE 1 PKG
0.50	TUNA, CANND, DRND,WATR, WHITE 3 OZ
0.50	PORK, CURED, BACON, REGUL,CKED3 SLICE
β_j	**Nutritional variable**
0.78	Protein Proportion
−0.78	Cholesterol Proportion x 1000

Table 4. Top 20 of 429 foods in layer 3.

$\mu = -0.60$	**Layer 3**
α_i	**Food**
−0.21	COLA, DIET, ASPARTAME ONLY 12 FL OZ
−0.21	PARSLEY, FREEZE-DRIED 1 TBSP
−0.21	COLA, DIET, ASPRTAME + SACCHRN12 FL OZ
−0.21	COFFEE, BREWED 6 FL OZ
−0.21	SALT 1 TSP
−0.21	TEA, BREWED 8 FL OZ
−0.21	CLUB SODA 12 FL OZ
−0.21	COLA, DIET, SACCHARIN ONLY 12 FL OZ
−0.21	LETTUCE, BUTTERHEAD, RAW,LEAVE1 LEAF
−0.21	TEA, INSTANT,PREPRD,UNSWEETEND8 FL OZ
−0.21	COFFEE, INSTANT, PREPARED 6 FL OZ
−0.19	PICKLES, CUCUMBER, DILL 1 PICKLE
−0.18	CELERY, PASCAL TYPE, RAW,STALK1 STALK
−0.17	BEEF BROTH, BOULLN, CONSM,CNND1 CUP
−0.17	ONION SOUP, DEHYDRATD, PREPRED1 PKT
−0.17	BEER, LIGHT 12 FL OZ
−0.16	CUCUMBER, W/ PEEL 6 SLICES
−0.16	LETTUCE, CRISPHEAD, RAW,WEDGE 1 WEDGE
−0.16	LETTUCE, CRISPHEAD, RAW, HEAD 1 HEAD
−0.16	VINEGAR, CIDER 1 TBSP
β_j	**Nutritional variable**
−0.24	Calories per Gram
0.02	Protein Proportion
0.04	Carbohydrate Proportion
0.07	Fat Proportion
0.11	Saturated Fat Proportion

Table 5. Top 20 of 270 foods in layer 4.

$\mu = 1.42$	**Layer 4**
α_i	**Food**
1.63	SUGAR, POWDERED, SIFTED 1 CUP
1.63	SUGAR, WHITE, GRANULATED 1 PKT
1.63	SUGAR, WHITE, GRANULATED 1 TBSP
1.61	SUGAR, WHITE, GRANULATED 1 CUP
1.58	HARD CANDY 1 OZ
1.48	SUGAR, BROWN, PRESSED DOWN 1 CUP
1.44	ONION POWDER 1 TSP
1.44	FONDANT, UNCOATED 1 OZ
1.30	SUGAR FROSTED FLAKES, KELLOGG 1 OZ
1.30	JELLY BEANS 1 OZ
1.30	SUPER SUGAR CRISP CEREAL 1 OZ
1.25	COCA PWDR W/O NONFAT DRY MILK 3/4 OZ
1.23	CAROB FLOUR 1 CUP
1.16	FROOT LOOPS CEREAL 1 OZ
1.16	RICE KRISPIES CEREAL 1 OZ
1.16	GUM DROPS 1 OZ
1.16	SUGAR SMACKS CEREAL 1 OZ
1.16	TRIX CEREAL 1 OZ
1.11	CINNAMON 1 TSP
1.06	POPCORN, SUGAR SYRUP COATED 1 CUP
β_j	**Nutritional variable**
0.00	Carbohydrate Proportion

Table 6. Top 20 of 59 foods in layer 5.

$\mu = 2.39$	**Layer 5**
α_i	**Food**
16.18	EGGS, RAW, YOLK 1 YOLK
6.13	CHICKEN LIVER, COOKED 1 LIVER
4.03	EGGS, COOKED, FRIED 1 EGG
3.72	BEEF LIVER, FRIED 3 OZ
3.54	EGGS, COOKED, HARD-COOKED 1 EGG
3.54	EGGS, RAW, WHOLE 1 EGG
3.51	EGGS, COOKED, POACHED 1 EGG
2.45	EGGS, COOKED, SCRAMBLED/OMELET1 EGG
0.51	BUTTER, UNSALTED 1 TBSP
0.51	BUTTER, SALTED 1 TBSP
0.50	POUND CAKE, COMMERCIAL 1 SLICE
0.49	BUTTER, UNSALTED 1 PAT
0.49	BUTTER, SALTED 1 PAT
0.49	POUND CAKE, COMMERCIAL 1 LOAF
0.47	BUTTER, UNSALTED 1/2 CUP
0.47	BUTTER, SALTED 1/2 CUP
0.41	SHRIMP, FRENCH FRIED 3 OZ
0.33	BRAUNSCHWEIGER 2 SLICES
-0.03	CHEESECAKE 1 CAKE
-0.03	CHEESECAKE 1 PIECE
β_j	**Nutritional variable**
0.00	Cholesterol Proportion x 1000

Layer 5, shown in Table 6, has foods that are on average 2.39 standard deviations above the mean in cholesterol. The cholesterol distribution is extremely skewed.

