Statistics and Its Interface Volume 7 (2014) 543–557

Nonparametric Bayesian functional clustering for time-course microarray data

Ziwen Wei* and Lynn Kuo†,‡

Time-course microarray experiments track gene expression levels across several time points. They provide valuable insights into genome-wide dynamic aspects of gene regulations. We focus on gene clustering analysis in this paper. We explore a nonparametric Bayesian method for constructing clusters in functional space from the characteristics of gene profiles. In particular, we model each gene profile using a B-spline basis. So each gene is characterized by the basis coefficients of the spline fitting. Then we place a Dirichlet process prior on the basis coefficients to determine clusters of the genes. We essentially construct a hierarchical Dirichlet processes mixing model that assigns genes into the same cluster if they share the same latent basis coefficients. A simulation study is conducted to compare the proposed method to the K-means clustering method, a model-based clustering method (MCLUST), and a two-stage version of them in terms of the adjusted Rand index. We show our new method has better adjusted Rand index number among all these methods. We apply this nonparametric Bayesian clustering method to a real data set with 6 time points to gain further insights into how genes with similar profiles are clustered together and we find their functional annotation in Gene-Ontology groups using GOstats.

Keywords and phrases: Dirichlet process, Time-course microarray, Functional data analysis.

1. INTRODUCTION

DNA microarrays are used to measure the expression levels of a large number of genes simultaneously. Time-course microarray experiments track the progress of gene expressions along time across one or more experimental conditions. They provide valuable insight into the dynamic mechanisms underlying the observed biological processes. Time-course microarray data have been collected more and more frequently due to reduced costs for running experiments and increased need for studying dynamic gene regulations. Two typical types of downstream data analysis from microarrays are determining which genes are up or down regulated between normal and diseased tissues; and deciding how to cluster genes into groups based on their "similarities". We focus on the latter activity in this paper. Clustering is a grouping technique, also called unsupervised learning in data mining. It partitions data into a small number of sets, so the subjects within each set are similar, and the subjects between sets are dissimilar. It reduces the complexity of data sets, and aids biologists in interpreting them. Additionally, researchers are able to infer probable functions of new genes based on their knowledge of the known genes and clustering membership, since genes that belong to the same cluster may co-regulate and/or participate in the same pathway.

The proper statistical gene-level analysis for time-course microarrays requires more sophisticated tools and complex statistical models than that for a single time point. Given gene expressions of the same subject are typically collected at a few time points, the usual time series analysis cannot be applied. In order to track the temporal changes of gene expressions, we adopt the functional data analysis (FDA, Ramsay and Silverman (1997)) approach that accounts for time dependency in gene expression data monitored over unequally spaced times. In particular, for each gene in the gene set, we model its gene profile using a B-spline basis, because it is more stable than power basis numerically. Then, we consider the Dirichlet process prior $\mathcal{DP}(\alpha G_0)$ (Ferguson (1973)) on the distribution of the basis coefficients and use the clustering property of the Dirichlet process (DP) to cluster genes. The process is effective in high-dimensional data reduction because of the built-in Bayesian penalty criterion for model complexity. Moreover, it has the appealing property that one needs not to specify the number of clusters a priori. On the other hand, it also provides a tuning parameter α which controls the number of clusters. If we choose large α, the number of clusters tends to be large.

In order to evaluate the performance of our clustering algorithm, we conduct a simulation study where four data sets with different locations and variations of the gene expression intensities at fixed time points were generated. We compare our nonparametric Bayesian method to K-means clustering (MacQueen (1967)), MCLUST (Fraley and Raftery (1999), Fraley and Raftery (2000)) and a two-stage version of each of them in terms of adjusted Rand index. The K-means clustering algorithm is one of the simplest clustering algorithms which partitions the set of genes into k clusters so the resulting intracluster similarity is high but the intercluster similarity is low. It is widely used in areas such as computer

*Merck Research Laboratories.
†Department of Statistics, University of Connecticut.
‡Corresponding author.

science, geostatistics and computational biology. The number of clusters k needs to be specified prior to the analysis. MCLUST, also relatively easy to carry out using the codes provided by http://www.stat.washington.edu/mclust/, is a well-known model-based clustering procedure using mixtures of normal models. The number of clusters is selected by the BIC criterion. Our proposed nonparametric Bayesian functional clustering is also a model-based procedure, but the number of clusters does not need to be pre-specified. The two-stage version of the two methods for comparisons are based on curve clustering using B-splines (Abraham et al. (2003)). First, it fits a cubic B-spline curve for each gene, and second, it clusters the basis coefficients by K-means or MCLUST. In the evaluation, we set the number of clusters for the K-means method to be the true number of clusters for simplicity. However, in real life, the true number of clusters are usually unknown, so the performance is evaluated in favor of the K-means method.

There are many previously developed gene clustering methods for time-course data. However, many of the developed methods, such as Smyth (2004) and Tai and Speed (2006), treat the gene profiles as multivariate observations. Unfortunately, the trend information of the gene profiles has not been incorporated in these analyses. Many other researchers apply FDA approach to time-course microarray data. For example, Storey et al. (2005) propose to model a gene-specific expression over time as a linear expansion of natural cubic spline basis functions. Angelini et al. (2007) develop a fully Bayesian approach for functional data. It expands each of the time-course gene expression curves over some standard orthonormal basis, such as Legendre polynomials or Fourier basis, and uses a truncated Poisson distribution to model the number of terms in the expansion. It also extends the error distribution from normal to all scale mixtures of normal distributions including student's t and double-exponential distributions. Both of these papers are concerned with determining differentially expressed genes. On clustering methods, both Luan and Li (2003) and Ma et al. (2006) view observed temporal gene expression profiles coming from underlying smooth curves. The former uses mixed-effects model with B-splines and the latter uses mixed-effects smoothing spline method and the rejection-controlled EM algorithm for gene-to-cluster assignments. Fu et al. (2013) propose a random-effects mixture model with the Dirichlet process prior to conduct gene clustering. Song et al. (2007) develop a unified approach for gene clustering and dimension reduction based on FDA to group observed curves with respect to their shapes or patterns. Ray and Mallick (2006) propose a nonparametric Bayesian model for clustering the functional data using wavelet basis function, placing a Dirichlet process prior on the basis coefficients. Wavelet basis function is extremely flexible especially when there are sharp curvatures at particular locations. However, the method requires more advanced techniques and wavelet basis is typically for equally spaced time points, which is not always the case in microarray data. There are other previous works on nonparametric Bayesian mixture models for clustering genes. For example, Medvedovic and Sivaganesan (2002) propose a hierarchical Bayesian infinite mixture model and model averaging algorithm for clustering genes. Qin (2006) proposes an iterative weighted Chinese restaurant seating scheme with a predictive updating where the optimal number of clusters can be determined simultaneously with cluster assignment. Dahl (2006) proposes a least-squares clustering algorithm for a DP mixing model. Kim et al. (2006) introduce a latent binary vector formulation for identifying variables and use a DP mixtures model to define the cluster structure. They use repeated Metropolis steps and obtain inference on the cluster structure by a split-merge Markov chain Monte Carlo technique, so both variable selection and clustering can be done simultaneously. But all these papers apply DP mixing on the latent variables of the data directly.

Our paper contains a curve fitting component and the cluster assignment is done simultaneously by the DP mixing on the basis coefficients. Given microarray data are typically noisy, our spline fitting has a smoothing effect. Therefore, we build this smoother in our first level, and apply DP mixing in the second level to borrow strength among genes through their intrinsic characters (coefficients of spline fitting). Hence we expect that our algorithm is not only more robust than the previous versions due to spline smoothing, but also more accurate and efficient due to that the space of basis coefficients better represents the gene characters than the latent parameter space of the original data and also has smaller dimension. Dunson (2010) reviews thoroughly many models in biostatistics where DP mixing is useful. They include a model with cubic spline basis functions construction and a DP mixing on the coefficients. Our paper is very similar to this approach, except we use B-spline instead of cubic spline as basis functions. In addition to this contribution, our paper is more focused given we only consider clustering for time-course gene expression data. We describe how to improve efficiency by implementing the blocked Gibbs sampler from the truncated stick-breaking DP prior (Ishwaran and James (2001)) to update blocks of parameters to avoid slow mixing. Moreover, we added a simulation study that shows our method is superior to K-means, MCLUST, and their two-step versions in terms of the adjusted Rand index. Furthermore, we apply our method for clustering to a real data set from a microarray experiment. We hence investigate the characteristics of each obtained cluster using GOstats that provides more insights into the gene functions for each cluster that relates to the same latent pattern of the time-course gene expression curves.

We should note our method is sensitive to the choice of hyperparameters. So caution is needed to select the hyperparameters. We also recommend employing sensitivity study on the hyperparameters in real data analysis. Another suggestion is to standardize the data within each gene first using the mean and the standard deviation of them over the time points and replications for each gene, then choose the hy-

perparameter to be a vector of 0 with an identity variance and covariance matrix.

