Published in final edited form as:
J Bioinform Comput Biol. 2008 August ; 6(4): 727–746.

KULLBACK-LEIBLER MARKOV CHAIN MONTE CARLO — A NEW ALGORITHM FOR FINITE MIXTURE ANALYSIS AND ITS APPLICATION TO GENE EXPRESSION DATA

TATIANA TATARINOVA
Department of Mathematics Loyola Marymount University Los Angeles, CA 90045, USA tatiana.tatarinova@lmu.edu

JOHN BOUCK
Ceres, Inc. 1535 Rancho Conejo Road Thousand Oaks, CA 91320, USA jbouck@ceres-inc.com

ALAN SCHUMITZKY
Department of Mathematics University of Southern California Los Angeles, CA 90089, USA schum@usc.edu

Abstract

In this paper, we study Bayesian analysis of nonlinear hierarchical mixture models with a finite but unknown number of components. Our approach is based on Markov chain Monte Carlo (MCMC) methods. One of the applications of our method is directed to the clustering problem in gene expression analysis. From a mathematical and statistical point of view, we discuss the following topics: theoretical and practical convergence problems of the MCMC method; determination of the number of components in the mixture; and computational problems associated with likelihood calculations. In the existing literature, these problems have mainly been addressed in the linear case. One of the main contributions of this paper is developing a method for the nonlinear case. Our approach is based on a combination of methods including Gibbs sampling, random permutation sampling, birth-death MCMC, and Kullback-Leibler distance.

Keywords

Gibbs sampling; permutation sampling; birth-death MCMC; Kullback-Leibler distance; gene expression; clustering; nonlinear mixture models

1. Motivation

A number of methods have been developed for clustering gene expression data. The choice of method depends mainly on the data representation. Gene expression data can be either in the form of the vector of measured intensities (or ratios) or in the relational form (i.e. as correlation coefficients between pairs of genes). We would like to suggest a new Bayesian method for clustering of data-rich time-series observations. Experimental points of the time series observations are ordered, and this fact sets them apart from other microarray measurements. Our approach assumes that every cluster can be described by a smooth centroid curve. If N genes can be grouped into K groups by their expression profile, then all genes that belong to a group $k = 1, 2, \ldots, K$ have similar values of the "trajectory" parameters Θ, where Θ is an n-dimensional vector. For every gene $i = 1, 2, \ldots, N$, values of Θ_i can be found by analyzing the following hierarchical model:

$$y_{ij} \tilde{} Normal\left(f\left(\Theta_i, t_j\right), \sigma_e^2\right) \quad \text{for} \quad i \in [1, N] \quad \text{and} \quad j \in [1, T], \tag{1}$$

where N is the number of genes, j is an index of an experiment, T is the total number of experiments, and σ_e is the experimental error. The expression of a gene i at time t_j is given by some, probably nonlinear, function $f(\Theta_i, t_j)$. Gene-specific parameters Θ_i are assumed to have a mixture distribution

$$\Theta_i \tilde{} \sum_{K}^{k=1} w_k Normal\left(\mu_k, \Sigma_k\right), \tag{2}$$

where *Normal* (μ, Σ) is a the multivariate normal distribution with mean μ and covariance Σ. Component weights are assumed to have Dirichlet distribution $w_k \sim Dirichlet(\alpha)$. Other parameters were given appropriate prior distributions.

The described model is a general nonlinear mixture model, and we propose to use a Bayesian approach. Our goal is to calculate the posterior density $\pi(q|Y)$, where $q = (\mu_1, \mu_2, \ldots, \mu_K, \Sigma_1, \Sigma_2, \ldots, \Sigma_K, w_1, w_2, \ldots, w_K, K)$ is the vector of unknown parameters and $Y = \{y_{ij}, i = 1, 2, \ldots, N; j = 1, 2, \ldots, T\}$ represents the data. For a known number of components K, the method we propose in this paper is a combination of Gibbs and Metropolis sampling. The theoretical convergence of the resulting Markov chain was studied by Müller[1] and Tatarinova.[2]

In practice, the resulting Markov chain can get computationally trapped in a local region of the posterior surface and cannot escape. Actually, the probability of escape is so small that it is below the round-off error on the computer. The trapping of a Markov chain is a serious problem, much worse than that of local convergence of, say, an expectation-maximization (EM)-type algorithm. For the local convergence problem, a sufficient number of starting positions can lead to a global solution. For the Gibbs trapping problem, if the true posterior is multi-modal, no starting position will in general lead to the true solution. This trapping problem led to the observation in Celeux *et al.*[3] that "almost the entirety of the Gibbs Markov chain Monte Carlo samples implemented for mixture models has failed to converge!" (See also Fruhwirth-Schnatter[4] for further discussion.)

