Clustering Using Objective Functions and Stochastic Search

James G. Booth
Department of Biological Statistics
& Computational Biology
Cornell University
Ithaca, NY 14853, USA
jb383@cornell.edu

George Casella
Department of Statistics
University of Florida
Gainesville, FL 32611 USA
casella@stat.ufl.edu

James P. Hobert
Department of Statistics
University of Florida
Gainesville, FL 32611 USA
jhobert@stat.ufl.edu

October 2005

Abstract

A new approach to clustering multivariate data, based on a multi-level linear mixed model, is proposed. A key feature of the model is that observations from the same cluster are correlated, because they share cluster specific random effects. The inclusion of such random effects allows parsimonious deviation of the mean profile, for a given cluster, from a given base model, that may be captured statistically via the posterior expectation, or best linear unbiased predictor. One of the parameters in the model is the true, underlying partition of the data, and the posterior distribution of this parameter, which is known up to a normalizing constant, is used to cluster the data. The problem of finding good partitions is not amenable to deterministic methods such as the EM algorithm. Thus, we propose a simple, Metropolis-Hastings Markov chain stochastic search, using a biased random walk to select candidate moves. The proposed methodology is fundamentally different from the well-known finite mixture model approach to clustering, which requires an independent and identically distributed structure, and does not explicitly include the partition as a parameter.

Keywords: Bayesian model, Best linear unbiased predictor, Cluster analysis, Linear mixed model, Markov chain Monte Carlo, Metropolis-Hastings algorithm, Microarray, Quadratic penalized splines, Set partition, Yeast cell cycle.

1 Introduction

Clustering and classification are some of the most fundamental data analysis tools in use today. Many standard clustering algorithms are based on the assumption that the measurements to be clustered are realizations of random vectors from some parametric statistical model. These models usually place no restriction on the mean structure via covariates or otherwise. However, in many applications there is potential for parsimonious representation of the mean. For example, microarray experiments often yield time series-type data where each p-dimensional vector consists of measurements at p different time points. In such cases, it seems natural to model the mean via regression, especially when tempered with the ability to detect clusters that are well defined but deviate from a specified parametric form. Recent related literature includes Serban and Wasserman (2005), Hitchcock et al. (2005), McLachlan et al. (2004) and Celeux et al. (2005). However, as we shall explain, there are some fundamental differences between existing approaches and those proposed in this article.

As a motivating example, we consider the expression profiles of yeast genes from the alpha-factor synchronization experiment discussed by Spellman et al. (1998). We look at a portion of these data consisting of the log expression ratios of 104 genes, which are known to be cell cycle-regulated, recorded at 18 equally-spaced time points. (Actually, there is some missing data that is described later in the paper.) The data are available online from Stanford University's Yeast Cell Cycle Analysis Project web site at

`http://genome-www.stanford.edu/cellcycle/data/rawdata/`.

The primary goal of cluster analysis in this context is to find groups of genes that are all part of a team performing some function; that is, groups of coregulated genes. An example of such a group is provided in Figure 1 which shows the profiles of a subset of eight histones that are known to show peak expression during the short synthesis phase. A naive way to cluster these data is to ignore the time aspect and simply apply a standard clustering algorithm to the 104 18-dimensional vectors. However, the arguments above

suggest a reason to model the mean as a function of time. Indeed, implicit in the analysis of Spellman et al. (1998) is the first-order Fourier series model

$$E\{y(t)\} = \frac{1}{2}a_0 + a_1 \cos(2\pi t/T) + b_1 \sin(2\pi t/T)\,, \tag{1}$$

where $y(t)$ denotes log expression ratio at time t, and T is the period of the cell-cycle. While it is tempting to make use of this model to parsimoniously represent the mean structure, one must recognize that strict adherence to such a model could cause problems. For example, the least squares fit of (1) to the eight histone profiles is overlayed in Figure 1. Note that the model is reasonably effective in identifying the phase of peak expression of each gene, but there is clearly a substantial lack-of-fit to these gene profiles. Indeed, if clusters of genes were ranked according to the fit of the Fourier series model, this extremely well-defined cluster might not fare too well. One of the key features of the mixed model methodology that we propose herein is the allowance for parsimonious deviation from an overly simplistic base model.

1.1 Background

The basic clustering problem is simple to state. Given a set of n distinguishable objects, we wish to distribute the objects into groups or clusters in such a way that the objects within each group are similar while the groups themselves are different. Let the integers $\mathbb{N}_n := \{1, 2, \ldots, n\}$ serve as labels for our n distinguishable objects. In mathematical terms, the output of a clustering algorithm is a partition of $\mathbb{N}_n$; that is, an unordered collection of non-empty subsets of $\mathbb{N}_n$. Unfortunately, it is not always clear exactly how to quantify the similarity of objects within clusters nor the difference between clusters. However, suppose that, in addition to the n objects, there is an objective function $\pi : \mathbb{P}_n \to \mathbb{R}^+$, where $\mathbb{P}_n$ denotes the set of all possible partitions of $\mathbb{N}_n$, which assigns a score to each partition reflecting the extent to which it achieves the overall clustering goal described above. In this case, the cluster analysis is tantamount to the straightforward optimization problem of finding the partition with the highest score. In this article we propose a general method

for constructing objective functions, through the use of linear mixed models, that take into account covariate information.

Suppose the objects to be clustered are n p-dimensional vectors denoted by $Y_i = (Y_{i1}, \ldots, Y_{ip})^T$, $i = 1, 2, \ldots, n$. The standard, model-based approach to clustering (see, e.g., McLachlan and Basford, 1988; McLachlan and Peel, 2000) begins with the assumption that these n vectors are realizations of n independent and identically distributed (iid) random vectors from the K-component mixture density

$$\sum_{k=1}^{K} \tau_k \, f(\cdot; \theta_k) \ , \tag{2}$$

where K is a fixed positive integer in $\mathbb{N}_n$, $\tau_k \in (0, 1)$, $\sum_{k=1}^{K} \tau_k = 1$, $\{f(\cdot\,; \theta) : \theta \in \Theta\}$ is a parametric family of densities on $\mathbb{R}^p$ and $\theta_k \in \Theta$ is an unknown vector of parameters associated with the kth component. A partition of the data is typically obtained as a byproduct of an EM algorithm designed to find the maximum likelihood (ML) estimates of the parameters, $\left\{(\tau_k)_{k=1}^{K}, (\theta_k)_{k=1}^{K}\right\}$. This EM algorithm is based on the following augmented version of the model. Let (W_i, Y_i) be iid pairs where each W_i is a K-dimensional multinomial random vector based on a single trial and probabilities $(\tau_1, \ldots, \tau_K)$, and, conditional on W_i having its 1 in the kth position, $Y_i \sim f(\cdot; \theta_k)$. Marginally, $Y_1, \ldots, Y_n$ are still iid with density (2). Not only do the latent W_i serve as the missing data upon which the EM algorithm is based, they also provide a construct to serve as an explicit statistical target for clustering. Specifically, the E-step consists of estimating the conditional expectation of the W_i's given the data. The key to the success of the EM algorithm in the mixture model context is the fact that the estimates have the simple form,

$$\widetilde{W}_{ik} = \frac{n_k f(y_i; \hat{\theta}_k)}{\sum_{r=1}^{K} n_r f(y_i; \hat{\theta}_r)} \ , \qquad k = 1, \ldots, K \ ,$$

where n_k is the number of objects in cluster k. It is important to recognize, however, that this formula is a consequence of the iid sampling model. The conditional expectations are not 0-1 vectors, but the components are positive and sum to one. A partition of the data is obtained through the "maximum likelihood classification rule," which assigns observation

i to the mixture component corresponding to the largest entry of $\widetilde{W}_i$ after the final iteration of the EM algorithm.

This method of cluster analysis is fast and convenient and hence popular in practice, but there are problems. Specifically, the statistical procedure and the optimization algorithm are fused together in an unnatural way in that the EM algorithm itself is a necessary part of the statistical inference procedure. To see this, note that mere knowledge of the ML estimates, or even the true values, of $(\tau_k)_{k=1}^K$ and $(\theta_k)_{k=1}^K$ is not sufficient for cluster analysis. Indeed, even if one had the true values of the parameters, one iteration of the EM algorithm would have to be run to cluster the data.

The motivation for using (2) as the basis for cluster analysis must surely be the fact that this model would be correct if the data were actually a random sample from a heterogeneous population with K groups whose sizes are proportional to the τ_i. However, in many applications of cluster analysis, including the genetics examples that we study, this sampling scheme is quite unrealistic. In fact, a more realistic assumption is that there is some fixed, unknown partition of $\mathbb{N}_n$, ω, that has $c = c(\omega)$ clusters denoted by $\mathcal{C}_1, \ldots, \mathcal{C}_c$ and that the data are a realization from a density of the form

$$f(y|\theta, \omega) = \prod_{k=1}^{c(\omega)} \prod_{j \in \mathcal{C}_k} f(y_j|\theta_k) \ . \tag{3}$$

Of course, $\cup_{k=1}^{c(\omega)} \mathcal{C}_k = \mathbb{N}_n$ and $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ whenever $i \neq j$. Note that, unlike the mixture model, (3) contains a parameter, ω, that is directly relevant to the basic clustering problem. The absence of such a parameter in the mixture model leads to an awkward role for the missing data in standard model-based cluster analysis.