We start with some preliminary work in Section 2, then introduce our proposed hierarchical model in Section 3. Posterior inference and some related issues are addressed in Section 4. Hyperparameters selection is discussed in Section 5. Then we discuss adjusted Rand index that is used for performance evaluation in Section 6. In Section 7, we perform a simulation study to evaluate the performance of our newly developed nonparametric Bayesian functional clustering method using adjusted Rand index. Two well-known clustering algorithms, K-means and MCLUST and their two stage version are also compared with the proposed approach. Subsection 7.1 describes how we simulate the data, and the analysis results are shown in Subsection 7.2. In Section 8, our proposed approach is applied to a real microarray dataset using Illumina WG-6v1 BeadChip to study bone development, and the analysis results are further annotated with gene functions. We finish the article by discussions in Section 9 with some concluding remarks.

2. PRELIMINARY WORK

Along the lines of many above cited papers, we adopt the same FDA approach which treats the entire series of a gene's expression levels evaluated at several time points as observed from a single curve. We believe that FDA is appropriate because time-course data can be generated by some underlying smooth function and the discrete measurements collected are snapshots of that function at various time points. Let $y_i(t)$ be the observed gene expression value for gene i, $i = 1, 2, ..., N$, at time point t. Write $y_i(t) = \eta_i(t) + \epsilon_i(t)$, where η_i is the function profile for gene i, and the error term $\epsilon_i(t) \sim N(0, \sigma^2)$. The profile function $\eta_i(t)$ can be written as $\eta_i(t) = \sum_{l=0}^{L} \beta_{il}\phi_l(t)$, where $\{\phi_0(t), ..., \phi_L(t)\}$ is a set of basis functions, and $\boldsymbol{\beta_i} = (\beta_{i1}, ..., \beta_{iL})'$ are the basis coefficients for gene i. Commonly used basis functions include power basis, Fourier basis, spline basis, and wavelet basis.

We next review the Dirichlet process that defines a prior distribution on the distributions of the unobserved latent basis coefficients. Dirichlet process, $\mathcal{DP}(\alpha G_0)$, introduced by Ferguson (1973), defines a nonparametric distribution over a large space of distribution functions. It has two parameters, G_0, the base distribution measure, and $\alpha > 0$, the scaling or concentration parameter. This prior process has the advantages of being flexible and adaptable, yet incorporates a sparseness-favoring structure that combats the curse of dimensionality (Dunson (2010)). The incorporation is done automatically by the Bayesian penalty for model complexity, and centering on a base parametric model.

Sethuraman and Tiwari (1982) and Sethuraman (1994) have given a constructive expression for the random draw G from the $\mathcal{DP}(\alpha G_0)$ prior, that is useful for understanding the process and posterior updating:

$$G = \sum_{r=1}^{\infty} \pi_r \delta_{\varphi_r},$$

where φ_r's are i.i.d. from G_0; δ_{φ_r} is a measure of mass one concentrated at φ_r; and π_r's are the stick-breaking weights defined by

$$\pi_r = V_r \prod_{i=1}^{r-1} (1 - V_i), \tag{2.1}$$

with $V_r \overset{i.i.d.}{\sim} Beta(1, \alpha)$.

A draw from the Dirichlet process yields a random discrete distribution G (denoted by $G \sim \mathcal{DP}(\alpha G_0)$) with random locations of jumps as i.i.d. variates from G_0, and the weights on these jumps are given by the stick-breaking weights (2.1). One of the properties of Dirichlet process is the clustering property, which can be applied to gene clustering. Specifically, we place a Dirichlet process prior on the gene-specific characters, for example, basis coefficients. Then, genes with same characters are clustered together by the construction of the Dirichlet process.

3. NONPARAMETRIC BAYESIAN MODEL

A typical time-course microarray dataset consists of expression measurements of N genes across J time points. The number of genes N is usually in thousands, much larger than the number of time points J. Mostly, such experiments are unreplicated due to the high cost or other limitations. But in some cases replicates are done, but the number of replicates is small. We consider the most general case in formulating our model. We assume gene i is measured at time points t_j, with $j = 1, ..., J$, each with k_{ij} replicates.

B-spline basis functions are defined recursively using the following expressions (de Boor (1987)), for $l = 0, ..., L$:

$$\phi_{l,1}(t) = \begin{cases} 1, & \text{if } t_l \le t \le t_{l+1}; \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\phi_{l,m}(t) = \frac{(t - t_l)\phi_{l,m-1}(t)}{t_{l+m-1} - t_l} + \frac{(t_{l+m} - t)\phi_{l+1,m-1}(t)}{t_{l+m} - t_{l+1}}, \tag{3.1}$$

for $m = 2,...,M$, where M, the order of the B-spline, controls the degree $(M - 1)$ of the resulting polynomial in t and the continuity of the curve. The knot values are located at $\{t_0, t_1, ..., t_{L+M}\}$. Note that $M = 4$ yields the cubic B-spline basis. For convenience, we write $\phi_{l,4}(t) = \phi_l(t)$. As mentioned earlier, the B-spline basis usually consists of more than one curve segment. For example, when $L = 9$, the cubic B-spline basis consists of 10 functions. These 10 spline basis curves are plotted in Figure 1. In this study, we use cubic B-spline with $L = 9$.

Let y_{ijk} denote the observed replicate k of the gene intensity for gene i at time point j, we fit a model with a cubic B-spline function for the time-course data, $y_{ijk} = \sum_{l=0}^{L} \beta_{il}\phi_l(t_j) + \epsilon_{ijk}$, for $i = 1, ..., N$, $j = 1, ..., J$, $k = 1, ...k_{ij}$, and $\epsilon_{ijk} \overset{i.i.d.}{\sim} N(0, \sigma^2)$. The matrix form of the

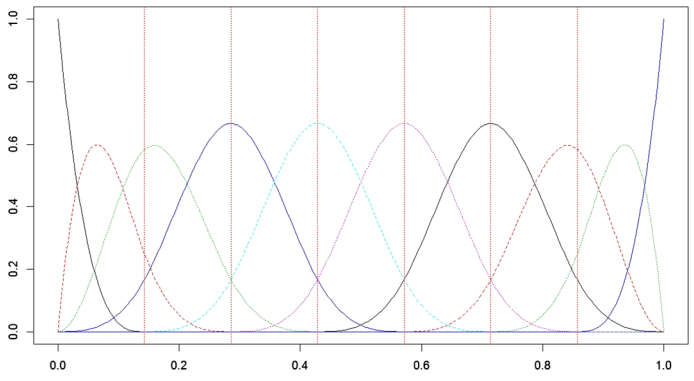

Figure 1. Cubic B-Spline Basis ($L = 9$).

model is given by

$$\boldsymbol{y}_i = X\boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i, \tag{3.2}$$

where $\boldsymbol{y}_i = (y_{i11}, ..., y_{i1k_{i1}}, ..., y_{iJ1}, ..., y_{iJk_{iJ}})'$ is the K_i-dim ($K_i = \sum_{j=1}^{J} k_{ij}$) column vector holding all measurements for gene i, $\boldsymbol{\beta}_i = (\beta_{i0}, ..., \beta_{iL})'$, X is the $K_i \times (L+1)$ block design matrix with the jth block being the vector $[\phi_0(t_j)...\phi_L(t_j)]$ stacked k_{ij} times, and $\boldsymbol{\epsilon}_i = (\epsilon_{i11}, ..., \epsilon_{i1k_{i1}}, ..., \epsilon_{iJ1}, ..., \epsilon_{iJk_{iJ}})'$ is the K_i-dim column vector of random errors following a multivariate normal distribution $MN(\mathbf{0}, \sigma^2 \boldsymbol{I}_{K_i})$. The nonparametric hierarchical model for the data can be described as follows: for $i = 1, \ldots, N$,

$$\begin{aligned} \boldsymbol{y_i}|\boldsymbol{\beta_i}, \sigma^2 &\overset{ind.}{\sim} MN(X\boldsymbol{\beta}_i, \sigma^2 \boldsymbol{I}_{K_i}) \\ \boldsymbol{\beta_i} &\overset{i.i.d.}{\sim} G \\ G &\sim \mathcal{DP}(\alpha G_0) \\ \tau = 1/\sigma^2 &\sim \mathcal{G}(n_0/2, W_0/2) \end{aligned} \tag{3.3}$$

where $\mathcal{G}(n_0/2, W_0/2)$ denotes a gamma distribution with mean n_0/W_0, and both G_0 and α are assumed to be known. The multivariate normal distribution $G_0 = MN_{L+1}(\boldsymbol{b}_0, B_0)$ is the base measure representing the prior mean of the random distribution G, with $\boldsymbol{b}_0$ and B_0 being the mean and variance-covariance matrix of the distribution of G_0, respectively. And α is the prior weight that represents the strength of the prior belief on G. Here we assume that $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_N$ are independently distributed from G, and given $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N)$ and σ^2, $\boldsymbol{y_1}, ..., \boldsymbol{y_N}$ are independent.