We have previously shown[2] that, in order to fix the problem of "trapping", the computed posterior distribution should be symmetric relative to all components of the mixture. Consequently, an observation will be equally likely to come from any component. This nonidentifiability is a fundamental property of mixture models (see Stephens[5-8]). To identify parameters uniquely, the clustering must then be done outside of the Markov chain Monte Carlo (MCMC) method.

2. Theoretical Background

In the framework of mixture models, the clustering problem is equivalent to the problem of determining which mixture component an observation is most likely to come from. Our approach is a combination of four previously developed methods:

1. birth-death MCMC approach by Stephens[5,7];
2. random permutation sampler (RPS) by Fruhwirth-Schnatter[4];
3. choosing the optimal number of components using the weighted Kullback- Leibler distance by Sahu and Cheng[9]; and

4. relabelling strategy by Stephens.[8]

Below we provide a brief description of these methods.

2.1. Markov chain Monte Carlo and the Gibbs sampler

Markov chain Monte Carlo (MCMC) methods were developed to facilitate the calculation of posterior distributions. The MCMC approach got its name because one uses the previous sample values to randomly generate the next sample value, thus producing a Markov chain. Calculation of the posterior distribution involves evaluation of complex multi-dimensional integrals; an efficient way to deal with this integration is by employing Monte Carlo approximation. In this framework, the integral $\int f(y|\chi)\pi(\chi)d\chi$ is approximated by $\frac{\sum_{i=1}^{n} f(y|x_i)}{n}$.

The Gibbs sampler is a Bayesian method that replaces the problem of calculating the posterior distribution with iteratively sampling from the full conditional distributions (see Casella and George[10] for an informative discussion). Along with Metropolis-Hastings, it is perhaps the most popular MCMC method. Since Gibbs sampling is a Bayesian method, prior distributions for the model parameters are necessarily needed.

2.2. BDMCMC

The birth-and-death Markov chain Monte Carlo (BDMCMC) method was developed by Matthew Stephens.[5-8] Under the BDMCMC methodology, a number of components of the mixture change dynamically: new components are created (birth) or an existing one is deleted (death), and model parameters are then recomputed. The posterior probability distribution for the number of components is estimated as

$$P(K=k_0|data)=\frac{\#\left\{t:K^{(t)}=k_0\right\}}{T}, \tag{3}$$

where $K^{(t)}$ is the number of components at the iteration t, and T is the total number of iterations. Following Stephens,[5] we assume, for fixed σ_e, that birth and death occur as independent Poisson processes with rates $\beta(Q)$and $\delta(Q)$, where the rates depend on the current state of Q in the process. Probabilities of birth and death are

$$\begin{aligned} P_b &= \frac{\beta(Q)}{\beta(Q)+\delta(Q)} \\ P_d &= \frac{\delta(Q)}{\beta(Q)+\delta(Q)}. \end{aligned} \tag{4}$$

If the birth of a component with (w, φ) occurs, then the process "jumps" from the K-component state $Q=\{(w_1,\phi_1),(w_2,\phi_2),\cdots,(w_K,\phi_K)\}$, where $\varphi_k = (\mu_k, \Sigma_k)$, to the $K + 1$ component state $Q\bigcup(w,\phi)=\{(w_1(1-w),\phi_1),(w_2(1-w),\phi_2),\cdots,(w_k(1-w),\phi_k),(w,\phi)\}$. If the death of the k_0th component occurs, then the process "jumps" from the K-component state $Q=\{(w_1,\phi_1)(w_2,\phi_2),\cdots,(w_K,\phi_K)\}$, to the K - 1 component state

$$Q\setminus(w_{k_0},\phi_{k_0})=\left\{\left(\frac{w_1}{1-w_{k_0}},\phi_1\right),\cdots,\left(\frac{w_{k_0}-1}{1-w_{k_0}}\phi_{k_0-1}\right),\left(\frac{w_{k_0+1}}{1-w_{k_0}},\phi_{k_0+1}\right),\cdots,\left(\frac{w_K}{1-w_{k_0}},\phi_K\right)\right\}.$$