6. Foreign Exchange Example

The raw data for this example are monthly foreign exchange values. For months $i = 0, 1, \ldots, n$ and currencies $j = 1, \ldots, p$, $X_{i,j}$ denotes the number of units of that currency that one US dollar purchased in that month. The values that we study are the monthly logarithmic returns to the currencies $Y_{ij} = -\log(X_{i,j}/X_{i-1,j})$, $i = 1, \ldots, n$, $j = 1, \ldots, p$. The raw data, obtained from Bloomberg, covers 277 months from January 1977 to January 2000 inclusive, and so there are $n = 276$ returns. There are $p = 18$ currencies, corresponding to Belgium, Canada, Denmark, Netherlands, Finland, France, Germany, India, Japan, Malaysia, Mexico, Norway, South Africa, Spain, Sri Lanka, Sweden, Switzerland, and the United States.

Because all of the data are in the same units, we chose not to make the variances equal. We also did not adjust for the mean. In this way, we have made the background layer correspond to the US dollar.

Setting row and column release criteria to 0.51 and preferring constant sign, the algorithm terminates after finding 3 layers. Layer one describes 71 months when the US dollar strengthened against 11 other currencies. Layer two describes 60 months when the Mexican peso weakened against the US dollar. Layer three describes 69 months when the US dollar weakened against the same 11 currencies from layer 1. The sizes of the layers are $\sigma_1^2 = 1.30$, $\sigma_2^2 = 1.82$ and $\sigma_3^2 = 1.11$. After fitting 3 layers, the residual has sum of squares equal to 2.39. The fourth layer, found in a greedy search, has months in which the currency of Sri Lanka weakened against the US dollar, but one of three shuffled layers had a larger size.

The 11 currencies in layers 1 and 3 were those of Belgium, Denmark, Finland, France, Germany, Japan, Netherlands, Norway, Spain, Sweden, and Switzerland. When the dollar was weakening, μ took the value 0.0349, and β was in a tight range from -0.0076 (Spain) to 0.0079 (Switzerland). The worst month for the dollar was October 1978 with $\alpha = 0.0409$. When the dollar was strengthening, μ took the value -0.0365. Japan lost the least ground in those months ($\beta = 0.0124$) while the other currencies ranged from -0.0054 (Switzerland) to 0.0041 (Finland).

Canada does not appear in these layers. The Canadian currency is closely tied to the US currency, which represents the background.

Layer 2 has $\mu = -0.1073$, corresponding to a more than 10% decline in the Mexican currency in one month. The distribution of the α values for this layer is very skewed. The extreme months are December 1982 ($\alpha = -0.55$), June 1977 ($\alpha = -0.50$) and February 1982 ($\alpha = -0.40$). There are 5 other months between -0.05 and -0.35.

7. Gene Expression Data

Figure 1 shows yeast gene expression data used by Eisen et al. (1998) The data are available at http://rana.stanford.edu/clustering. The columns represent timepoints within each of ten experimental series. These experiments are reported in DeRisi, Iyer and Brown (1997), Spellman, Sherlock, Zhang, Iyer, Anders, Eisen, Brown, Botstein and Futcher (1998), and Chu, DeRisi, Eisen, Mulholland, Botstein, Brown and Herskowitz (1998).

The columns in these data are denoted by the following prefixes: alpha (columns 1-18), Elu (19-32), cdc (33-47), spo (48-53), spo5 (54-56), spo- (57-58), heat (59-64), dtt (65-68), cold (69-72), diau (73-79). Experiments one to three examine the mitotic cell cycle. Experiments four to six track different strains of yeast during sporulation. Experiments seven to nine track expression following exposure to different types of shocks. Experiment ten studies the diauxic shift. Each of the 2467 rows represents a single probe on the microarray designed to detect the expression level of a particular gene. The rows are ordered according to the results of the hierarchical clustering algorithm to illustrate the relationships revealed by that approach. The colors in the image (red=high, blue=low) correspond to values of $Y_{ij} = \log_2 X_{ij}$, where X_{ij} is a measurement representing the expression level of gene i obtained from a scanned image of the microarray used to assay sample j. The original downloaded data contained values of Y_{ij}. Values for missing data (1.9% of the data) were imputed using the sum of the row and column means less the overall mean. Annotation of the genes in the downloaded file was slightly edited to save space.