4. POSTERIOR INFERENCE

The posterior distribution for the above Dirichlet process mixture model is not analytically tractable. Therefore, we develop a Gibbs sampling algorithm for the posterior inference. By using conjugate priors, the complete conditional densities needed for the Gibbs sampler can be written explicitly for the updating purpose. Ishwaran and James (2001) proposed the blocked Gibbs sampler which assumed the prior G to be $\mathcal{DP}(\alpha G_0)$ with a finite dimensional base measure G_0 and truncated at the Rth term by letting $V_R = 1$ and V_r for $r = 1, \ldots, R-1$ defined exactly as in the stick-breaking representation (2.1). So it circumvents the infinitely many parameters in the Dirichlet process. The blocked Gibbs sampler also enables us to update blocks of parameters, hence it is very efficient.

Let S_i be a cluster allocation function and $S_i = h$ denote that gene i belongs to cluster h. For genes in the same cluster, say cluster h, the corresponding $\boldsymbol{y}_i$'s share the same basis coefficients, denoted by $\boldsymbol{\theta}_h$. By truncated stick-breaking construction of the Dirichlet process, we have $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_R \overset{ind.}{\sim} MN_{L+1}(\boldsymbol{b}_0, B_0)$. The updating scheme of the blocked Gibbs sampler works as follows:

Step 1. Allocate genes to one of the R clusters by sampling S_i, $i = 1, ..., N$, from a discrete distribution with probabilities:

$$\begin{aligned} &Pr(S_i = h|-) \\ &= \frac{\{V_h \prod_{r<h}(1-V_r)\} \prod_{j=1}^{J} \prod_{k=1}^{k_{ij}} N(y_{ijk}; \sum_{l=0}^{L} \theta_{hl}\phi_l(t_j), \sigma^2)}{\sum_{v=1}^{R} \{V_v \prod_{s<v}(1-V_s)\} \prod_{j=1}^{J} \prod_{k=1}^{k_{ij}} N(y_{ijk}; \sum_{l=0}^{L} \theta_{vl}\phi_l(t_j), \sigma^2)}, \\ &h = 1, ..., R. \end{aligned} \tag{4.1}$$

Step 2. Update stick-breaking weights V_h from the conjugate beta posterior distribution below with $V_R = 1$:

$$\begin{aligned} (V_h|-) &\overset{ind.}{\sim} Beta\left(1 + \sum_{i=1}^{N} 1(S_i = h), \alpha + \sum_{i=1}^{N} 1(S_i > h)\right), \\ &h = 1, ..., R-1. \end{aligned} \tag{4.2}$$

Step 3. Update the atoms $\boldsymbol{\theta_h}$:

$$(\boldsymbol{\theta_h}|\tau, -) \sim N(\boldsymbol{b_1}, B_1), \tag{4.3}$$

where $\boldsymbol{b_1} = B_1(B_0^{-1}\boldsymbol{b_0} + \tau \sum_{i:S_i=h} X'\boldsymbol{y_i})$, and $B_1^{-1} = B_0^{-1} + \tau X'X \sum_{i=1}^{N} 1(S_i = h)$.

Step 4. Update τ:

$$(\tau|\boldsymbol{\theta_h}, -) \sim \mathcal{G}(\frac{n_1}{2}, \frac{W_1}{2}), \tag{4.4}$$

where $n_1 = n_0 + \sum_{i=1}^{N} \sum_{j=1}^{J} k_{ij}$, and $W_1 = W_0 + \sum_{i=1}^{N} (\boldsymbol{y_i} - X\boldsymbol{\theta}_{S_i})'(\boldsymbol{y_i} - X\boldsymbol{\theta}_{S_i})$.

Steps 3 and 4 in the above Gibbs sampling algorithm are derived as below. By the prior information, we have

$$\pi(\boldsymbol{\theta_h}, \tau) \sim MN_{L+1}(\boldsymbol{b}_0, B_0)\mathcal{G}(\frac{n_0}{2}, \frac{W_0}{2}),$$

and likelihood

$$\begin{aligned} &L(\boldsymbol{y}_{i:S_i=h}; \boldsymbol{\theta_h}, \tau) \\ &\propto \tau^{\frac{1}{2}\sum_{i:S_i=h}\sum_{j=1}^{J} k_{ij}} \\ &\cdot \exp\{-\frac{\tau}{2} \sum_{i:S_i=h} (\boldsymbol{y_i} - X\boldsymbol{\theta_h})'(\boldsymbol{y_i} - X\boldsymbol{\theta_h})\}. \end{aligned}$$

Therefore, we have the posterior

$$f(\boldsymbol{\theta_h}, \tau|-) \propto \tau^{\frac{1}{2}(n_0+\sum_{i:S_i=h}\sum_{j=1}^{J} k_{ij})-1} \cdot \exp\left\{-\frac{1}{2}A_1 - \frac{\tau}{2}[W_0 + A_2]\right\} \propto \exp\{A_3\}\tau^{\frac{n_1}{2}-1}\exp\left\{-\frac{\tau}{2}W_1\right\}. \tag{4.5}$$

where $A_1 = (\boldsymbol{\theta_h} - \boldsymbol{b_0})'B_0^{-1}(\boldsymbol{\theta_h} - \boldsymbol{b_0})$, $A_2 = \sum_{i:S_i=h}(\boldsymbol{y_i} - X\boldsymbol{\theta_h})'(\boldsymbol{y_i} - X\boldsymbol{\theta_h})$ and $A_3 = -\frac{1}{2}(\boldsymbol{\theta_h} - \boldsymbol{b_1})'B_1^{-1}(\boldsymbol{\theta_h} - \boldsymbol{b_1})$. Then (4.3) and (4.4) follow directly from (4.5). Note we have omitted the indices of the previously obtained parameters in the above Gibbs sampler discussion for simpler notation.

Other than the sampling algorithm, there are other issues that we need to address, including: (a) how to determine the stick-breaking truncation level R; (b) how to obtain clusters from the Gibbs samples given different iterations may imply different clusters; and last but not least, (c) how to evaluate the performance of the algorithm. We address (a) and (b) next, and discuss (c) in Section 6.

Truncation level R

We start with R clusters in Step 1, where R serves as the maximum number of clusters used in the blocked Gibbs sampler. In practice, the total number of clusters would often end up to be smaller than R. Let m_R and m_∞ denote the marginal density of the data $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$ in (3.3) with the $\mathcal{DP}(\alpha G_0)$ random measure for G truncated at R level and without truncation, respectively. The following theorem from Ishwaran and James (2001) provides a guidance on how to determine the truncation level R:

$$\int_{\mathcal{R}^{NJK}} |m_R(\boldsymbol{y}) - m_\infty(\boldsymbol{y})|d\boldsymbol{y} \le 4\left[1 - E\left\{\left(\sum_{r=1}^{R-1}\pi_r\right)^N\right\}\right] \approx 4N\exp\{-(R-1)/\alpha\}, \tag{4.6}$$

where N is the total number of subjects to be clustered. We also assume that $k_{ij} = K$ for all i and j. To make quantity (4.6) small enough, say less than a small positive number $\epsilon = 10^{-6}$, we solve the following equation for R:

$$4N\exp\{-(R-1)/\alpha\} = \epsilon. \tag{4.7}$$

For example, given $N = 100$ and $\alpha = 0.5$, (4.7) suggests truncating the stick-breaking expression (2.1) at R = 11, and for $\alpha = 1$, truncating it at $R = 21$.

Least-squares clustering

Each iteration from the Gibbs sampler yields a clustering rule that assigns genes into nonoverlapping clusters. These clustering rules are labeled by c_1, c_2,..., c_b,..., c_B corresponding to B iterations after burn-in. Out of all methods that estimate the clustering using draws from the posterior clustering distribution, the most straightforward one is to select the observed clustering that maximizes the density of the posterior clustering distribution (also known as the maximum *a posteriori* or MAP clustering). MAP may select a slightly more probable clustering than the next best alternative, but it may yield a very different allocation from the latter. Medvedovic and Sivaganesan (2002) define an $N \times N$ association matrix $\delta(c_b)$, for each b, with the (i,j)-th element $\delta_{ij}(c_b)$ equaling 1 if gene i and gene j are clustered together by the c_b, and equaling 0 otherwise. They further define a pairwise distance measure by $d_{ij}(c_b) = 1 - \delta_{ij}(c_b)$. Then they use complete linkage approach (Everitt (1993)) based on these distances measures to cluster similar expression profiles. It has been criticized by Dahl (2006) as ad hoc. Dahl (2006) introduces the least-squares clustering, which is based on the pairwise probability matrix $\hat{\pi}$ that is formed by averaging over all $\delta(c_b)$ over b elementwise. Then the algorithm identifies the optimal clustering by selecting the one that minimizes the sum of squared deviations from the average probability matrix $\hat{\pi}$. So it takes into account the information from all clusterings. We use Dahl's least-squares clustering to select the optimal clustering in the Markov chain. Then the least-squares clustering, denoted by c_{LS}, is obtained by

$$c_{LS} = \underset{c\in\{c_1,\ldots,c_B\}}{\operatorname{argmin}} \sum_{i=1}^{N}\sum_{j=1}^{N}(\delta_{ij}(c) - \hat{\pi}_{ij})^2, \tag{4.8}$$

where $\hat{\pi}_{ij}$ denotes the (i,j)-th element in the average probability matrix $\hat{\pi}$.