Note that birth and death moves do not violate the constraint $\Sigma_k w_k = 1$. As in previous works by Zhou[11] and Stephens,[5] we assume a Poisson prior on the number of components K:

$$P(K)=\frac{\lambda^K e^{-\lambda}}{K!}. \tag{5}$$

The likelihood $L(Q)$ is defined as $\prod_{i=1}^{n} P(Y_i|Q)$, where $Y_i=(y_{i1}, y_{i2}, \ldots, y_{iT})$. If we fix the birth rate $\beta(Q)=\lambda_b$, the death rate for each component $k \in \{1, 2, \ldots, K\}$ is calculated as

$$\delta_k(Q)=\lambda_b \frac{L(Q \setminus (w_k,\phi_k))}{L(Q)} \frac{P(K)}{(K+1)P(K+1)}=\frac{L(Q) \setminus (w_k,\phi_k)}{L(Q)} \frac{\lambda_b}{\lambda}, \tag{6}$$

and the total death rate at state Q is $\delta(Q) = \Sigma_k \delta_k(Q)$.

The birth-death MCMC sampler is constructed using Algorithm 4.2 from Stephens[5]:

1. Run the Gibbs sampler for n iterations. Update Q and σ_e.
2. Based on the Gibbs sampler output, compute the death rate and the probabilities of birth P_b and death P_d [Eq. (4)].
3. Simulate the birth-death process. Generate a random number $u \sim Uniform[0, 1]$.
4. If $P_b > u$ and $P_b > P$, simulate a point of birth (w_{K+1}, ϕ_{K+1}).
   a. If $P_d > u$ and $P_d > P_b$, choose a component to die with probability $\frac{\delta_k(Q)}{\delta(Q)}$. This is achieved by computing the cumulative death probability $F(k)= P$ (to kill a component with an index $\leq k$) and generating a random number $u \sim Uniform[0, 1]$. The component k_0 is chosen to die if $F(k_0 - 1) < u \leq F(k_0)$.
   b. Adjust prior distributions for the Gibbs sampler.
5. Repeat steps (1)-(4).

2.3. Random permutation sampler

Computational difficulties associated with the MCMC estimation of mixture models motivated Fruhwirth-Schnatter[4] to develop the random permutation sampler (RPS) method. She has made a simple but brilliant proposal to facilitate convergence of MCMC samplers for mixture models: conclude each sweep with "a random permutation of the current labeling of the states." Since permutation invariance is a necessary condition for convergence, forcing permutations between the sweeps of the Gibbs sampler cannot violate the properties of the algorithm. RPS works essentially as follows. Let the integer-valued allocation variables $\{Z_i\}$ be defined such that $P(Z_i = k) = w_k$, $k = 1, 2, \ldots, K$, $i = 1, 2, \ldots, N$ and $p(\theta_i|q,Z_i)=p\left(\theta|\mu_{z_i},\Sigma_{z_i}\right)$. After each cycle of the Gibbs/Metropolis sampler, the "labels" on the components $\{Z_i\}$ are randomly permuted. More precisely, after each cycle, the current vectors $\{Z_k, w_k, \mu_k, \Sigma_k\}$ are replaced by $\{\nu_{v(k)}, w_{v(k)}, \mu_{v(k)}, \Sigma_{v(k)}\}$, where $\{v(1), v(2), \ldots, v(K)\}$ is a random permutation of the set $\{1, 2, \ldots, K\}$. As was shown by Fruhwirth-Schnatter,[4] if the original MCMC algorithm is convergent, then the RPS algorithm is also convergent, and the target densities of RPS and the original MCMC are identical. In addition to preserving the convergence properties, RPS enhances the mixing of MCMC. See Fruhwirth-Schnatter[4] for more details.

2.4. Weighted Kullback-Leibler distance

In the framework of the weighted Kullback-Leibler distance method developed by Sahu and Cheng,[9] the optimal number of mixture components is defined as the smallest number of

components that adequately explains the structure of the data. This task is traditionally accomplished by the reductive stepwise method. In the framework of this method, one starts with the $K = K_0$ component mixture which adequately explains the observed data. Next, K is iteratively reduced until the fit is no longer adequate. Reduction of the number of components is achieved by collapsing two of the K components. Adequacy of the fit is assessed using the distance between probability distributions at two consecutive steps. There are several ways to define distance d between probability density functions. One of them, the Kullback-Leibler distance between two probability densities f and g, is defined as

$$d(f,g) = \int_{S(f)} f(\chi) \log\left(\frac{f(\chi)}{g(\chi)}\right) \mathrm{d}x, \tag{7}$$

where $S(f)$ is the support of density f, i.e. $S(f) = \{x: f(\chi) > 0\}$. Unfortunately, in general, the Kullback-Leibler distance cannot be evaluated analytically.