We used both gene and sample effects in the background and in the layers. This choice of background layer acknowledges that genes and samples both have different expression levels, and focuses the search for biological interpretation on their interactions. We searched for up to 40 layers with the unisign option on, row and column release criteria set to 0.5, and shuffling 3 times for each layer. After the 34th layer, the algorithm was unable to find a layer that retained any rows under the release criterion.

Overall, the 34 layers and the background contained 5568 parameters, fewer than 3% of the number of observations. Figure 2 shows the complete fitted model, which recovers much of the visually-apparent structure in the original data. Layers tend to decrease both in size of effect and number of genes as the algorithm proceeds. Toward the end, the algorithm discards large numbers of genes due to the release criterion. Not surprisingly, the typical sample belongs to more layers than does the typical gene. The number of columns per layer remains more or less constant throughout the analysis. Background alone accounted for 28% of the genes and of the samples in the data (see Table 7). An additional 42% of the genes were in a single layer. Overall, 88% of the data was explained by background alone. There was little overlap among the layers, with fewer than 1% of the data falling into more than one layer.

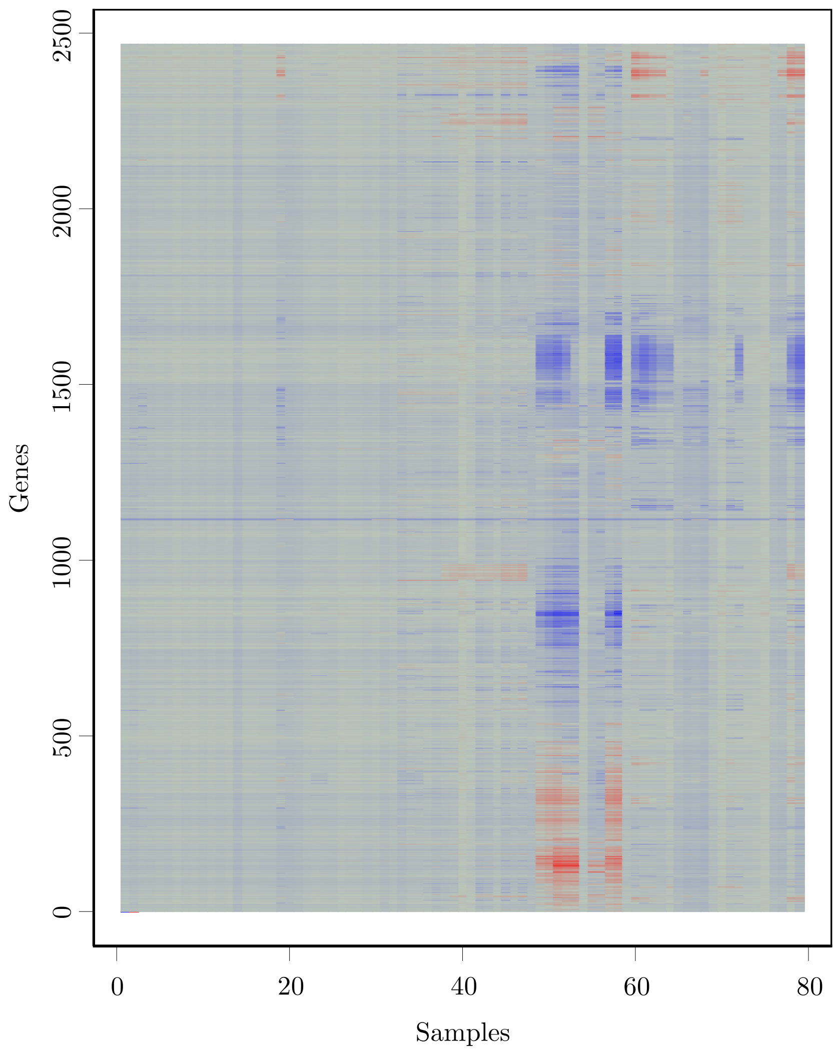

Figure 2. Fitted model for yeast data.

Table 7. Yeast summary showing the numbers of genes, samples and observations appearing in 0, 1 or more layers.

No. of Layers	**Genes**	**Samples**	**Observations**
0	703	22	170703
1	1031	5	22872
2	579	2	1307
3	142	11	11
4-18	12	39	0
Total	2467	79	194893

The plaid model consistently puts columns from the same experimental series together within layers. Table 8 shows the column effects in the first six layers. Only in layer 6 is an intermediate timepoint (column 41, cdc 170) excluded when timepoints both before and after are included in the layer. Similar patterns were seen in subsequent layers. The sporulation data enter into four of the first six layers because of the greater variability in those experiments.