Our objective function is based upon a generalization of (3) that takes into account covariate information and allows for dependence among data vectors in the same cluster. To be specific, the objective function is the posterior distribution, $\pi(\omega|y_1, \ldots, y_n)$, which is constructed by placing priors on all the parameters in the model and marginalizing over all parameters except ω. This general approach was suggested several decades ago in Binder (1978). However, stochastic search methods for finding partitions with high posterior prob-

ability were not feasible at that time. Also, the linear mixed model formulation proposed in this article is quite different from that proposed elsewhere in the literature.

1.2 Example

The basic ideas behind our method can be effectively conveyed by considering the simplest possible case where the observations are univariate; i.e., where $p = 1$. Here it is possible to visually cluster the data by looking for breaks in a plot of the ordered sequence of measurements. We will illustrate this using the well-known galaxy data, which were introduced to the statistical community by Roeder (1990). These data, which are shown in Figure 2, consist of the speeds of 82 galaxies and are often modeled as realizations of 82 iid random variables from a finite Gaussian mixture distribution. Alternatively, we postulate a mixed model based on an unknown, pre-existing partition of the 82 galaxies. Denote this partition by ω and let $\mathcal{C}_1, \ldots, \mathcal{C}_c$ denote its $c = c(\omega)$ clusters. Let $\mathcal{C}_k = \{m_{k,1}, \ldots, m_{k,n_k}\}$ for $k = 1, \ldots, c$ where $n_k = \#(\mathcal{C}_k)$ denotes the cardinality of $\mathcal{C}_k$. Assume that the speeds of galaxies in different clusters are independent, but the speeds of the galaxies from the same cluster are correlated. In particular, assume that for $k = 1, \ldots, c$ and $i = 1, \ldots, n_k$ we have

$$Y_{m_{k,i}} = \mu + V_k + \varepsilon_{m_{k,i}} \,, \tag{4}$$

where $V_1, \ldots, V_c$ are iid $N(0, \lambda\sigma^2)$, the εs are iid $N(0, \sigma^2)$ and independent of the V_ks. The V_ks are cluster-specific, random effects and λ is a tuning parameter whose value is assumed known. Using the priors described later in the paper, the posterior probability of a partition, ω, reduces to

$$\pi(\omega|y) \propto \pi^*(\omega|y) := \frac{\prod_{k=1}^{c(\omega)} n_k! \, (1 - w_k\lambda)^{1/2}}{\hat{\sigma}^n \left(\sum_{k=1}^{c(\omega)} w_k\right)^{1/2}} \,,$$

where $w_k = n_k/(1 + n_k\lambda)$, and

$$\hat{\sigma}^2 = \frac{1}{n} \left\{ \sum_{k=1}^{c(\omega)} \sum_{i \in \mathcal{C}_k} (Y_i - \bar{Y}_k)^2 + \sum_{k=1}^{c(\omega)} w_k (\bar{Y}_k - \hat{\mu})^2 \right\} \,, \tag{5}$$

where $\bar{Y}_k$ is the mean of the observations in the kth cluster and $\hat{\mu} = \sum_k w_k \bar{Y}_k / \sum_k w_k$. As an objective function for clustering, $\pi(\omega|y)$ has some appealing features. For example, it rewards partitions containing large homogeneous clusters because of the $n_k!$ in the numerator. We note however that, using priors from the family (7), this feature can be "tuned" to either further penalize or reward partitions for their complexity in terms of the number of clusters they contain.

Also, the two terms in (5) show that galaxies will tend to be clustered together if their velocities are similar to one another. Notice that the second term is given more weight if the (between clusters) effects variance is small.

We now consider searching for the best partition; that is, for $\arg\max_{\omega \in \mathbb{P}_{82}} \pi(\omega|y)$. Without loss of generality, assume that the observations are ordered from smallest to largest; i.e., $y_1 < y_2 < \cdots < y_{82}$. Because the data are univariate, it's easy to narrow down $\mathbb{P}_{82}$ to a much smaller set of partitions that warrant consideration. For example, it seems reasonable to restrict attention to those partitions whose clusters consist of consecutive integers. Unfortunately, there are 2^{81} such partitions and this is far too many to allow for an exhaustive search. We can narrow the field a bit more by insisting that galaxies with similar speeds be placed in the same cluster. In Figure 2, we have circled clusters of galaxies that are separated from their neighbors by at least 500 km/s. Vertical bars on the plot indicate additional gaps of at least 300 km/s. The total number of partitions that can be constructed using only the 12 gaps of 300 or more km/s is $2^{12} = 4096$, which is manageable.

To obtain a working value for the parameter λ, we fit the model (4) to a partition obtained by K-means clustering (Hartigan and Wong, 1979) with 5 clusters. This resulted in the estimate, $\hat{\lambda} = 66.7$, based on a partition with breaks between galaxy pairs (7,8), (24,25), (50,51) and (79,80). Table 1 gives the 25 partitions with the highest posterior probability calculated using $\hat{\lambda}$ out of the 4096 partitions described above. One interesting feature is that all except the 24th partition involve the same first three gap positions.

1.3 Optimizing the Objective Function

Unfortunately, in realistic problems involving multivariate observations it will not be possible to narrow down the set of viable partitions by inspection as was done in the univariate example above. In such problems, finding the partitions that yield the highest values of the objective function is a challenging optimization problem. The reason is that the total number of partitions of $\mathbb{N}_n$, $B_n = \#(\mathbb{P}_n)$, called the *Bell number* (Stanley, 1997, p.33), grows extremely rapidly with n; e.g., $B_{40} = 1.6 \times 10^{35}$ and $B_{100} = 4.8 \times 10^{115}$. Thus, even for moderately large n, it is computationally infeasible to enumerate $\mathbb{P}_n$. Not surprisingly, standard clustering algorithms typically fail to globally optimize any objective function.

A second contribution of this article is the development of a stochastic search algorithm for finding the maximizer of the objective function. The basic idea is to construct a Metropolis-Hastings (MH) Markov chain whose stationary distribution is proportional to the objective function. Of course, the key to success with the MH algorithm is the choice of the candidate transition kernel. We use a candidate based on a biased random walk over the partition space. This method is particularly simple to program and worked well in our examples.

For a simple illustration, consider the galaxy data. Note that two of the boundaries selected by the K-means procedure are not among our 12 choices. Hence, it is not too surprising that the (log) posterior probability of the K-means partition, -400.40, is not particularly high. However, there are in fact good partitions outside of our somewhat ad hoc set of 4096. For example, using the biased random walk stochastic search technique, described in § 3, we found the partition with boundaries between the pairs, (7,8), (9,10), (46,47), (76,77) and (79,80). This partition has an objective function value of -380.16, higher than any among the set of 4096.

In general, partitioning the data by finding the maximizer of an objective function alleviates several well-known difficulties associated with the standard clustering procedure. For example, one practical problem with the standard, EM-based procedure for the mixture model described above is that K must be fixed a priori. Fraley and Raftery (2002) suggest

optimizing the BIC criterion to solve this problem, but this means that the EM algorithm must be run once for every possible value of K that the user wishes to consider. While this may not be overly burdensome from a computational standpoint, it is not very satisfying. In contrast, our objective function can (at least in principle) be evaluated at every possible partition, and hence the fixed K problem is a non-issue. Another limitation of (2) is the independence assumption. Our model allows for the data vectors within a cluster to be correlated, which allows for parsimonious representation of the cluster means through the use of penalized splines.

One might argue that the methods proposed in this paper are computationally burdensome relative to more conventional clustering algorithms because of the requirement of a stochastic search. However, it is well known that methods such as K-means, and the mixture model-based approach, are sensitive to starting values. For example, the K-means algorithm can converge to substantially different solutions when re-run with a different random number generator seed. Different solutions must be evaluated using an objective function, such as a least squares criterion (in the case of K-means), or BIC (in the case of the mixture model). Since an exhaustive deterministic search is not practical, confidence in the solutions provided by these algorithms necessitates the use of some form of stochastic search. This fact has been recognized by other authors. For example, Selim and Alsutan (1991) attempt to minimize the K-means least squares criterion using a simulated annealing algorithm, and Celeux and Govaert (1992) propose two stochastic clustering methods based on the EM algorithm. The approach described in this paper can be thought of as a formalization of this process which leads to probability-based criteria for selecting good partitions based on a flexible class of statistical models.

1.4 Outline of Paper

The remainder of this article is organized as follows. In Section 2, we describe the mixed model framework that leads to a probability-based objective function for cluster analysis. A stochastic search procedure for maximizing the objective function is discussed in Section 3.

In Section 4 we apply the proposed method to the yeast cell cycle data described earlier in this section, and to data from an experiment on corneal wound healing in rats. In the latter example, there is no obvious base model for the cluster means. Thus, we propose the use of quadratic penalized splines as a parsimonious but flexible class of curves to describe cluster mean profiles. Since these functions may be estimated as best linear unbiased predictors obtained from appropriately defined mixed models (Ruppert et al., 2003), the proposed methodology can be applied in contexts where no a priori model is available. We conclude in Section 5 with some discussion, including possible alternative search procedures. Some of the calculations underlying the derivation of our objective function are provided in the Appendix.