Note Wu et al. (2014) recently propose a new method for tracking configuration for DP sampling which can be used to construct clusters. It would be worthwhile to compare the least squares clustering to this new method for future work.

5. HYPERPARAMETERS SELECTION

In (3.3), we have assumed that τ has a gamma prior with a shape parameter $n_0/2$ and a scale parameter $W_0/2$, yielding a mean of n_0/W_0 and a variance of $2n_0/{W_0}^2$. When setting the hyperparameters n_0 and W_0, we can choose values such that W_0/n_0 matches the estimated variance of the data. For example, if the variance of the dataset is estimated to be 0.01, we can choose $n_0 = 20{,}000$ and $W_0 = 200$, so that $W_0/n_0 = 0.01$ matches the estimated variance, and the variance of the prior τ at this time is $2n_0/{W_0}^2 = 1$. Note that when the location parameter matches the data, the dispersion can be relatively small.

The hyperparameters $\boldsymbol{b}_0$ and B_0 for the mean and variance-covariance of G_0 should be chosen with caution, because the number of clusters is very sensitive to the selection of these two hyperparameters (Dunson (2010)). One of the solutions is to standardize the data within each gene first using the mean and the standard deviation of them over the time points and replications of each gene, then choose $\boldsymbol{b}_0$ to be a vector of 0 and B_0 to be an identity matrix.

Table 1. Contingency table of two partitions

	v_1	v_2	...	v_S	Total
u_1	n_{11}	n_{12}	...	n_{1S}	$n_{1.}$
u_2	n_{21}	n_{22}	...	n_{2S}	$n_{2.}$
.	.	.		.	.
.	.	.		.	.
u_R	n_{R1}	n_{R2}	...	n_{RS}	$n_{R.}$
Total	$n_{.1}$	$n_{.2}$	...	$n_{.S}$	$n_{..} = N$

The Dirichlet process imposes a prior on the number of clusters, which depends on the total number of genes and the concentration parameter of the Dirichlet process α. The number of clusters increases with increasing α. By default, we let $\alpha = 1$, which is one of the common choices in applications. Of course, one can set a gamma hyperprior, say $\mathcal{G}(1,1)$, for α to obtain more robust results (Escobar and West (1995)).

6. ADJUSTED RAND INDEX

One important issue in cluster analysis is the evaluation of clustering results, also referred to as cluster validation, which is to assess the quality of the clustering relative to clustering created by other algorithms, or by the same algorithm using different parameter settings. We will consider Rand index and adjusted Rand index for such evaluation.

The Rand index is a measure of agreement between two clusterings for a pair of objects. It is defined by the proportion of the number of agreements by two clustering methods for a pair of objects out of all possible pairs. More specifically, let $\mathbf{U} = \{u_1, ..., u_R\}$ and $\mathbf{V} = \{v_1, ..., v_S\}$ be the two resulting partitions of a set of N objects of $\mathbf{Y} = \{Y_1, ..., Y_N\}$ according to the clustering c and c', respectively. We can construct an $R \times S$ contingency table (Table 1) for the two clusterings with the (i, j)-th element, n_{ij}, as the number of common objects between the ith subset of partition c and jth subset of partition c'.

Consider selecting two objects at random from N objects, there are $\binom{N}{2}$ possible distinct pairs. The Rand index $Rand(c, c')$ of the two clusterings c and c', is defined as the chance of agreement where a pair of objects are placed in the same subset or in different subsets by both c and c', i.e.,

$$Rand(c, c') = \frac{A}{\binom{N}{2}},$$

where A counts the number of agreements between c and c'. Brennan and Light (1974) showed

$$A = \binom{N}{2} + 2\sum_{i=1}^{R}\sum_{j=1}^{S}\binom{n_{ij}}{2} - \left[\sum_{i=1}^{R}\binom{n_{i.}}{2} + \sum_{j=1}^{S}\binom{n_{.j}}{2}\right].$$

The adjusted Rand index in Hubert and Arabie (1985) is a corrected-for-chance version of the Rand index, which measures how often pairs of observations are agreed by clustering rules adjusting for the expected chance agreements. It has the following general form:

$$\frac{\text{Index} - \text{Expected Index}}{\text{Maximum Index} - \text{Expected Index}}. \tag{6.1}$$

So the adjusted Rand index $Rand'(c, c')$ for c and c', derived from (6.1), is given by

$$Rand'(c, c') = \frac{D_1 - D_2}{\frac{1}{2}D_3 - D_2}, \tag{6.2}$$

as shown in Hubert and Arabie (1985) and Rand (1971), where $D_1 = \sum_{i=1}^{R}\sum_{j=1}^{S}\binom{n_{ij}}{2}$, $D_2 = \sum_{i=1}^{R}\binom{n_{i.}}{2}\sum_{j=1}^{S}\binom{n_{.j}}{2}/\binom{n_{..}}{2}$ and $D_3 = \sum_{i=1}^{R}\binom{n_{i.}}{2} + \sum_{j=1}^{S}\binom{n_{.j}}{2}$.

Note $Rand'(c, c')$ ranges from 0 to 1. Larger $Rand'(c, c')$ suggests higher similarity between c and c'. And $Rand'(c, c') = 1$ indicates perfect agreement between the two clustering rules. The adjusted Rand index is a preferred measure to evaluate the performance of a clustering algorithm because of its normalized value. In our simulation study, we use the adjusted Rand index to measure the similarity between the clustering from each algorithm and the simulated truth.

7. SIMULATION STUDY

We first simulate 6 data sets by three choices of σ (0.1, 0.3 or 0.5) and two choices of offsets (0.5 or 1) with details given in Subsection 7.1. Then we describe the prior parameters chosen for the proposed nonparametric clustering method in the same subsection. We compare the results of our method to that of the K-means method, MCLUST method and two-stage version of the above methods in Subsection 7.2.

7.1 Simulated data

We generate 6 synthetic data sets where each set contains the gene expressions for 100 genes, 2 replicates each, at 4 time points ($N = 100$, $J = 4$, $k_{ij} = K = 2$ for all i and j). The 4 time points, unevenly spaced between 0 and 10, are assumed to be 0, 3, 7 and 10. First, we generate all data y_{ijk} independently from $N(0, \sigma^2)$ with σ known for each $i = 1, \ldots, 100$, $j = 1, \ldots, 4$ and $k = 1, 2$. Then, we add or subtract the offset effect at different time points to formulate 10 clusters with 10 genes in each cluster. Specifically, for the ith cluster, with $i = 1, \ldots, 4$, add the offset effect of 1 to the data points at the ith time point for all the 10 genes in this cluster; when $i = 5, \ldots, 8$, subtract the offset effect of 1 from the data point at the $i \bmod 4$ time point respectively; for the 9th cluster, add the offset effect 1 to all the data points at both the 1st and the 3rd time points; and for the 10th cluster, add the offset effect 1 to all the data points at both the 2nd and the 4th time points.