Table 8. Column effects for the first 6 layers of the yeast expression data. Columns that do not appear in these layers are omitted, unless they fall between two timepoints included in a single layer.

Sample effects $(\mu + \beta_j)$ in first 6 layers						
Sample	**1**	**2**	**3**	**4**	**5**	**6**
19 Elu 0				0.74		
39 cdc 130						0.44
40 cdc 150						0.29
41 cdc 170						
42 cdc 190						0.52
43 cdc 210						0.46
44 cdc 230						0.53
45 cdc 250						0.82
46 cdc 270						0.64
47 cdc 290						0.89
49 spo 2	0.72	−1.18	−0.81			
50 spo 5	1.10	−1.18	−1.21			
51 spo 7	1.36	−1.32	−1.12			0.93
52 spo 9	1.08	−0.75	−1.33			0.99
53 spo 11	1.06		−1.12			0.92
55 spo5 7						0.94
56 spo5 11						0.76
57 spo-early	1.19	−2.14	−1.03			
58 spo-mid	1.41	−2.19	−1.43			
60 heat 10		−1.19		1.57	−1.06	
61 heat 20		−1.70		1.10	−1.15	
62 heat 40		−1.23		0.61	−1.00	
63 heat 80		−0.70		0.53	−0.55	
64 heat 160		−0.80			−0.65	
66 dtt 30						0.55
67 dtt 60						
68 dtt 120				0.47	−0.31	
71 cold 40					−0.59	
72 cold 160		−0.90				
77 diau e				0.55		
78 diau f		−1.20		1.30	−0.64	
79 diau g		−1.60		1.42	−0.87	0.48

Twenty-five layers capture at least one of the 47 microarrays from the first three experimental series, all of which were designed to track the mitotic cell division cycle. The difference between these series is the laboratory method used to synchronize the cell cycle at time zero. In general, synchronization effects appear to dominate cell cycle effects in determining layer membership. No layer contains more than one microarray from each of the three series. However, six layers have six to twelve microarrays from one synchronization method and none from the other two methods. Five of these six are selected from the third series, which is the most evident of these series under the plaid model. The evidence here suggests that cell cycle effects shared across synchronization methods are modest relative to differences among the experimental series. Due to its overlapping layers, the plaid model has the potential to simultaneously identify layers based on synchronization method and layers based on aspects of the cell cycle.

Layers 1 and 3 contain the same seven samples and share no genes in common. In Figure 1, these layers correspond to the band running through columns 49–58. The timepoints are hours 2, 5, 7, 9 and 11 during sporulation in one yeast strain, and hours 5 and 7 in a second yeast strain (these were the only times at which the second strain was assayed). On average, the 567 genes in layer 1 are upregulated to 219% of their background levels, whereas the 251 genes of layer 3 are downregulated to 44% of their background level.

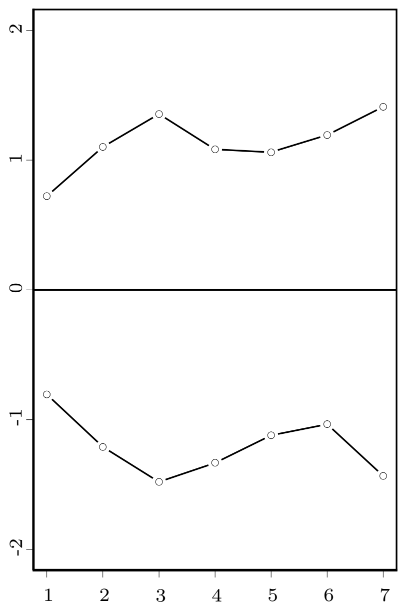
Sample Effects in Layers 1 and 3.

Figure 3

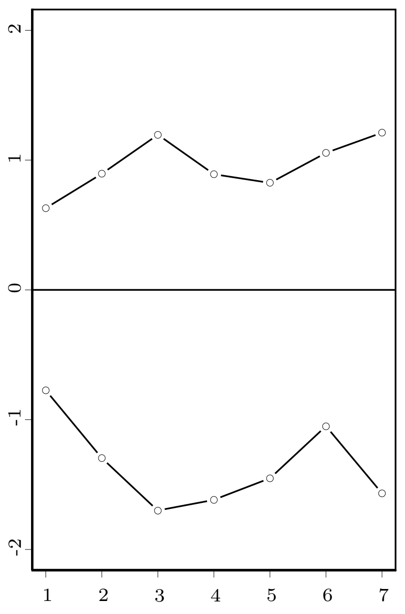
Data Means in Layers 1 and 3.

Figure 4

Figure 3 shows how the effects of layers 1 and 3 mirror each other across the selected samples. Figure 4 shows the means of the data for the same genes and samples. The mirroring effect is not visible in the original data, but is visible in the plaid model after subtraction of the background and other layers.