2 Model-based Objective Functions

Suppose that the data vector, Y_i, measured on the ith object actually consists of r replicate profiles; that is,

$$Y_i = \left(Y_{i11}, \ldots, Y_{i1p}, \ldots, Y_{ir1}, \ldots, Y_{irp}\right)^T = (Y_{i1}^T, \ldots, Y_{ir}^T)^T \ ,$$

for $i = 1, \ldots, n$. A particular setting where this data structure arises is microarray experiments in which replicate measurements are made on each gene. Of course, if there is no replication, we can drop the second subscript. We now describe a model for the data vectors, $Y_1, \ldots, Y_n$. Fix $\omega \in \mathbb{P}_n$ and let $\theta = (\theta_k)_{k=1}^{c(\omega)}$ denote a set of cluster specific parameter vectors where $\theta_k \in \Theta$. We assume that, given (ω, θ), the data vectors are partitioned into c clusters according to ω and that the clusters of data are mutually independent. However, the random vectors within each cluster may be correlated and the joint distribution depends on the value of the corresponding θ_k. In the most general case, we suppose that dependence among the Y_i within a cluster, and among replicate profiles from the same object, is induced by cluster and object specific random effects. To be specific, for $l \in \{1, 2\}$, let $\{g_l(\cdot|\theta) : \theta \in \Theta\}$ denote a parametric family of densities, each having support $S_l \subset \mathbb{R}^{s_l}$, and let $\{h(\cdot|u, v, \theta) : u \in S_1, v \in S_2, \theta \in \Theta\}$ denote another family with common sup-

port that is a subset of $\mathbb{R}^p$. Then, for a given fixed value of (ω, θ), the joint density of $Y = (Y_1^T, \ldots, Y_n^T)^T$ is given by

$$f(y|\theta, \omega) = \prod_{k=1}^{c(\omega)} \int_{S_2} \left[\prod_{i \in \mathcal{C}_k} \int_{S_1} \left\{ \prod_{j=1}^{r} h(y_{ij}|u_i, v_k, \theta_k) \right\} g_1(u_i|\theta_k)\, du_i \right] g_2(v_k|\theta_k)\, dv_k \ . \quad (6)$$

The density h may depend on known covariates, but this is suppressed notationally. This model is similar in structure to the "parametric partition models" used by Hartigan (1990) and Crowley (1997) and also to a model used by Consonni and Veronese (1995). However, in those models, there is within cluster independence given (ω, θ). Furthermore, these authors were not specifically concerned with cluster analysis.

After observing the data profiles, the density in (6) is referred to as the "classification likelihood" for the full parameter vector (ω, θ) (see e.g. Banfield and Raftery, 1993). One approach to clustering is to attempt to find values $(\hat{\omega}, \hat{\theta})$, which jointly maximize (6). In contrast, we propose a Bayesian approach in which the nuisance parameter θ is integrated out after multiplication be a suitable prior distribution. Since the dimension of θ depends on ω, it is natural to use a hierarchical prior of the form $\pi(\theta|\omega)\, \pi(\omega)$ (see Green, 1995). Several authors have suggested default prior distributions on $\mathbb{P}_n$. Arguing that the prior aught to promote parsimony by giving more prior weight to partitions with a small number of clusters, Consonni and Veronese (1995) suggest

$$\pi(\omega) \propto \frac{c(\omega)^{-1}}{S(n, c(\omega))} \ ,$$

where $S(n, k)$ is a Stirling number of the second kind; i.e, the number of partitions of n objects that have exactly k clusters. Hence, aside from a normalizing constant, this prior gives total weight c^{-1} to the set of partitions with c clusters, and gives equal weight to all partitions with the same number of clusters. Crowley (1997) considers the family of priors,

$$\pi(\omega) \propto m^{c(\omega)} \prod_{k=1}^{c(\omega)} (n_k - 1)! \ , \quad (7)$$

where, again, $n_k = \#\big(\mathcal{C}_k\big)$ and $m > 0$ is a parameter. Heuristic arguments based on predictive likelihood led us to the prior $\pi(\omega) \propto \prod_{k=1}^{c(\omega)} n_k!$, which is similar to Crowley's

prior with $m = 1$. Clearly, choosing $m < 1$ encourages partitions with a small number of clusters, whereas $m > 1$ does the opposite. Both choices may be desirable depending on the application. As for $\pi(\theta|\omega)$, we assume that conditional on ω, the random vectors $\theta_1, \ldots, \theta_c$ are exchangeable, but the precise form will depend on the specific structure of the model.

Under this formulation of the clustering problem, it is clear that prediction of ω is the key to identifying the clusters. Furthermore, θ is a nuisance parameter with respect to the cluster assignments. All of this suggests that cluster analysis should be based on the marginal posterior of ω given by

$$\pi(\omega|y) \propto \int f(y|\theta, \omega)\pi(\theta|\omega)\, \pi(\omega)\, d\theta \,. \tag{8}$$

We propose using this marginal posterior as an objective function for cluster analysis.

In this paper, we focus on a particular version of (6) in which the joint distribution of the response vectors in $\mathcal{C}_k$, given (θ, ω), is described by a linear mixed model. In order to avoid excessive subscripting, assume for the time being that $\mathcal{C}_k = \{1, \ldots, n_k\}$. We assume that the data vectors corresponding to objects in the kth cluster follow the model

$$Y_{ij} = X\beta_k + Z_1 U_i + Z_2 V_k + \varepsilon_{ij} \,, \tag{9}$$

where $i = 1, \ldots, n_k$, $j = 1, \ldots, r$, the ε_{ij} are iid $N_p(0, \sigma_k^2 I_p)$, the U_i are iid $\mathrm{N}_{s_1}(0, \lambda_1 \sigma_k^2 I_{s_1})$ and $V_k \sim \mathrm{N}_{s_2}(0, \lambda_2 \sigma_k^2 I_{s_2})$. We assume that the ε_{ij}, U_i and V_k are mutually independent. In terms of the general model, we have taken $g_l(\cdot; \theta_k)$ to be an s_l-variate normal density with zero mean and covariance matrix $\lambda_l \sigma_k^2 I_{s_l}$, for $l \in \{1, 2\}$. The matrix X is $p \times q$ ($q < p$) with full column rank, β_k is a q-dimensional regression parameter, and the matrix Z_l is $p \times s_l$ with rank $s_l^* \leq s_l$. In this case, $\theta_k = (\beta_k, \sigma_k^2)$, and λ_1 and λ_2 are tuning parameters. We complete the specification of the normal-normal model by taking the prior $\pi(\beta, \sigma^2|\omega) \propto \prod_{k=1}^{c(\omega)} (1/\sigma_k^2)^{\alpha+1}$. We now work out the exact form of (8) under these specific assumptions.

Let Y_k^* denote the $n_k rp \times 1$ vector consisting of all the responses in $\mathcal{C}_k$ stacked on top of one another. Then, it is readily verified that $Y_k^* \sim N_{n_k rp}\left[(1_{n_k r} \otimes X)\beta_k, \sigma_k^2 M_k\right]$, where

$M_k = I_{n_k} \otimes A + J_{n_k} \otimes B$, 1_m is a 1-vector of length m, $J_m = 1_m 1_m^T$, and the matrices A and B are given by

$$A = I_r \otimes I_p + J_r \otimes \lambda_1 Z_1 Z_1^T \quad \text{and} \quad B = J_r \otimes \lambda_2 Z_2 Z_2^T \,.$$

Let $\bar{Y}_k$ represent the mean profile in the kth cluster; that is, the average of the $n_k r$ p-dimensional vectors that comprise Y_k^*. Also define

$$W_k = \left(I_p + r\lambda_1 Z_1 Z_1^T + n_k r \lambda_2 Z_2 Z_2^T \right)^{-1} \,.$$

We show in the Appendix that, for a given partition, the statistics

$$\hat{\beta}_k = (X^T W_k X)^{-1} X^T W_k \bar{Y}_k \,,$$

and

$$\hat{\sigma}_k^2 = \frac{1}{n_k r p} \sum_{i \in \mathcal{C}_k} \left(Y_i - 1_r \otimes \bar{Y}_k \right)^T A^{-1} \left(Y_i - 1_r \otimes \bar{Y}_k \right) + \frac{1}{p} \left(\bar{Y}_k - X\hat{\beta}_k \right)^T W_k \left(\bar{Y}_k - X\hat{\beta}_k \right) \,, \quad (10)$$

for $k = 1, \ldots, c(\omega)$, jointly maximize (6). Furthermore, the joint density of the measurements on the objects in cluster k can be written as

$$f(y_k^* | \theta, \omega) = \left| 2\pi\sigma_k^2 M_k \right|^{-1/2} \exp\left[-\frac{n_k r}{2\sigma_k^2} \left\{ (\beta_k - \hat{\beta}_k)^T X^T W_k X (\beta_k - \hat{\beta}_k) + p\hat{\sigma}_k^2 \right\} \right] \,. \quad (11)$$

Let $\delta_1, \ldots, \delta_p$ be the eigenvalues of the matrix $D = \left(I_p + r\lambda_1 Z_1 Z_1^T \right)^{-1} \lambda_2 Z_2 Z_2^T$ and note that

$$|M_k| = |A|^{n_k} \prod_{i=1}^{p} (1 + n_k r \delta_i) \,.$$

Now, integrating the product of (11) and $(1/\sigma_k^2)^{\alpha+1}$ with respect to β_k and σ_k^2, we obtain

$$\frac{2^\alpha \, \Gamma\left[\frac{n_k r p - q}{2} + \alpha \right]}{\pi^{(n_k r p - q)/2} \, (n_k r p \hat{\sigma}_k^2)^{(n_k r p - q)/2 + \alpha} \, |n_k r X' W_k X|^{1/2} \, |A|^{n_k/2} \, \prod_{i=1}^{p} (1 + n_k r \delta_i)^{1/2}} \,.$$

Finally, taking the product over the clusters, and multiplying by the prior $\pi(\omega)$ yields our proposed objective function for cluster analysis

$$\pi(\omega | y) \quad \propto \quad \pi(\omega) \prod_{k=1}^{c(\omega)} \frac{2^\alpha \, \pi^{q/2} \, \Gamma\left[\frac{n_k r p - q}{2} + \alpha \right] \prod_{i=1}^{p} (1 + n_k r \delta_i)^{-1/2}}{(n_k r)^{q/2} \left(n_k r p \hat{\sigma}_k^2 \right)^{(n_k r p - q)/2 + \alpha} \, |X^T W_k X|^{1/2}} \,. \quad (12)$$

Notice that, if each measurement is rescaled by a factor a, say, then the value of $\hat{\sigma}_k^2$ changes to $a^2\hat{\sigma}_k^2$, resulting in a multiplicative change in (12) of $\prod_{k=1}^{c} a^{-(n_k rp-q+2\alpha)} \propto a^{c(q-2\alpha)}$. This motivates the choice $q/2$ for the hyperparameter α on the grounds of scale invariance. This is of practical importance in microarray studies, for example, where the responses are log expression ratios, and the choice of base is arbitrary.