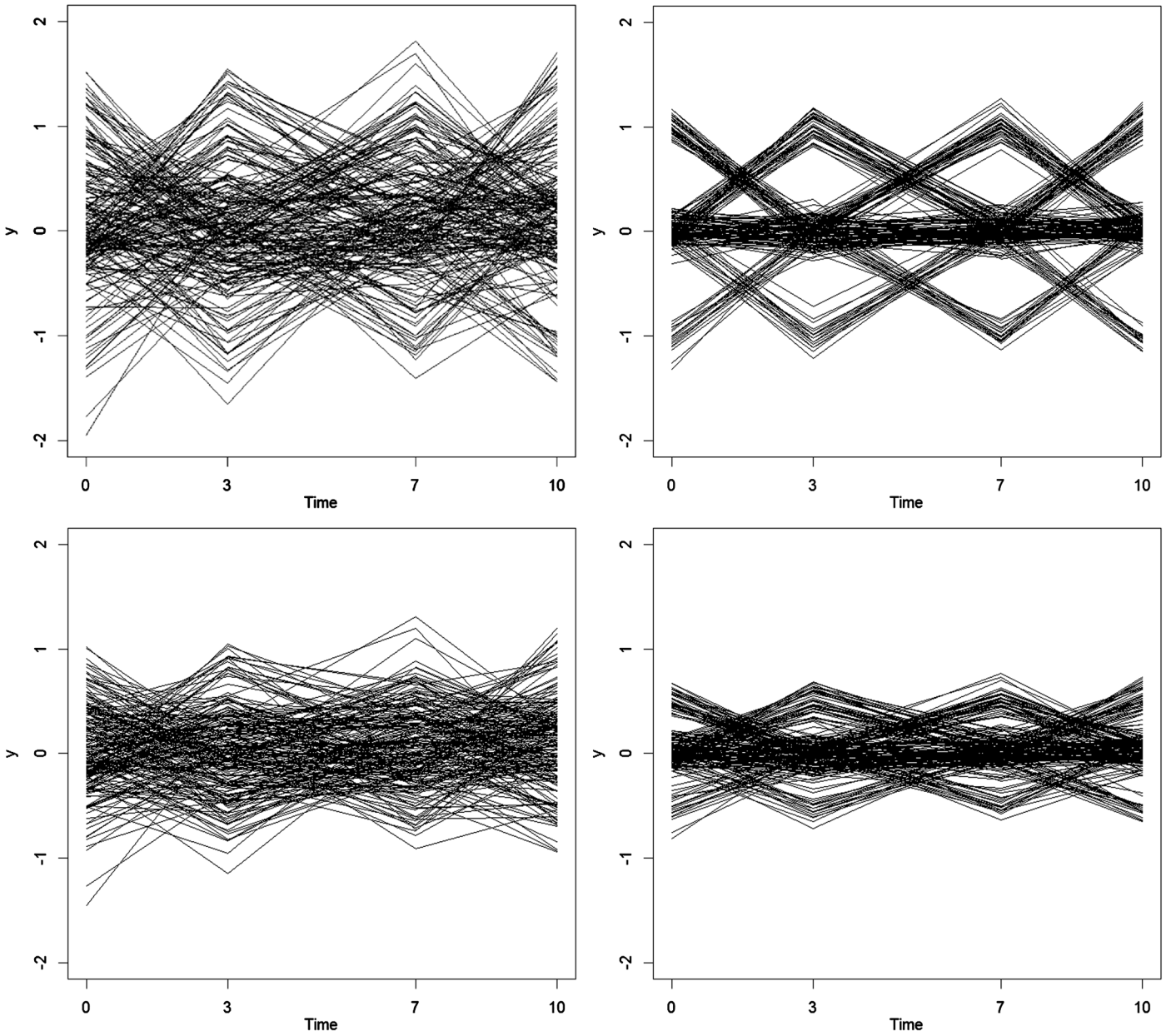

Figure 2. Simulated data: upper left is the high-effect case (offset effect = 1) with $\sigma = 0.3$, upper right is the high-effect case with $\sigma = 0.1$, lower left is the low-effect case (offset effect = 0.5) with $\sigma = 0.3$, lower right is the low-effect case with $\sigma = 0.1$.

Here we set σ to be 0.1, 0.3 or 0.5, to reflect different variations for the data. Moreover, we also repeat the three simulation studies for a low offset effect level of 0.5 instead of 1. To distinguish between the different offset levels, we refer the case with offset effect 1 to be "high-effect case", and the other to be "low-effect case". Figure 2 shows the plots for the four simulated data sets ($\sigma = 0.1$ and 0.3, each with high and low effect) to visually provide some ideas about the simulated data.

Next, we use our proposed Bayesian functional clustering approach (called DP clustering) to cluster the 100 genes. To perform the blocked Gibbs sampler on the simulated data, we need to specify prior parameters. Given $\sigma = 0.3$, we choose $n_0 = 2{,}200$ and $W_0 = 200$, so that $\tau \sim \mathcal{G}(1{,}100, 100)$. The stick-breaking weights V_1,..., V_{R-1} have initial values sampled independently from $Beta(1, \alpha)$, where $\alpha = 1$ or 0.5. Also, as mentioned in Section 5, the initial values of basis coefficient $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_R$ are sampled independently from a 10-dim multivariate normal $MN(\boldsymbol{b}_0, B_0)$ distribution, where $\boldsymbol{b}_0$ is chosen to be a 10-dim $\mathbf{0}$ vector, and B_0 is chosen to be a 10×10 matrix with all diagonal elements equal to 1 and 0 elsewhere.

Table 2. Comparison of the Bayesian functional clustering to K-means and MCLUST methods: high-effect case

	Method	Adjusted Rand index Mean (Standard Error)
$\sigma = 0.1$	Bayesian functional clustering ($\alpha = 1$)	0.9340 (0.0034)
	Bayesian functional clustering ($\alpha = 0.5$)	0.9286 (0.0073)
	MCLUST	0.8936 (0.0000)
	K-means clustering	0.8252 (0.0090)
	Two-stage MCLUST	0.3973 (0.0193)
	Two-stage K-means clustering	0.2053 (0.0050)
$\sigma = 0.3$	Bayesian functional clustering ($\alpha = 0.5$)	0.8847 (0.0073)
	K-means clustering	0.7618 (0.0072)
	Bayesian functional clustering ($\alpha = 1$)	0.7191 (0.0042)
	MCLUST	0.6443 (0.0229)
	Two-stage MCLUST	0.5112 (0.0248)
	Two-stage K-means clustering	0.1696 (0.0036)
$\sigma = 0.5$	Bayesian functional clustering ($\alpha = 0.5$)	0.5411 (0.0086)
	Bayesian functional clustering ($\alpha = 1$)	0.4249 (0.0053)
	K-means clustering	0.3119 (0.0047)
	Two-stage K-means clustering	0.1158 (0.0028)
	Two-stage MCLUST	0.0155 (0.0055)
	MCLUST	<0.0001 (<0.0001)

Table 3. Comparison of the Bayesian functional clustering to K-means and MCLUST methods: low-effect case

	Method	Adjusted Rand index Mean (Standard Error)
$\sigma = 0.1$	Bayesian functional clustering ($\alpha = 0.5$)	0.8897 (0.0080)
	MCLUST	0.8791 (0.0015)
	Bayesian functional clustering ($\alpha = 1$)	0.8740 (0.0039)
	K-means clustering	0.8604 (0.0078)
	Two-stage MCLUST	0.3035 (0.0193)
	Two-stage K-means clustering	0.1998 (0.0046)
$\sigma = 0.3$	Bayesian functional clustering ($\alpha = 0.5$)	0.3771 (0.0070)
	Bayesian functional clustering ($\alpha = 1$)	0.3112 (0.0050)
	K-means clustering	0.2077 (0.0038)
	Two-stage K-means clustering	0.0893 (0.0023)
	Two-stage MCLUST	0.0051 (0.0023)
	MCLUST	0.0009 (0.0005)
$\sigma = 0.5$	Bayesian functional clustering ($\alpha = 0.5$)	0.1391 (0.0037)
	Bayesian functional clustering ($\alpha = 1$)	0.1086 (0.0031)
	K-means clustering	0.0753 (0.0018)
	Two-stage K-means clustering	0.0417 (0.0019)
	Two-stage MCLUST	0.0009 (0.0008)
	MCLUST	<0.0001 (<0.0001)

7.2 Simulation results

We consider $\alpha = 0.5$ and $\alpha = 1$ for our proposed Bayesian function clustering algorithm. According to (4.7), they lead to maximum numbers of clusters $R = 11$ and $R = 21$, respectively for $N = 100$. We compare our method with two choices of α to the K-means clustering, MCLUST, and their two-step versions in terms of the adjusted Rand index. The mean of the adjusted Rand index (6.2) and its standard error are calculated from 100 replicated synthetic datasets for each of the six clustering methods.

The results are shown in Tables 2 and 3 for high-effect and low-effect cases, respectively. In each table, we see that our proposed Bayesian functional clustering achieves higher adjusted Rand index among all methods. The choice of $\alpha = 0.5$ is usually better than $\alpha = 1$ except for the high effect case with smallest variation $\sigma = 0.1$. In high-effect case (Table 2), the between-cluster variation in the simulated data is larger, therefore, it is relatively easier for these algorithms to separate clusters precisely, hence in general, the adjusted Rand index values in the high-effect case are higher than that in the low-effect case. Moreover, in each table, the ad-