Table 9 shows the 12 most affected genes under layers 1 and 3. Layer 1 includes many genes involved in the cell cycle. Layer 3 includes many genes involved in glycolysis.

Table 9. Top 12 genes of layers 1 and 3.

$\mu = 1.13$	**Layer 1**
α_i	**Gene, known function**
3.34	ECM11, cell wall biogenesis
2.77	LEU1, leucine biosynthesis, 3-isopropylmalate dehydratase
2.65	PDS1, cell cycle, anaphase inhibitor (putative)
2.35	CDC5, cell cycle, G2-M protein kinase
2.02	CIK1, cytoskeleton, spindle pole body associated protein
1.77	CLB5, cell cycle, G1-S cyclin
1.64	PCH2 meiosis, checkpoint
1.56	STU2 cytoskelton, spindle pole body component
1.56	BAT1, branched chain amino acid, transaminase
1.53	ORC3, DNA replication, origin recognition complex, ...
1.56	APC4, cell cycle, anaphase-promoting complex subunit
1.51	MIP6, MRNA export, putative, RNA-binding proteinlization
$\mu = -1.20$	**Layer 3**
α_i	**Gene, known function**
-2.11	TDH1, glycolysis, glyceraldehyde-3-phosphate dehydrogenase 1
-2.02	TKL1, pentose phosphate cycle, transketolase
-1.99	PGK1, glycolysis, phosphoglycerate kinase
-1.97	ENO2, glycolysis, enolase II
-1.86	TDH2, glycolysis, glyceraldehyde-3-phosphate dehydrogenase 2
-1.79	YGP1, diauxic shift, response to nutrient limitation
-1.70	TDH3, glycolysis, glyceraldehyde-3-phosphate dehydrogenase 3
-1.68	TPI1, glycolysis, triophosphate isomerase
-1.59	FBA1, glycolysis, aldolase
-1.52	BUD7, bud site selection
-1.49	GPM1, glycolysis, phosphoglycerate mutase
-1.42	ALD6, ethanol utilization, acetaldehyde dehydrogenase

Layer 2 is dominated by genes that produce ribosomal proteins involved in protein synthesis in which MRNA is translated. The layer contains 14 samples, 216 genes and has $\sigma_2^2 = 6437$. It includes 114 of the 124 non-mitochondrial ribosomal proteins identified in the data. Most of the ribosomal protein genes (107) are among the 130 most affected genes in layer 2. Layer 2 includes all five of the acidic ribosomal proteins and none of the 49 mitochondrial ribosomal proteins. Layer 2 contains several other genes also involved in translation and in transcription. The genes in layer 2 are downregulated ($\mu = -1.29$) reaching 22% to 66% of their background levels. This downregulation occurs within 14 samples from the earlier stages of sporulation, in the diauxic shift and following cold and heat shock (see Table 8).

Table 10 lists all layers containing more than one ribosomal protein. Nine of the ten non-mitochondrial ribosomal proteins that do not appear in layer 2 appear in layer 5. Interestingly, seven of these form a contiguous set (numbers 1510-1517) in the hierarchical-clustering order, separated in that analysis from the main group of ribosomal proteins by four other genes. Layer 5 is less strongly downregulated ($\mu = -0.76$) than layer 2 and contains none of the sporulation samples (see Table 8).

Table 10. Types of ribosomal protein within all layers containing more than one such gene.

Types of ribosomal proteins by layer												
Layer	**2**	**3**	**5**	**6**	**11**	**12**	**13**	**14**	**15**	**24**	**29**	**31**	**All**
Genes	216	251	87	47	98	110	111	253	54	46	39	89	2467
Acidic	5	0	0	0	0	0	0	0	0	1	0	0	5
Mito.	0	7	3	8	2	0	22	10	1	0	1	0	49
Other	109	3	9	1	0	6	1	14	3	6	3	13	119
All RPs	114	10	12	9	2	6	23	24	4	7	4	13	173

Later layers also exhibit biological patterns. For example, the 89 genes in layer 32 include 18 of the 33 proteasome subunits. Interestingly, the most down-regulated gene in layer 32 is FET3 ($\mu_{32} + \alpha_{i,32} = -1.51$), which encodes a cell surface ferroxidase involved in transport. FET3 is also the most downregulated non-ribosomal protein in layer 2 ($\mu_2 + \alpha_{i,2} = -1.86$). Layers 2 and 32 share only one other gene ASN2, an asparagine synthetase involved in asparagine biosynthesis. Neither FET3 nor ASN2 appear in any other layers. An interpretation is that FET3 and ASN2 behave like ribosomal proteins under the conditions represented within layer 2 and like the proteasome subunits under the conditions represented in layer 33. These latter conditions include the mitotic cell cycle (columns 23, 24, 33, 34, 35) and later stages of sporulation (columns 52, 53, 55, 56).