Once again, $\pi(\omega|y)$ turns out to be an intuitively reasonable objective function for clustering in that it rewards large homogeneous clusters; i.e., those in which $\hat{\sigma}_k^2$ is small and n_k is large, but also includes penalties for extreme partitions in which $c(\omega)$ is very small or very large. Moreover, examining the form of $\hat{\sigma}_k^2$ shows that there are two distinct parts to this variance. The first piece measures a within cluster variance, and will help to identify clusters of similar objects even if they do not follow the specified model. The second piece measures lack-of-fit, and will identify clusters of objects whose average profile closely follows the assumed model determined by the matrix X.

Another nice feature of the mixed model approach is that the predicted mean for a given cluster is a compromise between a projection on to the columns of X and the columns of $X|Z_2$. This allows cluster means to deviate from the base model while retaining parsimony. Due to the flat prior specification for β, provided $n_k rp - q > 1$ the posterior expectation of $X\beta_k + Z_2 V_k$ is equal to the best linear unbiased predictor given by

$$\begin{aligned} X\hat{\beta}_k + Z_2\hat{V}_k &= X\hat{\beta}_k + n_k r\lambda_2 Z_2 Z_2^T W_k(\bar{Y}_k - X\hat{\beta}_k) \\ &= n_k r\lambda_2 Z_2 Z_2^T W_k \bar{Y}_k + (I - n_k r\lambda_2 Z_2 Z_2^T W_k) X\hat{\beta}_k \ . \end{aligned} \tag{13}$$

Thus, predicted values in large clusters shrink towards the more complex model. In particular, if $Z_1 = 0$ and $Z_2 = I$, the predicted values in the kth cluster are a convex combination of $X\hat{\beta}_k$ and $\bar{Y}_k$, the estimate based on a completely unstructured mean.

A variation on the model is one in which the parameters are the same in all clusters; i.e., $\beta_k = \beta$ and $\sigma_k^2 = \sigma^2$ for all k. In this case, similar arguments lead to the posterior distribution,

$$\pi(\omega|y) \quad \propto \quad \frac{\pi(\omega)\prod_{i=1}^{p}(1+n_k r\delta_i)^{-1/2}}{(\hat{\sigma}^2)^{(nrp-q)/2+\alpha}\left|\sum_{k=1}^{c(\omega)} n_k r X^T W_k X\right|^{1/2}} \ , \tag{14}$$

where

$$\begin{aligned}
\hat{\beta} &= \left(\sum_{k=1}^{c(\omega)} n_k X^T W_k X\right)^{-1} \sum_{k=1}^{c(\omega)} n_k X^T W_k \bar{Y}_k \\
\hat{\sigma}^2 &= \frac{1}{nrp} \sum_{k=1}^{c(\omega)} (Y_k^* - 1_{n_k r} \otimes X\hat{\beta})^T M_k^{-1} (Y_k^* - 1_{n_k r} \otimes X\hat{\beta}) \\
&= \frac{1}{nrp} \left\{ \sum_{i=1}^{n} (Y_i - 1_r \otimes \bar{Y}_k)^T A^{-1} (Y_i - 1_r \otimes \bar{Y}_k) + \sum_{k=1}^{c(\omega)} n_k r (\bar{Y}_k - X\hat{\beta})^T W_k (\bar{Y}_k - X\hat{\beta}) \right\} .
\end{aligned}$$

Our objective function for clustering the galaxy data is a special case of (14).

3 Stochastic search

Once we have constructed an objective function, $\pi : \mathbb{P}_n \to \mathbb{R}^+$, that measures the goodness of partitions, we are left with a potentially difficult optimization problem. As explained in Section 1, B_n grows extremely rapidly with n. Therefore, unless n is very small, it is impossible to maximize π by brute force enumeration. An alternative approach might be to generate a completely random sample of partitions from $\mathbb{P}_n$ and to evaluate their π-values. Surprisingly, simulation from the uniform distribution on $\mathbb{P}_n$ is a non-trivial problem - see, e.g., Pitman (1997). Moreover, this method is quite inefficient. For example, even if n is only 20, and one billion random partitions are generated, the probability that none of the top 1000 partitions are observed is about 0.98. Thus, it is clear that a more intelligent search algorithm is required.

Our task is a special case of the general problem of finding the maximum of an objective function over a large combinatorial set. Problems of this type are often amenable to MCMC optimization, which entails running a Markov chain on the combinatorial set of interest and evaluating the objective function at the successive states (see, e.g., Section 12.6 of Jerrum and Sinclair, 1996). In the next subsection, we describe a MH Markov chain with state space $\mathbb{P}_n$ and stationary mass function proportional to π that is very simple to simulate and worked well in the examples we have considered.

3.1 A Metropolis-Hastings algorithm based on a biased random walk

In general, the MH algorithm allows one to simulate a Markov chain with a prespecified stationary distribution by "correcting" an easy to simulate candidate Markov chain. When MH is used to solve combinatorial optimization problems, the candidate Markov chain is often taken to be a random walk on a graph that defines a neighborhood structure for the combinatorial set in question. Let G_n be a connected, undirected graph with vertex set $\mathbb{P}_n$ such that there is an edge between two vertices, ω_i and ω_j, if and only if it is possible to get from partition ω_i to partition ω_j by moving exactly one of the n objects in ω_i to a different cluster. The graph G_n determines a neighborhood structure on $\mathbb{P}_n$; i.e., ω_i and ω_j are neighbors if and only if they share an edge. For example, when $n = 3$ the Bell number is 5 and the partitions are

$$\omega_1 : \{1,2,3\} \quad \omega_2 : \{1,2\}\{3\} \quad \omega_3 : \{1,3\}\{2\} \quad \omega_4 : \{2,3\}\{1\} \quad \omega_5 : \{1\}\{2\}\{3\} \ .$$

The only two partitions that do not share an edge in G_5 are ω_1 and ω_5.

Let $d(\omega)$ denote the number of neighbors (or degree) of the vertex ω in G_n. An obvious candidate Markov chain for the MH algorithm is the nearest neighbor random walk on G_n which moves from ω' to ω with probability $1/d(\omega')$ if ω and ω' are neighbors and zero otherwise. Our simple example with $n = 5$ illustrates that different partitions may have different numbers of neighbors. Consequently, the transition matrix of this random walk is not symmetric. For example, $\mathrm{pr}(\omega_1 \to \omega_2) = 1/3$, but $\mathrm{pr}(\omega_2 \to \omega_1) = 1/4$.

Consider programming the MH algorithm with the nearest neighbor random walk as the candidate. Let the current state be ω'. In order to simulate the next state, we require a method of sampling uniformly at random from the $d(\omega')$ neighbors of ω' - call the selected neighbor ω - and an algorithm for calculating $d(\omega)$, so that the acceptance probability may be computed. This can become quite computationally intensive, as we must enumerate all of the possible neighbors. However, it turns out that a slightly different candidate leads to a MH algorithm that is as effective and much simpler to program.

The alternative candidate Markov chain evolves as follows. Let c denote the number of

clusters in the current state (partition). There are two cases: $c = 1$ and $c \geq 2$. If $c = 1$, choose one of the n objects uniformly at random and move the chosen object to its own cluster. If $c \geq 2$, choose one of the n objects uniformly at random. If the chosen object is a singleton (i.e., forms its own cluster), then move it to one of the other $c - 1$ clusters, each with probability $1/(c-1)$. If the chosen object is not a singleton, then move it to one of the other $c - 1$ clusters, each with probability $1/c$, or make the chosen object its own cluster with probability $1/c$. As with the nearest neighbor random walk, the move $\omega' \to \omega$ has positive probability if and only if ω' and ω share an edge in G_n. We call this Markov chain the biased random walk on G_n.

At first glance, one might think that what we have just described is simply an algorithm for simulating the nearest neighbor random walk. This is not the case however. For example, under the new dynamics, in the $n = 5$ example, $\mathrm{pr}(\omega_1 \to \omega_2) = \mathrm{pr}(\omega_2 \to \omega_1) = 1/3$. In fact, straightforward arguments show that the transition matrix of the biased random walk is symmetric. Therefore, when this chain is used as the candidate in the MH algorithm, the acceptance probability is simply $\min\{1, \pi(\omega)/\pi(\omega')\}$. That is, running this algorithm does not require finding or counting the neighbors of ω and ω'. Hence, this alternative candidate results in a MH algorithm that is much easier to program and faster in the sense of more iterations per unit time. There are myriad other (random and nonrandom) algorithms that could be used to search for the maximum of π and a few of these are briefly described in the next subsection.