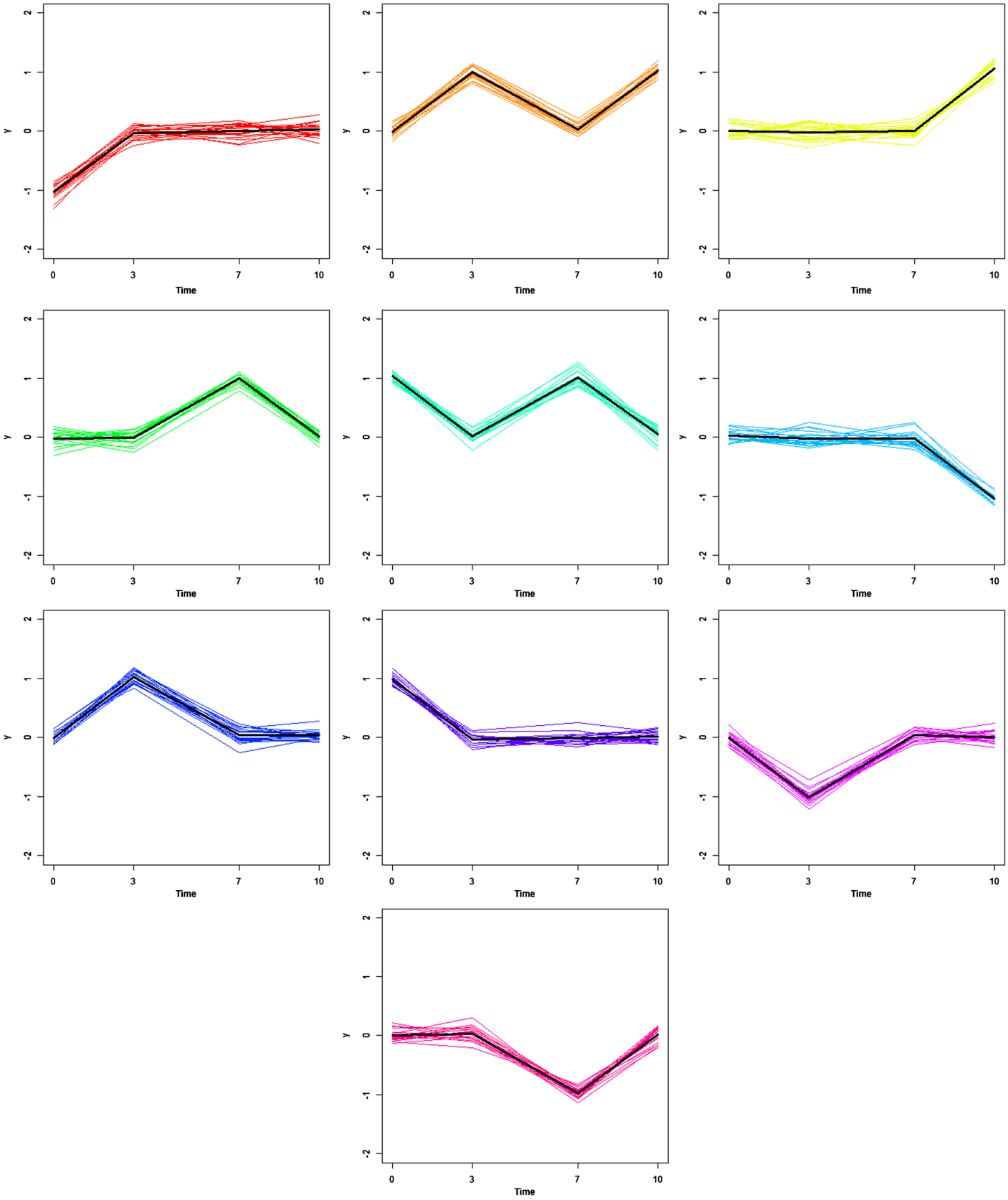

Figure 3. Plot of resulting clusters by the DP clustering with $\alpha = 0.5$ *for the simulated data with* $\sigma = 0.1$ *and high offset.*

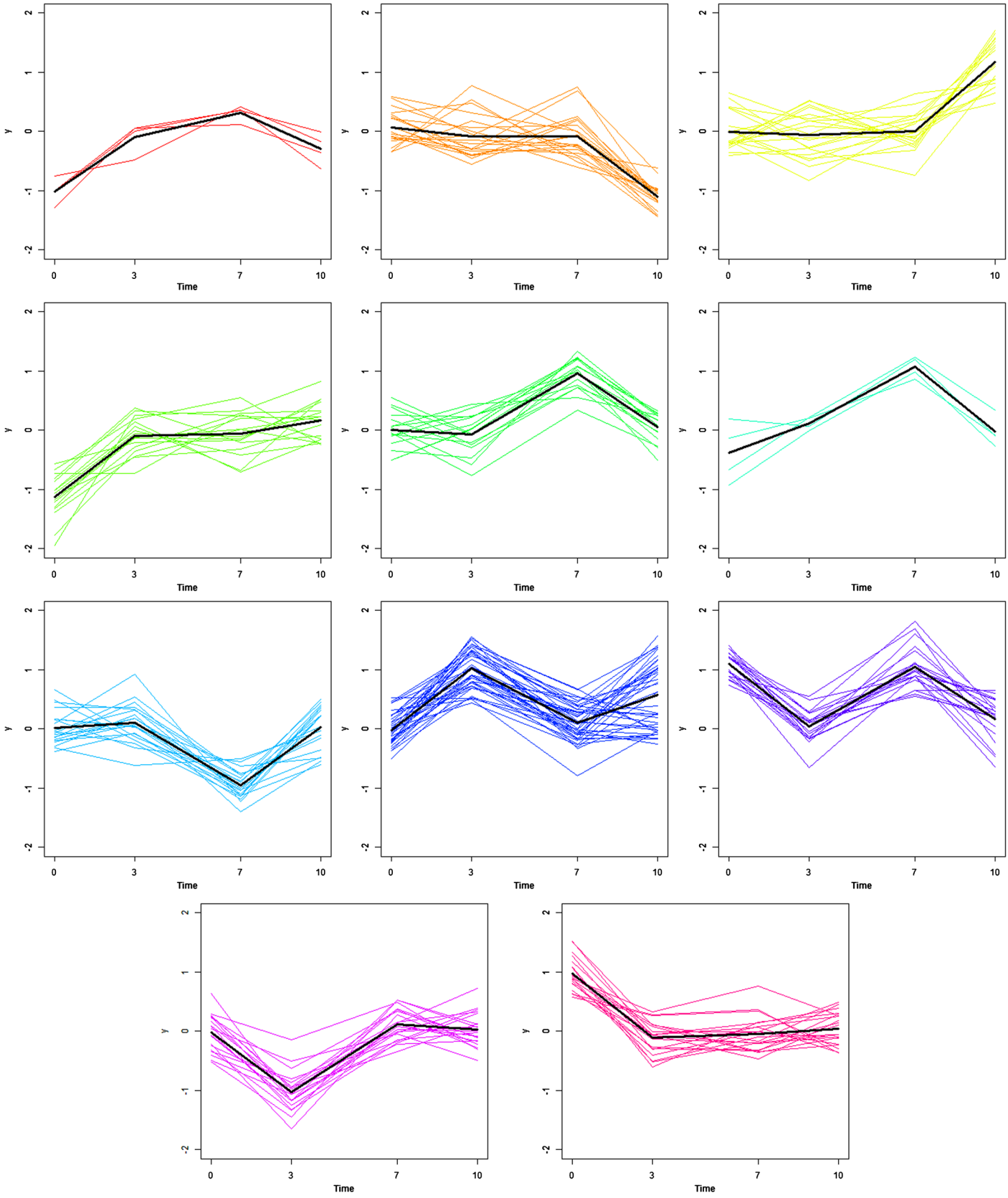

Figure 4. Plot of resulting clusters by the DP clustering with $\alpha = 0.5$ for the simulated data with $\sigma = 0.3$ and high offset.

justed Rand index values are higher for smaller σ. This is reasonable in that genes with smaller within-cluster variation stay closer together, which also make it easier for the algorithms to cluster them correctly. We also observe the Bayesain functional clustering has much better performance than the MCLUST for the low-effect case whereas the difference is not so distinct for the high effect case. This is perhaps due to the high-effect case, every method is doing better, so their distinction is not so pronounced. When the effect gets low and the σ increases, all methods are not doing well. In particular, the MCLUST performs a lot worse. This is perhaps due to the overparametrization of the MCLUST in this case in contrast to that the Bayesian functional clustering tends to be more parsimonious in model selection.

We see the Bayesian functional clustering is also better than K-means and MCLUST for each case, it is primarily due to the excellent clustering capability for the DP mixing for the former. It may be argued that the proposed method clusters the coefficients for the B-spline basis whereas K-means and MCLUST clusters the original data, which makes the comparison a little 'unfair'. So we added the two stage procedure as in Abraham et al. (2003). It can be seen from the simulations, the proposed Bayesian functional clustering method is much better than the two-stage methods with a fixed σ. The two classes of methods all use spline smoothing and then cluster genes using the coefficients of spline fitting. However, there are two major differences between them. (1) The two-stage method conducts data fitting using splines for all the data first. Then it clusters genes explicitly based on the results of stage 1. The Bayesian functional clustering does the two-stages hierarchically where the clustering on the second stage is done implicitly from the latent variables constructed in stage 1. (2) The biggest difference is in the clustering techniques. The Bayesian functional clustering uses DP mixing for clustering, and the other class uses k-means and MCLUST. Our interpretation from the simulation study is that the good performance of our proposed approach attributes more to the DP mixing than to the spline fitting. We suspect that spline fitting alone in this particular simulated data does not necessarily do a good job. Furthermore, the two-stage methods cluster the data indirectly by clustering the fitted coefficients rather than the original data. Hence they may work relatively limited comparing to one-stage methods which cluster the original data directly.

Using high-effect case (offset effect = 1) with $\alpha = 0.5$ as an example, we plot the clustering results using the Bayesian functional clustering in Figures 3 ($\sigma = 0.1$) and 4 ($\sigma = 0.3$). For each case, we only show the plots for one simulation to illustrate the clusters graphically. However, for different simulated data, these plots can be different due to randomness of the noise added to the data. Each cluster is presented in one plot with the bold curve representing the estimated curve described by the model in (3.3).