Some of the later layers are quite small. Layer 30 contains fifteen samples and only seven genes, three of which are cytoskeleton genes. The complete layer is shown in Table 11.

Table 11. Layer 30 of the yeast expression data.

$\mu = 0.49$	**Layer 30**
$\mu + \alpha_i$	**Gene, known function**
0.72	DAK1, carbohydrate metabolism, dihydroxyacetone kinase
0.72	AIP1, cytoskeleton, actin cortical patch component
0.53	CAP2, cytoskeleton, F-actin capping protein subunit
0.45	CYP5, protein folding, peptidyl-prolyl cis-trans isomerase
0.42	PUP1, protein degradation, 20s proteasome subunit (beta2)
0.29	PDX1, glycolysis, pyruvate dehydrogenase
0.27	MYO3, cytoskeleton, myosin, class I
$\mu + \beta_j$	**Column, sample**
0.68	60, heat 10
0.66	61, heat 20
0.64	68, dtt 120
0.60	67, dtt 60
0.55	19, Elu 0
0.50	31, Elu 360
0.50	78, diau f
0.49	54, spo5 2
0.48	45, cdc 250
0.46	47, cdc 290
0.43	30, Elu 330
0.36	32, Elu 390
0.36	65, dtt 15
0.35	46, cdc 270
0.23	57, spo- early

8. Comparisons

This section surveys the literature and describes methods related to the plaid model. We begin by introducing the singular value decomposition of a matrix. It has long been known (Eckart and Young (1936)) that truncating the SVD of a matrix Y to k terms produces the rank k matrix closest to Y as measured by summed squared difference. The SVD structure provides a common thread that Lee and Seung (1999) used to link a number of data analysis methods. We add some more methods to their list, and we use this idea to compare the plaid model to the others. Space does not permit an exhaustive discussion of the details of all the algorithms.

8.1. Singular value decomposition

The singular value decomposition (SVD) of a matrix Y is a sum

$$\sum_{k=1}^{m} \lambda_k u_k v_k^T, \quad m = \min(n, p), \tag{16}$$

where $u_1, \ldots, u_m$ are mutually orthogonal n vectors, $v_1, \ldots, v_m$ are mutually orthogonal p vectors, and $\lambda_k \geq 0$ are the singular values. This is similar to a plaid model with $\theta_{ijk} = \lambda_k$ and $(\rho_{1k}, \ldots, \rho_{nk}) = u_k$, and $(\kappa_{1k}, \ldots, \kappa_{pk}) = v_k$. Alter, Brown and Botstein (2000) apply the singular value decomposition directly to microarray data.

The plaid model differs in that the vectors from different layers are not constrained to be orthogonal. In plaid models, ρ_{ik} and κ_{jk} are constrained to take values in $\{0, 1\}$. More complicated plaid models with α_{ik} and β_{jk} terms can not be written as differently constrained SVDs.

8.2. Semidiscrete decomposition

The semidiscrete decomposition (SDD) takes the form (16) except that the elements of u_k and v_k belong to the set $\{-1, 0, 1\}$. Kolda and O'Leary (1998) report that the SDD provides faster and more space efficient information retrieval (IR) than the SVD. In IR applications, row i represents a term (such as a word), column j represents a document (such as an article or web page), and the raw data value X_{ij} contains the number of times that term i appears in document j. Typically a transformation is applied to the X_{ij} values before fitting an SVD or an SDD. The algorithm for estimating the SDD alternates between updating λ_k values and u_k and v_k values. That algorithm keeps $\lambda_k \in \{-1, 0, 1\}$ at all stages.

8.3. Non-negative matrix factorization

Lee and Seung (1999) describe a non-negative matrix factorization. It can be written as in equation (16) with all $\lambda_k = 1$, and with all elements of u_k and v_k constrained to be non-negative. They illustrate the decomposition on images and on information retrieval problems. Once again, an alternating iterative algorithm is used. Their preferred algorithm is based on a Poisson likelihood, though they also describe one based on sums of squares.

8.4. Clustering

As Lee and Seung (1999) point out, k-means style clustering of data rows (or of columns) can be cast in the form (16). To cluster the rows of Y take $\lambda_k = 1$ and constrain each u_k to have one element equal to 1 and all the others 0. Then v_k are the cluster centers.

Additive clustering (Shepard and Arabie (1979)) is a method for describing similarities among a group of observations. Here $n = p$, the rows and columns of Y describe the same set of objects, and the value Y_{ij} is a number describing the similarity between objects i and j. Additive clustering fits the model (16) with the constraints $u_{ik} = v_{ik} \in \{0, 1\}$.