3.2 Alternative methods

The Gibbs sampler, which is an alternative MCMC algorithm, evolves as follows. Suppose the current state of the Markov chain is the partition $\omega \in \mathbb{P}_n$ and fix $i \in \mathbb{N}_n$. Let ω_{-i} denote ω with the ith object removed and define S to be the set of all partitions in $\mathbb{P}_n$ that, when the ith object is removed, are identical to ω_{-i}. The new state is drawn from S using probabilities proportional to the objective function. It is straightforward to show that this basic transition is reversible with respect to the (normalized) objective function. Of

course, i must be varied in order to guarantee irreducibility. In the "random-scan" Gibbs sampler, i is randomly selected from $\mathbb{N}_n$ at each iteration, while the "deterministic-scan" version simply cycles through the is in some predetermined order. (The algorithm used by Crowley (1997) is the deterministic scan Gibbs sampler, although she does not call it that.) Note that the random scan Gibbs sampler and our MH algorithm have the same basic structure. At each iteration, one of the n objects is chosen at random and the chosen object is either moved to a new cluster or left where it is. However, one iteration of the Gibbs sampler requires $c(\omega_{-i}) + 1$ evaluations of π while our MH algorithm requires only 2. Thus, it is possible to perform many more iterations of the MH algorithm per unit time.

The MH algorithm and the Gibbs sampler make small (or local) moves in the sense that only one object at a time is moved from one cluster to another. An alternative candidate for the MH algorithm that proposes more drastic changes (that are less likely to be accepted) is the "split-merge" candidate (Green, 1995; Jain and Neal, 2004). A Bernoulli random variable is first used to (randomly) decide between a merge move and a split move. A merge proposal is constructed by merging two randomly chosen clusters in the current partition. A split proposal is created by randomly choosing a cluster and then randomly splitting it into two clusters conditional on neither being empty. Green (1995) provides formulas for the acceptance probabilities. The split-merge candidate can either be used by itself or in conjunction with another candidate, such as the biased random walk candidate.

The simulated tempering algorithm (Geyer and Thompson, 1995; Marinari and Parisi, 1992) was designed to combat the difficulties that arise when the stationary distribution has isolated modes. The basic idea is to construct a sequence (or ladder) of distributions that gradually change from the distribution of interest to the uniform distribution (assuming a finite state space). The Markov chain underlying the algorithm has two components, which are the variable of interest (the partition in our case) and the "temperature" variable that dictates which of the distributions in the ladder will be sampled. When the temperature variable eventually becomes large, the more flat distributions are sampled, which allows the variable of interest to move with ease away from what might be a local mode under

the distributions at the other end of the ladder. One advantage of the simulated tempering algorithm over its predecessor, the simulated annealing algorithm (Kirkpatrick et al., 1983), is that its driving Markov chain is time homogeneous (Jerrum and Sinclair, 1996; Jerrum and Sorkin, 1998). A related algorithm that circumvents problems related to fine tuning the temperature ladder is Neal's (1996) tempered transitions algorithm.

The performance of some of these algorithms in optimizing our objective function was investigated in a number of examples; see the beginning of Section 5 for details. We found that the MH algorithm described in Section 3.1 was the best choice for us, not only because of the ease of implementation and programming, but also because it was dependable in finding a (nearly) optimal solution.

4 Examples

4.1 Yeast cell cycle

We begin by describing our analysis of the yeast cell cycle data described in Section 1. We focus on profiles obtained from 104 genes previously identified as being cell cycle-regulated. The profiles consist of log expression ratios taken from 18 cDNA microarrays equally spaced at 7 minute intervals. About 80% of the profiles are complete, and all except one had 4 or fewer missing values. The one gene with more than 4 missing values is omitted in the subsequent analysis.

We used the first-order Fourier series model (1) to fill in the missing values and to register the profiles. More specifically, note that if T is known, then the model is linear in the intercept and slope parameters, a_0 and (a_1, b_1). In our analysis we fixed T at 62, which is the least squares estimate of T obtained by Booth et al. (2004) in a previous analysis of these same data. We then estimated the regression parameters for each gene separately via least squares and used the resulting fitted models to fill in the missing data. While it is actually possible to perform the cluster analysis without first filling in the missing values, having balanced data greatly simplifies the computations in the cluster analysis. Indeed,

when the data are balanced, the estimated regression coefficients for a given cluster are simply averages of the least squares estimates for the genes in the cluster. As missing values comprised less than 2% of the data, this substitution has little impact on our conclusions. Finally, in order to register the profiles at the same overall level, we further modified the data by subtracting the estimated intercept from each profile. This step is similar to the mean subtraction employed by Spellman et al. (1998).

In our cluster analysis, we used the linear mixed model (9) with $Z_1 = 0$, $Z_2 = I$, and an X matrix based on (1); i.e., X is 18×2 with the row corresponding to time t equal to $(\cos(2\pi t/62), \sin(2\pi t/62))$. The intercept term is absent because of the registration, and the smoothing coefficient, λ_1, is zero in this application because there is no replication. To obtain a value for the coefficient, λ_2, we first applied the K-means clustering algorithm to the data with the number of clusters fixed at five, corresponding to the number of phases of the cell-cycle. We then fit a simplified linear mixed model with homogeneous error variance to the data grouped by K-means, and estimated λ_2 by the ratio of effects to error variances. This resulted in a value $\hat{\lambda}_2 = 1.63$. (A method for incorporating the tuning parameters into the Bayesian analysis is discussed in the next section.)

We searched for the maximizer of the objective function by running 10^5 iterations of the MH algorithm with the biased random walk candidate. Starting the algorithm at the K-means solution, the best overall partition found contained 36 clusters. Twenty of these were singletons with profiles that are distinctly different from any other gene. Of the remaining 16 clusters, six contained only two genes. The ten largest clusters are shown in Figure 3, with the histones discussed in the Introduction captured in cluster number 3.

An obvious concern is the extent to which the results of our search algorithm depend on the starting value. Based on our experimentation, it appears that even drastically different starting values lead to qualitatively very similar answers. For example, we ran our algorithm a second time starting with every gene in its own cluster and this resulted in a best overall partition with six of the largest ten clusters identical to those shown in Figure 3. The remaining four largest clusters each only differed by one gene. This run also identified

20 singletons, 17 of which were in agreement with the previous solution.

This insensitivity to the starting value is a property not shared by the K-means procedure. Indeed, the first partition we obtained by running the K-means procedure with 36 clusters was substantially different from the one shown in Figure 3 with only one cluster containing nine genes, and none other larger than six. In particular, the histone cluster was not identified. In order for the K-means procedure to be effective, it must involve some form of stochastic search, with good partitions being identified presumably by its least squares criterion. Similar comments can be made about other clustering algorithms that converge to different solutions depending upon the starting values.

4.2 Corneal wound healing

Our next example concerns 646 gene expression profiles obtained from Affymetrix gene chip microarrays at days 0, 1, 2, 3, 4, 5, 6, 7, 14, 21, 42, and 98 of a study of corneal wound healing in rats at the University of Florida. There were two technical replicate measurements at each timepoint. The day 0 sample was taken prior to photorefractive keratectomy (corrective eye surgery), and hence represents a baseline value to which the profiles are expected to return over the treatment period. Unlike the yeast cell cycle example, here there is no obvious parametric base model for the profiles. Hence, we consider a quadratic penalized spline model with knots at each interior timepoint. This can be formulated as the best linear unbiased predictor from a linear mixed model fit (Ruppert et al., 2003). To be specific, consider the equally-spaced and centered time-scale, $t_j = (j - 5.5)$, $j = 0, 1, \ldots, 11$. Then, the X-matrix in (9) is 12×3 with jth row given by $(1, t_j, t_j^2)$, and Z_2 is 12×10, with the entries in column $i = 1, \ldots, 10$ equal to $(t_j - t_i)_+^2$, $j = 0, \ldots, 11$. We allowed for correlation between replicate profiles using gene specific penalized splines with two interior knots at -2 and $+2$. Hence, differences between replicate profiles are attributable to two sources: a systematic deviation in shape and random noise.

As with the cell cycle data, an initial partition of the data was obtained by applying the K-means procedure. In this case we arbitrarily set the number of clusters equal to 20.

Then, to obtain values for the parameters λ_1 and λ_2 for use in a stochastic search, we fit the mixed model to the data grouped by K-means again using a homogeneous error variance across clusters. This resulted in the restricted ML estimates $\hat{\lambda}_1 = 0$, and $\hat{\lambda}_2 = 3.79$. Thus, there is considerable within cluster correlation, but correlation within genes appears to be negligible in this data set.

The best partition found after running the stochastic search algorithm for 10^5 iterations, using the K-means solution to initiate the Markov chain, consisted of 28 clusters. The twelve clusters with the highest range of expression values are displayed in Figure 4. Clusters of genes with large changes in expression levels over the time course may be of particular scientific interest and may warrant further investigation. None of the six largest clusters, which contain well over half of the genes are included in this set. These large clusters consist of genes with relatively constant expression levels, and are therefore of less scientific interest.

5 Discussion

In both of the examples of Section 4, we reported the best partitions found using the MH algorithm described in Subsection 3.1. We also programmed two of the alternative stochastic search methods discussed in Subsection 3.2. The first was a MH algorithm in which the candidate was a mixture of split/merge moves and biased random walk moves. The second was the tempered transitions algorithm of Neal (1996). These methods are much more complicated to program and involve tuning parameters which in practice must be chosen on a somewhat ad hoc basis. In neither case did we find any evidence of an improvement over the simple MH algorithm. An explanation in the split/merge case is that the bigger moves proposed by this method were rarely accepted in practice. On the other hand, the rationale behind the tempered transitions approach is to avoid getting stuck in isolated modes. However, we find it difficult to contemplate a practical situation in which two or more good partitions are isolated from one another in terms of the neighborhood structure used to de-

fine the biased random walk. Our conjecture is that the good partitions all lie in the same "cloud" with respect to this neighborhood structure topology.