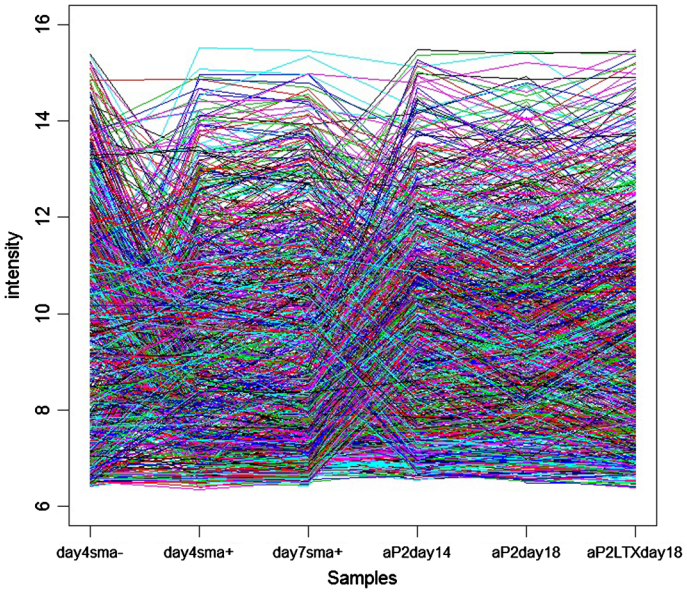

Figure 5. Plot of mouse adipose data.

8. REAL DATA ANALYSIS

8.1 Mouse adipose tissue data

For a real data analysis, we use Illumina data generated from microarray experiments on mouse adipose tissue from Dr. David Rowe's lab at the University of Connecticut Health Center. The objective is to examine gene expression changes during a series of 6 consecutive cellular events including preadipoblast, early adipoblast, and mature adipoblast. These six events are labeled in order by different markers as day4sma−, day4sma+, day7sma+, aP2day14, aP2day18, and aP2LTXday18. There are a total of 16,454 genes in the dataset and 2 replicates for each cellular events. The gene expression intensities are normalized first using the R Bioconductor package "lumi" (Du et al. (2008)) and averaged over 2 replicates to represent the gene intensity at each cellular event. In our cluster analysis, we concentrate on the $N = 1{,}040$ most differentially expressed genes selected by the rule of more than 3-fold changes in at least one of the adjacent pairwise comparisons. The data is plotted in Figure 5.

8.2 Analysis results

We apply the Bayesian functional clustering algorithm to the dataset of $N = 1{,}040$ genes. We choose $\alpha = 1$ (R = 24 by (4.7)) and σ^2 (=$1/\tau$) prior as suggested in Section 5. Given the observed sample standard deviation is roughly 2, we have chosen $n_0 = 0.1$ and $W_0 = 0.4$ as one of the many ways that match the empirical variance to W_0/n_0. The proposed Bayesian clustering algorithm groups the 1,040 genes into 16 clusters, with cluster size ranges from 29 to 173 genes. The plot for the 16 clusters are shown in Figure 6.

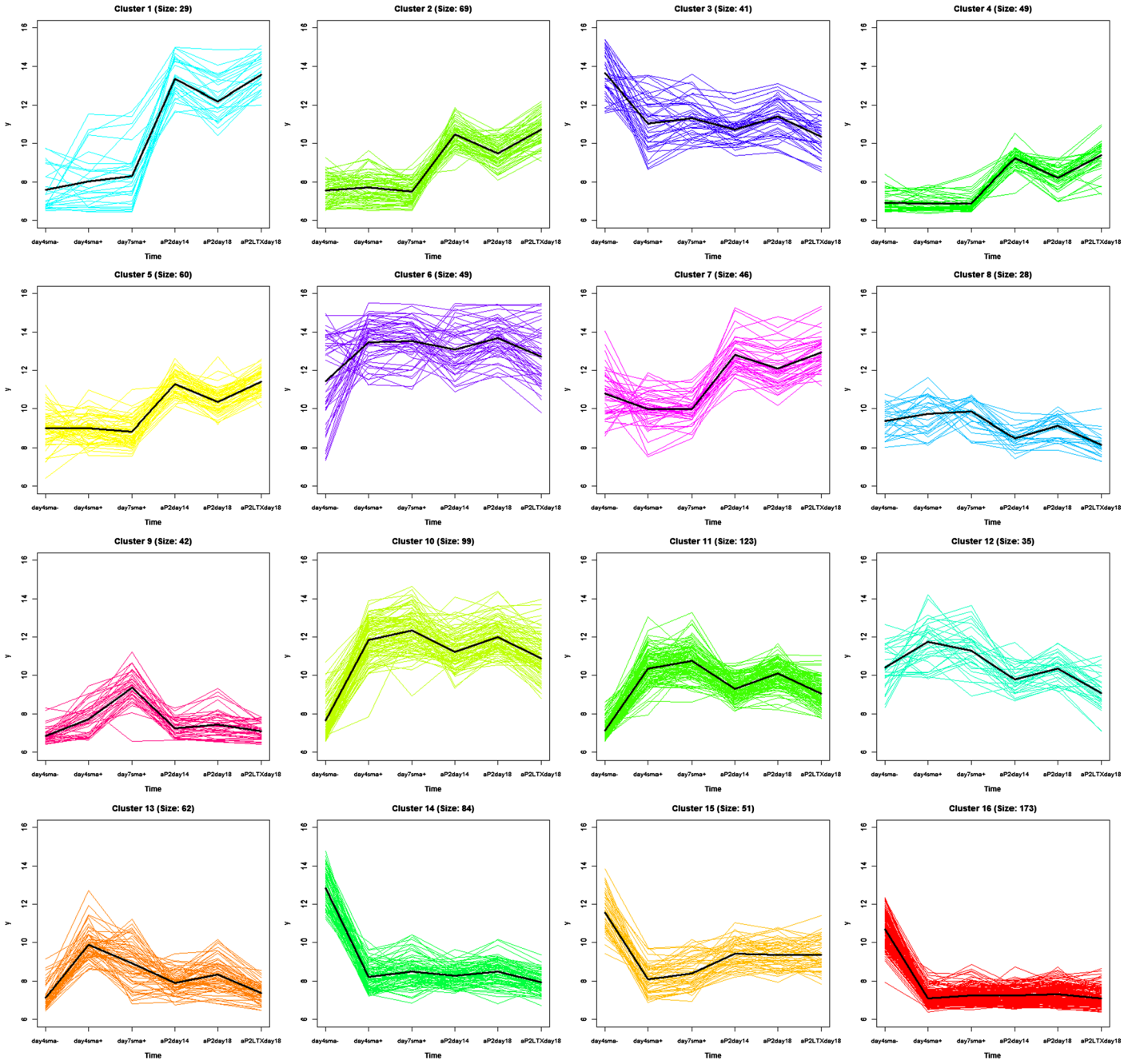

Figure 6. Plot of resulting clusters by the DP clustering for the mouse adipose data.

After clustering is done, one can look at such cluster plots, identify patterns of interest, and investigate the genes belonging to such clusters. The Gene Ontology (GO) project (http://www.geneontology.org) provides GO terms which are structured vocabularies and classifications of several molecular and cellular biological functions. GOstats analysis is a useful bioinformatic tool for scientists to assess the associations between GO terms and genes in a selected gene list. R Bioconductor package "GOstats" (Falcon and Gentleman (2007)) has been widely used to perform such computations. We have selected clusters 1, 11, 13, and 14 for biologists' interest. Table 4 lists the most significant GO terms for these four clusters. In this table, cellular component, biological process, and molecular function are abbreviated by CC, BP, and MF. Moreover, $\frac{\text{Count}}{\text{ClusterSize}}$ indicates the ratio of the number of genes in the cluster that belong to the particular GO term to the size of the cluster.

This GOstats result can provide guidance on which GO terms are associated with the regulation pattern of each particular cluster. From this table, we can observe that cluster