Additive clustering and plaid models both allow clusters to overlap. Arabie and Hubert (1996) survey the literature on overlapping clusters.

The box clustering method is described by Mirkin (1998). Like additive clustering, box clustering takes the form (16) with u_{ik} and v_{jk} both in $\{0, 1\}$. But box clustering does not require that the rows and columns correspond to the same set of entities. Box clustering was developed for nonnegative entries interpretable as scaled probabilities, and the algorithms for it use certain naturally arising weighted sums of squares as criteria.

Eisen et al. (1998) cluster the rows and columns of their data using hierarchical clustering. Tibshirani, Hastie, Eisen, Ross, Botstein and Brown (1999) introduce gene shaving. In gene shaving a cluster is formed around the largest principal component of the data. Structure corresponding to such a cluster is then removed, and a new cluster forms around the principal component of the remainder. In gene shaving the model takes the form (16) with $\lambda_k = 1$, $u_{ik} \in \{-1, 0, 1\}$ and the principal component vector elements v_{jk} unconstrained.

Hofmann, Puzicha and Jordan (1999) propose a two-sided clustering model for dyadic (co-occurence) data. This data takes the form $Y_{ij} \in \{0, 1\}$, with for example, a 1 representing that person i has seen movie j. They describe clusters through unobserved latent class variables, and employ an EM algorithm to estimate their model. Hartigan (1972) is an early reference on two-sided non-overlapping clustering.

Getz, Levine and Domany (2000) perform a recursive partitioning of the data matrix, iterating between one-way clustering of the samples and one-way clustering of the genes. At each step, either genes or samples are clustered separately within each partition defined by the current sample and gene clusters. This leads to a rectangular representation of the data. Biologically significant clusters are identified as a certain subset of these rectangles, by keeping track of the parental clusters leading to them.

8.5. Further references

Here we mention additional related work that does not fit into the SVD framework above. Banfield and Raftery (1993) introduce model-based clustering that, like plaid, uses a global numerical goodness measure. They use Gaussian and other likelihoods. Ben-Dor, Shamir and Yakhini (1999) propose a clustering method for expression data based on an n by n gene similarity matrix.

Plaid is a merger of clustering and ANOVA methods. The sources mentioned above are versions of clustering and double clustering. ANOVA has also been applied to microarray data. Kerr and Churchill (2001) and Kerr, Martin and Churchill (2000) fit higher-order ANOVA models to microarray expression data. The data typically include known treatment or varietal types as variables and the effects of interest are the gene-variety interactions. Additional higher order terms account for array, spot and dye effects. In contrast, the plaid model attempts to find a clustering that can represent the gene-variety interactions by different layer-membership patterns for the genes. The use of higher order ANOVA models suggests natural extensions of the plaid model to the supervised context.

9. Discussion

The plaid model is a form of overlapping two-sided clustering, with an embedded ANOVA in each layer.

In this article we have presented an algorithm for finding plaid models. We have also proposed a sequential permutation strategy for protecting against the introduction of noisy terms in greedy model search. We expect that the algorithm could be improved. Our updating algorithms adjust each ρ_i and κ_j individually instead of jointly. This is driven by considerations of speed: $n + p$ ratios are much faster to compute than is an $n + p$ dimensional optimization. In some cases this simultaneous adjustment gives rise to null layers with all $\rho_i = 0$. We also acknowledge that other researchers might prefer to use a larger number of random permutations than we do.

Plaid models are exploratory tools, like cluster analysis. Just as in cluster analysis and many other multivariate methods, the results are sensitive to scaling of the data. If the columns of the yeast expression data are scaled to have a common variance, then the ribosomal protein layer is found first. Similarly when the food variables are not scaled the first layer only involves the calories per gram of food column, because that one has by far the largest variance. If the foods are not normalized by weight, then the first layer is dominated by the large foods, such as entire cakes, pies, loafs and half gallons of ice cream. These other clusterings are as real as the ones we report. When a choice is to be made among them, it must be based on a decision of what aspect of the data is of interest. Scaling issues for microarray data have been discussed by Yang, Dudoit, Luu and Speed (2001) and Tseng, Oh, Rohlin, Liao and Wong (2001).

We have also found that the results can change in response to changes in the algorithm. The features we present are ones that have been found more than once in repeated analyses of the data. This is consistent with advice given by Hartigan (1975) for clustering. For the food data, the first layer is invariably driven by lard, butter and oils, but the number of foods in this layer can depend on options of the algorithm. For the yeast data, the mirror imaged sporulation layers can be the first and second ones after the background, or they can be the first and fourth, but they always seem to be in the model somewhere. Changing parameters like the row release threshold, or making small changes in the data can change some of the resulting binary quantities ρ_{ik} and κ_{jk} in ways that are quite different from what we see in problems without binary parameters: sometimes no binary variables change at all; other times, a large change can appear. This problem could be mitigated by working with quantities in $[0, 1]$, but elements of $\{0, 1\}$ are more interpretable.