In the examples of Section 4 the "tuning" parameter, $\lambda = (\lambda_1, \lambda_2)$, was fixed throughout the stochastic search. We chose its value by fitting the proposed linear mixed model to an initial partition obtained using the K-means procedure. An alternative approach is to incorporate λ into the Bayesian analysis by specifying its prior distribution. However, no choice of prior leads to a tractable form for the marginal posterior over the space of partitions. One possibility is to use a transition kernel that is a mixture of the form

$$\begin{aligned} Q\left\{(\omega, \lambda'); (\omega', \lambda')\right\} &= p \times q(\omega; \omega') \\ Q\left\{(\omega', \lambda); (\omega', \lambda')\right\} &= (1-p) \times s(\lambda; \lambda') \end{aligned}$$

where (ω', λ') is the current state. That is, at each iteration of the chain, a candidate partition is selected according to a biased random walk move with probability p, or a candidate value of λ is selected from a density, s, with probability $1-p$. We have successfully implemented this approach for λ_2 using the candidate density of the form, $s(\lambda_2, \lambda_2') = t[(\lambda_2 - \lambda_2')/\kappa]/\kappa$, where t denotes a Student's-t density. We chose scale parameter, κ, to match the fit of a Gaussian distribution to the posterior as a function of λ_2 at the initial partition. (The posterior $\pi(\omega|y)$ is integrable with respect to λ_2 if $Z_2 = I$.) An issue that arises with this approach is that computation of marginal posterior probabilities for a large number of partitions, based on the realized Markov chain, is not really feasible, since most partitions are visited very infrequently. Hence, good partitions must be selected based on the joint posterior density of (ω, λ).

In conclusion, we have proposed a multi-level mixed model for clustering multivariate data. Our model leads to a tractable, probability-based, objective function for identifying good partitions. One key difference between the proposed approach, and most conventional clustering algorithms, is that it is not necessary to specify the number of clusters in advance. A second difference is that measurements on different objects, within the same cluster, are correlated because they share cluster specific random effects. The inclusion of such random effects allows for parsimonious deviation of the mean profile for a given cluster

from a given base model, that may be captured statistically via the best linear unbiased predictor. We also allow for a second level of dependence when replicate observations are obtained on each object, a situation that is quite common in microarray experiments. This second type of dependence can also be incorporated into the mixture model framework (2) by letting $f(\cdot;\theta_k)$ be the density of the entire observation vector for an object from cluster k. For example, both McLachlan et al. (2004) and Celeux et al. (2005) propose mixed model formulations of the density f in which dependence between replicate observations is induced, as it is in our model, through the inclusion of object specific random effects. These models fit within the mixture model framework because the observation vectors from different objects in the same cluster are iid. It is also possible to modify the standard EM fitting algorithm to apply in this case. In contrast, the inclusion of cluster specific random effects is not possible within the mixture model framework.

Acknowledgment: The authors are grateful to Dr. Henry Baker, College of Medicine, University of Florida for providing the wound healing dataset. This research was partially supported by NSF Grants DMS-04-05543 (Booth and Casella) and DMS-05-03648 (Hobert).

A Derivation of Posterior

For a fixed partition, the maximizer of (6) with respect to β_k is given by

$$\hat{\beta}_k = \left[(1_{n_k r} \otimes X)^T M_k^{-1} (1_{n_k r} \otimes X)\right]^{-1} (1_{n_k r} \otimes X)^T M_k^{-1} Y_k^* \ .$$

To show that $\hat{\beta}_k = (X^T W_k X)^{-1} X^T W_k \bar{Y}_k$, we will establish the following two facts:

1. $(1_{n_k} \otimes 1_r \otimes X)^T M_k^{-1} (1_{n_k} \otimes 1_r \otimes X) = n_k r X^T W_k X$; and
2. $(1_{n_k} \otimes 1_r \otimes X)^T M_k^{-1} Y_k^* = n_k r W_k \bar{Y}_k$.

A key matrix inversion result that will be used is:

$$\left(I_{ml} + J_m \otimes C\right)^{-1} = I_{ml} - J_m \otimes (I_l + mC)^{-1} C \, . \tag{15}$$

Using (15), we obtain

$$\begin{aligned} M_k^{-1} &= \left[I_{n_k rp} - J_{n_k} \otimes (I_{rp} + n_k A^{-1}B)^{-1} A^{-1} B\right] (I_{n_k} \otimes A^{-1}) \\ &= I_{n_k} \otimes A^{-1} - J_{n_k} \otimes (I_{rp} + n_k A^{-1}B)^{-1} A^{-1} B A^{-1} \,. \end{aligned} \tag{16}$$

Now,

$$\begin{aligned} n_k A^{-1} B &= J_r \otimes \left[I_p - (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} r\lambda_1 Z_1 Z_1^T\right] n_k \lambda_2 Z_2 Z_2^T \\ &= J_r \otimes (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} n_k \lambda_2 Z_2 Z_2^T \\ &= J_r \otimes n_k D \,, \end{aligned}$$

where $D = (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} \lambda_2 Z_2 Z_2^T$. It follows that

$$\begin{aligned} &(I_{rp} + n_k A^{-1} B)^{-1} A^{-1} \\ &= (I_{rp} + J_r \otimes n_k D)^{-1} \left[I_{rp} - J_r \otimes (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} \lambda_1 Z_1 Z_1^T\right] \\ &= I_{rp} - J_r \otimes (I_p + r n_k D)^{-1} n_k D - J_r \otimes (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} \lambda_1 Z_1 Z_1^T \\ &\qquad - J_r \otimes (I_p + r n_k D)^{-1} n_k D (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} r\lambda_1 Z_1 Z_1^T \\ &= I_{rp} - J_r \otimes (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} \lambda_1 Z_1 Z_1^T \\ &\qquad - J_r \otimes (I_p + r n_k D)^{-1} n_k D (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} \,. \end{aligned}$$

This combined with (16) yields

$$\begin{aligned} M_k^{-1}(1_{n_k} \otimes 1_r \otimes X) &= 1_{n_k} \otimes \left[A^{-1} - (I_{rp} + n_k A^{-1} B)^{-1} n_k A^{-1} B A^{-1}\right] (1_r \otimes X) \\ &= 1_{n_k} \otimes (I_{rp} + n_k A^{-1} B)^{-1} A^{-1} (1_r \otimes X) \\ &= 1_{n_k r} \otimes W_k X \,, \end{aligned} \tag{17}$$

where we have used the fact that

$$\begin{aligned} W_k &= (I_p + r\lambda_1 Z_1 Z_1^T + n_k r\lambda_2 Z_2 Z_2^T)^{-1} \\ &= \left[I_p + (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} n_k r\lambda_2 Z_2 Z_2^T\right]^{-1} (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} \\ &= (I_p + r n_k D)^{-1} (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} \\ &= I_p - (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} r\lambda_1 Z_1 Z_1^T - (I_p + r n_k D)^{-1} r n_k D (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} \end{aligned}$$

The two facts now follow directly from (17). Thus, we may now write

$$\left[Y_k^* - (1_{n_k r} \otimes X)\beta_k\right]^T M_k^{-1} \left[Y_k^* - (1_{n_k r} \otimes X)\beta_k\right] = n_k r \left\{ \left(\beta_k - \hat{\beta}_k\right)^T X^T W_k X \left(\beta_k - \hat{\beta}_k\right) + p\hat{\sigma}_k^2 \right\}, \tag{18}$$

where

$$\hat{\sigma}_k^2 = \frac{1}{n_k r p} \left[Y_k^* - (1_{n_k r} \otimes X)\hat{\beta}_k\right]^T M_k^{-1} \left[Y_k^* - (1_{n_k r} \otimes X)\hat{\beta}_k\right] .$$

We conclude by establishing (10). By adding and subtracting the term $1_{n_k r} \otimes \bar{Y}_k$ and multiplying, we obtain

$$\begin{aligned} n_k r p \hat{\sigma}_k^2 \;=\; & (Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^T M_k^{-1} (Y_k^* - 1_{n_k r} \otimes \bar{Y}_k) \\ & + \left[1_{n_k r} \otimes (\bar{Y}_k - X\hat{\beta}_k)\right]^T M_k^{-1} \left[1_{n_k r} \otimes (\bar{Y}_k - X\hat{\beta}_k)\right] \\ & + 2(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^T M_k^{-1} \left[1_{n_k r} \otimes (\bar{Y}_k - X\hat{\beta}_k)\right] . \end{aligned}$$

Straightforward arguments similar to those above show that the cross-term (third term) is zero and that the second term can be written as $n_k r(\bar{Y}_k - X\hat{\beta}_k)^T W_k (\bar{Y}_k - X\hat{\beta}_k)$. Finally, the first term is

$$(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^T \left[I_{n_k} \otimes A^{-1} + J_{n_k} \otimes (I_{rp} + n_k A^{-1} B)^{-1} A^{-1} B A^{-1}\right] (Y_k^* - 1_{n_k r} \otimes \bar{Y}_k) .$$