Table 4. GOstats results for selected DP clusters for the mouse adipose data

Cluster	Ontology	GO ID	GO Term	$\frac{\text{Count}}{\text{Cluster Size}}$	P-value
Cluster 1	BP	GO:0032787	monocarboxylic acid metabolic process	9/29	8.53E-06
	BP	GO:0006082	organic acid metabolic process	10/29	8.97E-06
	BP	GO:0019752	carboxylic acid metabolic process	10/29	8.97E-06
	BP	GO:0042180	cellular ketone metabolic process	10/29	8.97E-06
	BP	GO:0043436	oxoacid metabolic process	10/29	8.97E-06
	BP	GO:0044255	cellular lipid metabolic process	10/29	1.99E-05
	BP	GO:0006629	lipid metabolic process	11/29	6.70E-05
	BP	GO:0006631	fatty acid metabolic process	7/29	1.71E-04
	MF	GO:0016747	transferase activity, transferring acyl groups other than amino-acyl groups	4/29	2.12E-04
	MF	GO:0016746	transferase activity, transferring acyl groups	4/29	3.00E-04
Cluster 11	BP	GO:0048706	embryonic skeletal system development	9/123	3.47E-09
	BP	GO:0001501	skeletal system development	16/123	3.52E-08
	BP	GO:0009888	tissue development	29/123	4.93E-08
	BP	GO:0009792	embryo development ending in birth or egg hatching	17/123	1.89E-07
	BP	GO:0043009	chordate embryonic development	17/123	1.89E-07
	BP	GO:0051216	cartilage development	10/123	2.51E-07
	BP	GO:0048704	embryonic skeletal system morphogenesis	7/123	2.80E-07
	BP	GO:0007423	sensory organ development	12/123	5.14E-07
	BP	GO:0048568	embryonic organ development	13/123	7.12E-07
	BP	GO:0007507	heart development	14/123	8.26E-07
Cluster 13	BP	GO:0031175	neuron projection development	10/62	2.00E-05
	BP	GO:0048523	negative regulation of cellular process	26/62	4.51E-05
	BP	GO:0048666	neuron development	10/62	8.24E-05
	BP	GO:0010975	regulation of neuron projection development	7/62	8.43E-05
	BP	GO:0072089	stem cell proliferation	4/62	9.28E-05
	CC	GO:0000267	cell fraction	17/62	9.77E-05
	BP	GO:0060284	regulation of cell development	10/62	1.06E-04
	BP	GO:0002009	morphogenesis of an epithelium	9/62	1.40E-04
	BP	GO:0030030	cell projection organization	11/62	1.47E-04
	BP	GO:0048468	cell development	15/62	1.91E-04
Cluster 14	BP	GO:0002376	immune system process	38/84	9.29E-14
	BP	GO:0006955	immune response	27/84	1.49E-12
	BP	GO:0050778	positive regulation of immune response	16/84	3.06E-10
	BP	GO:0050776	regulation of immune response	18/84	4.12E-10
	BP	GO:0002682	regulation of immune system process	24/84	1.30E-09
	BP	GO:0006952	defense response	25/84	5.31E-09
	BP	GO:0002252	immune effector process	17/84	6.29E-09
	BP	GO:0002443	leukocyte mediated immunity	12/84	2.43E-08
	BP	GO:0051707	response to other organism	16/84	7.86E-08
	BP	GO:0002684	positive regulation of immune system process	17/84	8.66E-08

1 is more enriched with fat related functional terms, cluster 11 is more enriched with skeletal system and tissue development, cluster 13 is more enriched with neuron and cell development functions, and cluster 14 is a clear immune enriched cluster.

9. DISCUSSIONS

We develop a model-based Bayesian functional clustering approach using Dirichlet process to cluster genes with time-course microarray data. We construct a Dirichlet process on the unknown distribution of the basis coefficients of the spline fitting of the time course data. The hierarchical Dirichlet mixing process allows flexible nonparametric mixture modeling. The number of mixture components is not specified in advance and can grow as new data come in. In practice, we specify an upper limit on the number of clusters in order to construct the blocked Gibbs sampler. We take advantage of the clustering property of the Dirichlet process, which automatically assigns genes into appropriate number of clusters, hence it does not need to pre-specify the number of clusters like some other cluster algorithms. The Bayesian functional clustering algorithm is evaluated along with two widely used clustering procedures, K-means, MCLUST, and their two-stage version by adjusted Rand index. From our simulation study, we have shown that, our proposed method achieves higher adjusted Rand index values among all the procedures considered. In addition, the proposed algorithm is applied to a real dataset, and the clusters are interpreted by the GOstats analysis to facilitate the function analysis afterwards.

Gene clustering has a lot of important applications. For example, in Golub et al. (1999), cluster analysis is used to reveal tumor groups; newly discovered classes are compared with known classes. Ross and Perou (2001) use cDNA microarrays to study gene expression in the 60 cell lines from the National Cancer Institute's anticancer drug screen. Hierarchical clustering of the cell lines reveals a correspondence between gene expressions and tissues of the origin of tumors. In van't Veer et al. (2002), cluster analysis is used to investigate clinical outcomes of breast cancer and identify subsets of genes that show different expression patterns between different types of cancers. Our proposed functional clustering algorithm provides a flexible tool to cluster high-dimensional data for a wide range of experimental designs. Moreover, it can be applied in various areas such as drug discovery, disease diagnosis, cancer research, and personalized medicine.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. David Rowe and Dr. Dong-Guk Shin for sharing the data set used in the real data analysis. They further acknowledge a stem cell grant 06SCC04 from Connecticut Innovations, Connecticut, that provided partial support for the early phase of the first author's graduate study.

Received 1 November 2013

REFERENCES

Abraham, C., Cornillon, P. A., Matzner-Løber, E., and Molinari, N. (2003). Unsupervised curve clustering using B-splines. *Scandinavian Journal of Statistics*, 30(3):581–595. MR2002229

Angelini, C., De Canditiis, D., Mutarelli, M., and Pensky, M. (2007). A Bayesian approach to estimation and testing in time-course microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 6:24. MR2349917

Brennan, R. L. and Light, R. J. (1974). Measuring agreement when two observers classify people into categories not defined in advance. *British Journal of Mathematical and Statistical Psychology*, 27:154–163.

Dahl, D. B. (2006). Model-based clustering for expression data via a Dirichlet process mixture model. In Do, K.-A., Müller, P., and Vannucci, M., editors, *Bayesian Inference for Gene Expression and Proteomics*, pages 201–218. Cambridge University Press, New York.

de Boor, C. (1987). *A Practical Guide to Splines*. Springer, New York.

Du, P., Kibbe, W. A., and Lin, S. M. (2008). lumi: a pipeline for processing Illumina microarray. *Bioinformatics*, 24(13):1547–1554.

Dunson, D. B. (2010). Nonparametric Bayes applications to biostatistics. In Hjort, N. L., editor, *Bayesian Non-parametrics*, pages 223–270. Cambridge University Press, New York. MR2730665

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588. MR1340510

Everitt, B. S. (1993). *Cluster Analysis*. Edward Arnold, London. MR1217964

Falcon, S. and Gentleman, R. (2007). Using GOstats to test gene lists for GO term association. *Bioinformatics*, 23(2):257–265.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1:209–230. MR0350949

Fraley, C. and Raftery, A. E. (1999). MCLUST: software for model-based cluster analysis. *Journal of Classification*, 16:297–306.

Fraley, C. and Raftery, A. E. (2000). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97:611–631. MR1951635

Fu, A. Q., Russell, S., Bray, S. J., and Tavaré, S. (2013). Bayesian clustering of replicated time-course gene expression data with weak signals. *Annals of Applied Statistics*, in press.

Golub, T. R., Slonim, D. K., Tamayo1, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

Ishwaran, H. and James, L. F. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 101:179–194. MR1952729

Kim, S., Tadesse, M. G., and Vannucci, M. (2006). Variable selection in clustering via Dirichlet process mixture models. *Biometrika*, 93:877–893. MR2285077

Luan, Y. and Li, H. (2003). Clustering of time-course gene expression data using a mixed-effects model with B-splines. *Bioinformatics*, 19:474–482.

Ma, P., Castillo-Davis, C. I., Zhong, W., and Liu, J. S. (2006). A data-driven clustering method for time course gene expression data. *Nucleic Acids Research*, 34(4):1261–1269.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, Berkeley. MR0214227

Medvedovic, M. and Sivaganesan, S. (2002). Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics*, 18:1194–1206.

Qin, Z. S. (2006). Clustering microarray gene expression data using weighted chinese restaurant process. *Bioinformatics*, 22:1988–1997.

Ramsay, J. O. and Silverman, B. W. (1997). *Functional Data Analysis*. Springer, New York. MR2168993

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850.

Ray, S. and Mallick, B. (2006). Functional clustering by Byesian wavelet methods. *Journal of the Royal Statistical Society*, B, 68, Part 2:305–332. MR2188987

Ross, D. T. and Perou, C. M. (2001). A comparison of gene expression signatures from breast tumors and breast tissue derived cell lines. *Disease Markers*, 17(2):99–109.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650. MR1309433

Sethuraman, J. and Tiwari, R. C. (1982). Convergence of Dirichlet measures and the interpretation of their parameter. In Gupta, S. S. and Berger, J. O., editors, *Statistical Decision Theory and Related Topics III*, volume 2, pages 305–315. Academic Press, New York, 2nd edition. MR0705321

Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3:Article 3. MR2101454

Song, J. J., Lee, H. J., Morris, J. S., and Kang, S. (2007). Clustering of time-course gene expression data using functional data analysis. *Computational Biology and Chemistry*, 31(4):265–274.

Storey, J. D., Xiao, W., Leek, J. T., Tompkins, R. G., and Davis, R. W. (2005). Significance analysis of time course microarray experiments. *Bioinformatics*, 21(1):71–79.

Tai, Y. C. and Speed, T. P. (2006). A multivariate empirical Bayes statistic for replicated microarray time course data. *Annals of Statistics*, 34(5):2387–2412. MR2291504

van't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. H. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–535.

Wu, R., Chen, M.-H., Kuo, L., and Lewis, P. O. (2014). A new method for tracking configuration for Dirichlet process sampling. *Sri Lankan Journal of Applied Statistics*, in press.

Ziwen Wei
126 E. Lincoln Ave
Rahway, NJ 07065
USA
E-mail address: ziwen.wei@merck.com

Lynn Kuo
Department of Statistics
University of Connecticut
215 Glenbrook Road
Storrs, CT 06269
USA
E-mail address: lynn.kuo@uconn.edu