In our examples, we have found interpretable structure in genetics data, foreign exchange data, and nutrition data. These structures are clearly not noise artifacts.

Acknowledgements

We thank Chiara Sabatti, David Siegmund, Stephen Boyd, David Botstein, Patrick Brown and Trevor Hastie for helpful discussions. We thank Jorge Picazo for getting the Bloomberg data, as well as Rob Tibshirani, Asbjorn Berge, and an anonymous reviewer, for bringing additional references to our attention. We thank two anonymous reviewers for helpful comments. This work was supported by NIH grants CA59039 and GM53275, and by NSF grant DMS-9704495.

References

Alter, O., Brown, P. O. and Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences* **97**, 10101-10106.

Arabie, P. and Hubert, L. J. (1996). An overview of combinatorial data analysis. In *Clustering and Classification* (Edited by P. Arabie, L. J. Hubert and G. D. Soete) 6-64. World Scientific Publishers, River Edge, NJ,.

Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-gaussian clustering. *Biometrics* **49**, 803-821.

Ben-Dor, A., Shamir, R. and Yakhini, Z. (1999). Clustering gene expression patterns. *J. Comput. Biology* **6**, 281-297.

Bertin, J. (1983). *Semiology of Graphics.* University of Wisconsin Press, London.

Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P. and Herskowitz, I. (1998). The transcriptional program of sporulation in budding yeast. *Science* **282**, 699-705.

DeRisi, J., Iyer, V. and Brown, P. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* **278**, 680-686.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika* **1**, 211-218.

Eisen, M. B., Spellman, P. T., Brown, P. O. and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Science* **95**, 14863-14868.

Getz, G., Levine, E. and Domany, E. (2000). Coupled two-way clustering analysis of gene microarray data. *Proceedings of the National Academy of Science* **97**, 12079-12084.

Gordon, A. D. (1996). Hierarchical classification. In *Clustering and Classification* (Edited by P. Arabie, L. J. Hubert and G. D. Soete), 65-121. World Scientific Publishers, River Edge, NJ.

Hansen, P. and Jaumard, B. (1997). Cluster analysis and mathematical programming. *Math. Programming* **79**, 191-215.

Hartigan, J. (1975). *Clustering Algorithms.* Wiley, New York.

Hartigan, J. A. (1972). Direct clustering of a data matrix. *J. Amer. Statist. Assoc.* **67**, 123-129.

Hofmann, T., Puzicha, J. and Jordan, M. I. (1999). Learning from dyadic data. In Advances in neural information processing systems 11, NIPS 98. MIT Press.

Kerr, M. K. and Churchill, G. A. (2001). Statistical design and the analysis of gene expression microarrays. *Generial Research* **77**, 123-128.

Kerr, M. K., Martin, M. and Churchill, G. A. (2000). Analysis of variance for gene expression microarray data. *J. Comput. Biology* **7**, 819-837.

Kolda, T. G. and O'Leary, D. P. (1998). A semidiscrete matrix decomposition for latent semantic indexing in information retrieval. *ACM Transactions on Information Systems* **16**, 322-346.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature* **401**, 788-791.

Mirkin, B. (1998). Least-squares structuring, clustering, and data processing issues. *Comput. J.* **41**, 518-536.

Needham, R. (1965). Automatic classification: models and problems. In Mathematics and computer science in biology and medicine. H. M. Stationery Office.

Shepard, R. N. and Arabie, P. (1979). Additive clustering: representation of similarities as combinations of discrete overlapping properties. *Psychological Rev.* **86**, 87-123.

Spellman, P. T., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D. and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell* **9**, 3273-3297.

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

Tseng, G. C., Oh, M.-K., Rohlin, L., Liao, J. C. and Wong, W. H. (2001). Issues in cdna microarray analysis: quality filtering, channel normalization, models of variation and assessment of gene effects. *Nucleic Acids Research* **29**, 2549-2577.

Wolsey, L. A. (1998). *Integer Programming.* Wiley-Interscience, New York.

Yang, Y. H., Dudoit, S., Luu, P. and Speed, T. P. (2001). Normalization for cdna microarray data. In SPIE BiOS 2001.

Department of Health Research and Policy, Division of Biostatistics, Stanford University School of Medicine, HRP Redwood Building, Stanford, CA 94305-5405, U.S.A.

E-mail: lazzeroni@stanford.edu

Statistics Department, Stanford University, Sequoia Hall, 370 Serra Mall, Stanford, CA 94305-9510, U.S.A.

E-mail: art@stat.stanford.edu

(Received March 2001; accepted October 2001)