It's easy to show that $BA^{-1} = J_r \otimes D^T$ and it follows from what was done above that

$$\left(I_{rp} + n_k A^{-1} B\right)^{-1} A^{-1} = I_{rp} - J_r \otimes H_k ,$$

where $H_k = (I_p + r\lambda_1 Z_1 Z_1^T)^{-1} \lambda_1 Z_1 Z_1^T + (I_p + r n_k D)^{-1} n_k D (I_p + r\lambda_1 Z_1 Z_1^T)^{-1}$. Hence,

$$(I_{rp} + n_k A^{-1} B)^{-1} A^{-1} B A^{-1} = J_r \otimes (I_p - r H_k) D^T = J_r \otimes G_k ,$$

say. It follows that

$$(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^T \left[J_{n_k} \otimes (I_{rp} + n_k A^{-1} B)^{-1} A^{-1} B A^{-1}\right] (Y_k^* - 1_{n_k r} \otimes \bar{Y}_k) =$$

$$(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^T (J_{n_k r} \otimes G_k)(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k) = 0 ,$$

and so,

$$
\begin{aligned}
&(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^T M_k^{-1} (Y_k^* - 1_{n_k r} \otimes \bar{Y}_k) \\
&\qquad = \quad (Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^T (I_{n_k} \otimes A^{-1}) (Y_k^* - 1_{n_k r} \otimes \bar{Y}_k) \\
&\qquad = \quad \sum_{i \in \mathcal{C}_k} (Y_i - 1_r \otimes \bar{Y}_k)^T A^{-1} (Y_i - 1_r \otimes \bar{Y}_k) \ .
\end{aligned}
$$

References

BANFIELD, J. D. and RAFTERY, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, **49** 803–821.

BINDER, D. A. (1978). Bayesian cluster analysis. *Biometrika*, **65** 31–38.

BOOTH, J. G., CASELLA, G., COOKE, J. E. K. and DAVIS, J. M. (2004). Clustering periodically-expressed genes using microarray data: A statistical analysis of the yeast cell cycle data. Tech. rep., Cornell University, Department of Biological Statistics and Computational Biology.

CELEUX, G. and GOVAERT, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, **14** 315–332.

CELEUX, G., LAVERGNE, C. and MARTIN, O. (2005). Mixture of linear mixed miodels: Application to repeated data clustering. *Statistical Modelling*, **5** 243–267.

CONSONNI, G. and VERONESE, P. (1995). A Bayesian method for combining results from several binomial experiments. *Journal of the American Statistical Association*, **90** 935–944.

CROWLEY, E. M. (1997). Product partition models for normal means. *Journal of the American Statistical Association*, **92** 192–198.

FRALEY, C. and RAFTERY, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, **97** 611–631.

GEYER, C. J. and THOMPSON, E. A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, **90** 909–920.

GREEN, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82** 711–732.

HARTIGAN, J. A. (1990). Partition models. *Communications in Statistics - Theory and Methods*, **19** 2745–2756.

HARTIGAN, J. A. and WONG, M. A. (1979). A K-means clustering algorithm. *Applied Statistics*, **28** 100–108.

HITCHCOCK, D. B., CASELLA, G. and BOOTH, J. (2005). Improved estimation of dissimilarities by smoothing functional data. *Journal of the American Statistical Association* to appear.

JAIN, S. and NEAL, R. M. (2004). A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, **13** 158–182.

JERRUM, M. and SINCLAIR, A. (1996). The Markov chain Monte Carlo method: An approach to approximate counting and integration, in *Approximation Algorithms for NP-hard Problems*. PWS Publishing, Boston.

JERRUM, M. and SORKIN, G. B. (1998). The Metropolis algorithm for graph bisection. *Discrete Applied Mathematics*, **82** 155–175.

KIRKPATRICK, S., GELATT, C. D. and VECCHI, M. P. (1983). Optimization by simulated annealing. *Science*, **220** 671–680.

MARINARI, E. and PARISI, G. (1992). Simulated tempering: A new Monte Carlo scheme. *Europhysics Letters*, **19** 451–458.

MCLACHLAN, G. J. and BASFORD, K. E. (1988). *Mixture models: Inference and Applications to Clustering*. Marcel Dekker, Inc., New York.

MCLACHLAN, G. J., DO, K.-A. and AMBROISE, C. (2004). *Analyzing Microarray Gene Expression Data*. John Wiley & Sons.

MCLACHLAN, G. J. and PEEL, D. (2000). *Finite Mixture Models*. Wiley.

NEAL, R. M. (1996). Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, **6** 353–366.

PITMAN, J. (1997). Some probabilistic aspects of set partitions. *American Mathematical Monthly*, **104** 201–209.

ROEDER, K. (1990). Density estimation with confidence sets exemplified by superclusters and voids in the galaxies. *Journal of the American Statistical Association*, **85** 617–624.

RUPPERT, D., WAND, M. P. and CARROLL, R. J. (2003). *Semiparametric Regression*. Cambridge University Press, New York.

SELIM, S. Z. and ALSUTAN, K. (1991). A simulated annealing algorithm for the clustering problem. *The Journal of the Pattern Recognition Society*, **24** 1003–1008.

SERBAN, N. and WASSERMAN, L. (2005). CATS: Clustering after transformation and smoothing. *Journal of the American Statistical Association*, **100** 990–999.

SPELLMAN, P., SHERLOCK, G., ZHANG, M. Q., IYER, R. I., ANDERS, K., EISEN, M. B., BROWN, P. O., 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.

STANLEY, R. P. (1997). *Enumerative Combinatorics*, vol. I. Cambridge University Press, New York.

Expression Profiles for Eight Yeast Genes

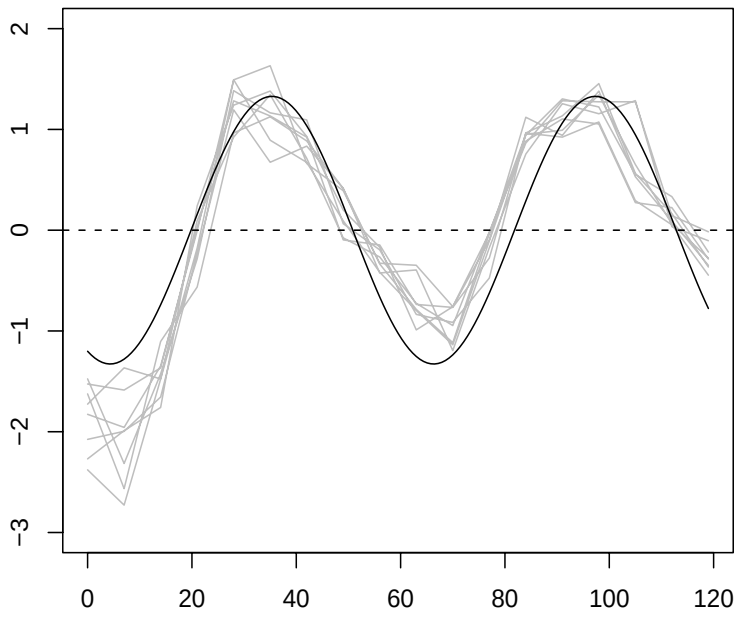

Figure 1: Gene expression profiles for the eight histones. Time is on the abscissa and log expression ratio is on the ordinate. The solid black line is the first-order Fourier series model (1) fit to the pointwise average profile.

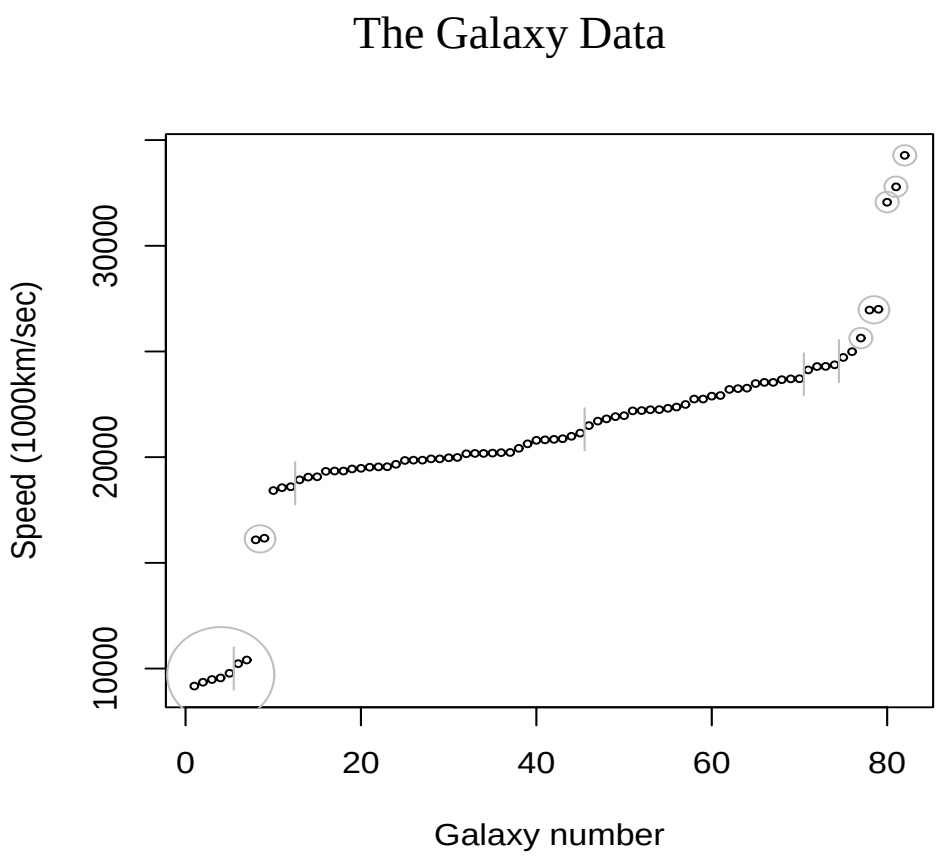

Figure 2: Speeds in kilometers per second of 82 galaxies. The galaxies are ordered from slowest to fastest. Circled clusters are separated from their nearest neighbors by at least 500 km/sec. Vertical bars indicate the positions of additional gaps of at least 300 km/sec.

Partitions of the Galaxy Data

Gap Positions	$c(\omega)$	$\ln \pi^*(\omega\|y)$
011010010100	6	-380.26
011010100100	6	-381.40
011010001100	6	-381.56
011010011100	7	-382.64
011011001100	7	-382.71
011010101100	7	-382.75
011010010101	7	-383.07
011010110100	7	-383.37
011010010110	7	-383.45
011011010100	7	-383.64
011010100101	7	-384.23
011010001101	7	-384.38
011011000100	6	-384.44
011010100110	7	-384.64
011010001110	7	-384.73
011011001101	8	-385.30
011010011101	8	-385.31
011010101101	8	-385.38
011010111100	8	-385.60
011011011100	8	-385.70
011010011110	8	-385.71
011011001110	8	-385.84
011010101110	8	-385.84
010000000100	3	-385.93
011010110101	8	-386.03

Table 1: Best 25 partitions of 82 galaxies out of 4096 that can be formed using the 12 gaps exceeding 300km/s. The gaps positions follow galaxy numbers 5, 7, 9, 12, 45, 70, 74, 76, 77, 79, 80, 81 respectively. The gap position sequence indicates where the boundaries between clusters are in the partition.

Clustering of the Yeast Cell Cycle Data

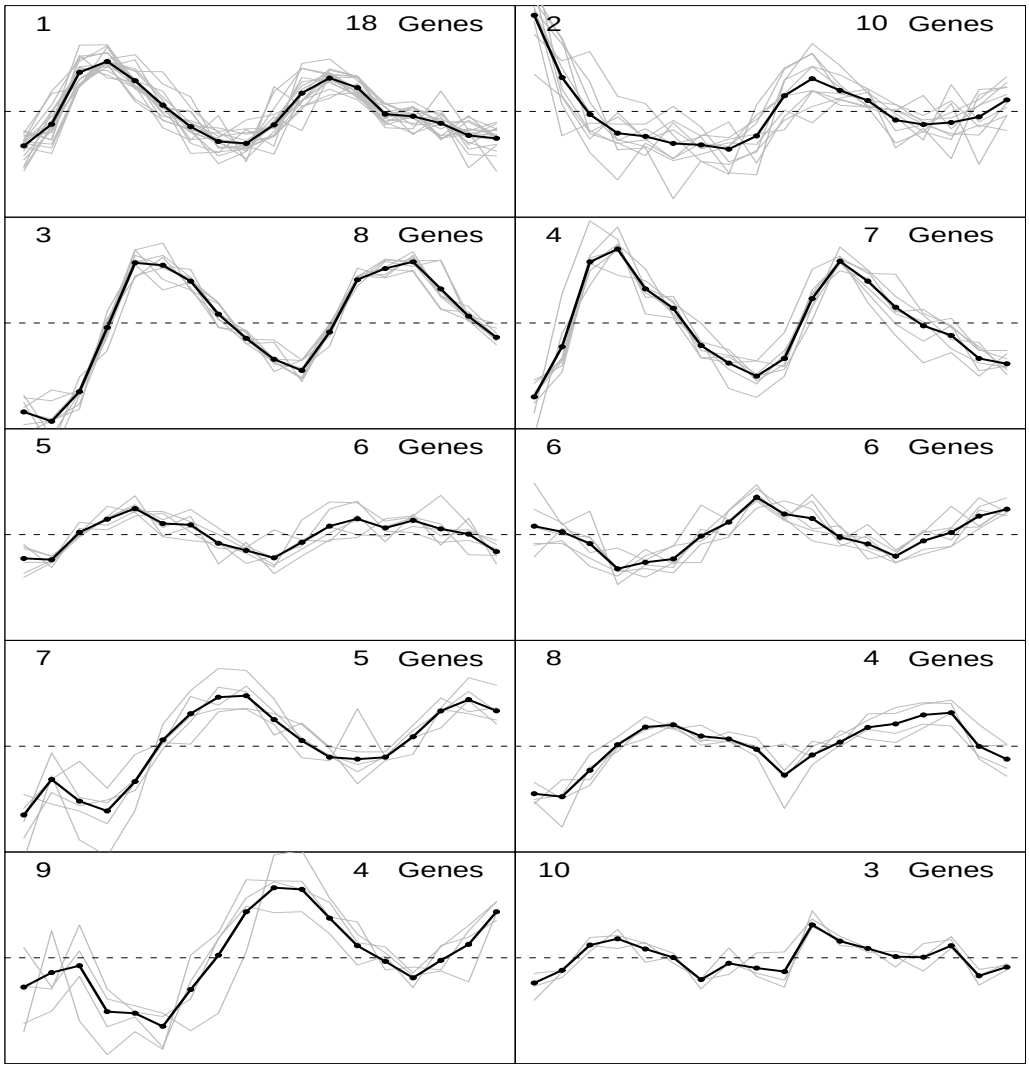

Figure 3: Clusters of yeast cell cycle gene profiles with 3 or more members. The solid black line joins the best linear unbiased predictors calculated using (13). The clusters not shown consisted of 6 doubletons, and 20 singletons. Cluster number 3 consists of the eight histones.

Most variable clusters of Wound Healing profiles

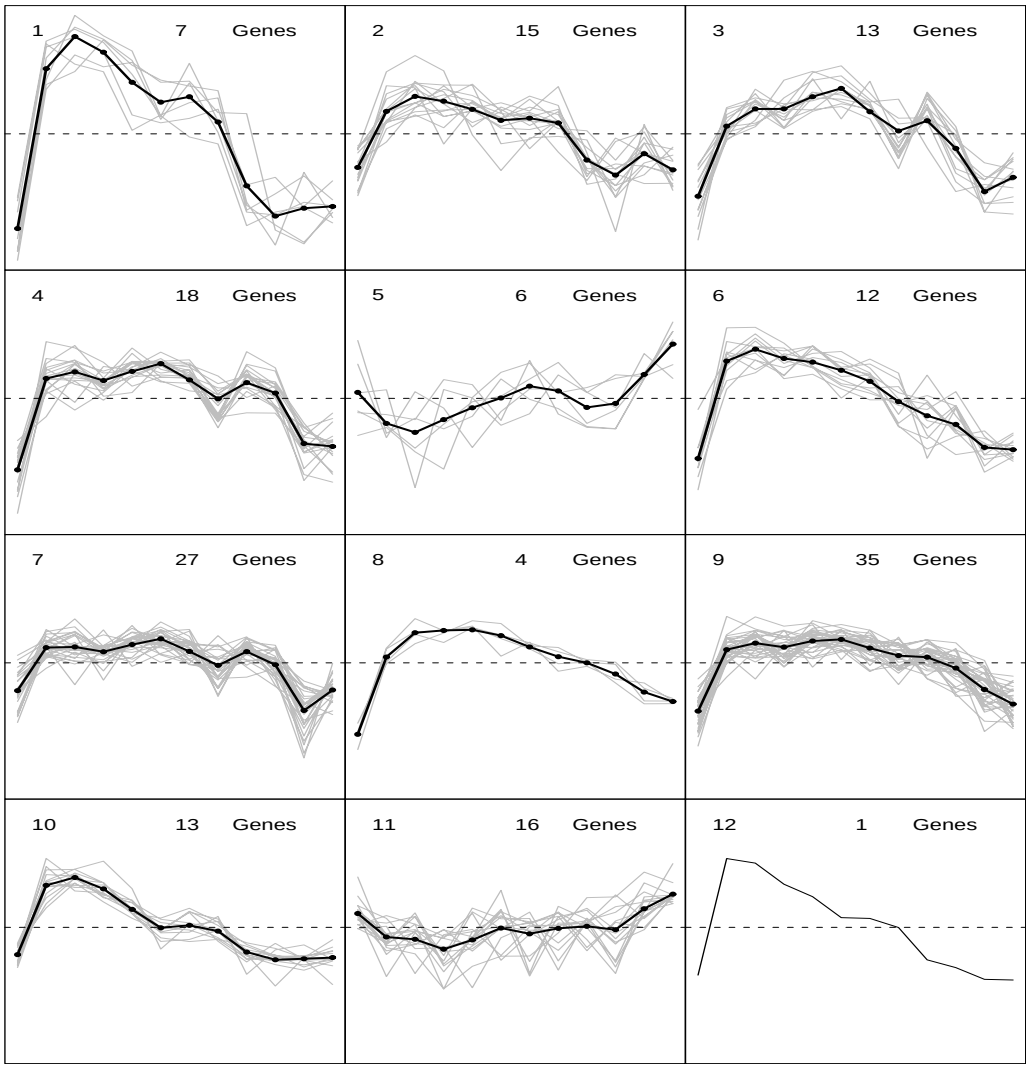

Figure 4: Twelve clusters from the corneal wound healing experiment with the highest range of expression values. Grey profiles are averages over the two replicates for each gene. The black lines join the best linear unbiased predictors at the twelve time points: 0, 1, 2, 3, 4, 5, 6, 7, 14, 21, 42, and 98 days. The time scale is transformed so that the points are equally spaced.