Consider two mixture densities $f^{(K)}(\chi) = \Sigma_{j=1}^{K} w_j f_j\left(\chi|\varphi_j\right)$ and the mixture distribution $g^{(K)}(\chi) = \Sigma_{j=1}^{K} w_j g_j\left(\chi|\varphi_j^*\right)$, which have the same component weights w_j as $f^{(K)}$. Then, Sahu and Cheng[9] suggested using the weighted Kullback-Leibler distance, defined as

$$d^* = \sum_{K}^{j=1} w_j d\left(f_j, g_j\right). \tag{8}$$

The weighted Kullback-Leibler distance $d^*(f^{(K)}, g^{(K)})$ is defined as a weighted sum of the Kullback-Leibler distances $d(f_j, g_j)$ between corresponding component densities. If $f_{ij}^{*(K-1)} = \Sigma_{j=1}^{K} w_j f_j\left(\chi|\varphi_j^*\right)$ is a collapsed version of $f^{(K)}$, where $\varphi_i^* = \varphi_j^* = \varphi^*$ and other K - 2 components are unchanged, then

$$\begin{aligned} d^*\left(f^{(K)}, f_{(ij)}^{*(K-1)}\right) &= w_i d\left(f_i(\cdot|\varphi_i), f_i(\cdot|\varphi^*)\right) + w_j d\left(f_j\left(\cdot|\varphi_j\right), f_j(\cdot|\varphi^*)\right) \\ &= w_i E_i \ln\left(\frac{f_i(\cdot|\varphi_i)}{f_i(\cdot|\varphi^*)}\right) + w_j E_j \ln\left(\frac{f_j(\cdot|\varphi_j)}{f_j(\cdot|\varphi^*)}\right), \end{aligned} \tag{9}$$

where E_i is an expectation under $f(\chi|\varphi_i)$ and φ^* minimizes $d^*\left(f^{(K)}, f_{(ij)}^{*(K-1)}\right)$:

$$\varphi^* = \arg\left\{\min_{\varphi^*} \sum_{k=i,j} w_k E_k \inf_k(\cdot|\varphi_k)\right\}. \tag{10}$$

For the normal mixture models we are considering in this paper, the above expectations and minimizations can be done analytically.[2,9] The best collapsed version denoted by $f^{*(K-1)}$ is the one that minimizes $d^*\left(f^{(K)}, f_{(ij)}^{*(K-1)}\right)$ over all i, j such as $i \neq j$. Notice that the minimum weighted Kullback-Leibler distance is invariant under permutations of $\mathbf{w}$ and φ. The K-component model can be replaced by the (K - 1)-component model if the distance between the best collapsed version and the original model is less than some cut-off c. The choice of c depends on the data structure and type of the distribution used.

2.5. Relabeling

In order to obtain the accurate parameter estimation from $K!$-modal distributions for the K-component mixture, Matthew Stephens[8] presented relabeling strategies based on minimizing the posterior expectation of a loss function. Function $L(a; \theta)$ is a loss due to some action a, when the true value of a model parameter is θ. For instance, in the case of linear normal mixtures, we need to choose a relabeling that makes the posterior distribution of component weights and means look independent and normal. Relabeling is the final postprocessing step, performed to obtain cluster memberships and cluster parameters. For our applications, we use the relabeling algorithm S2 of Stephens,[8] developed for the special case of normal mixture models.

3. Kullback-Leibler Markov Chain Monte Carlo (KLMCMC)

When we decided to develop our new transdimensional method, we were motivated by several factors. The BDMCMC method calls for repeated evaluation of the likelihood. When the number of observations is large, the likelihood can become very small, resulting in singularities in the death rate. For nonlinear models, evaluation of the likelihood involves integration, which can negatively affect the performance of the algorithm. One of the most important properties of the Gibbs sampler is that calculating the full conditionals does not require calculating the likelihood. This property is essential for nonlinear pharmacokinetic models, where calculation of the likelihood requires high-dimensional integration over the model parameters. Since this integration is required at essentially every iteration of the Gibbs sampler, the resulting algorithm would be unacceptably slow. On the other hand, for linear models the likelihood can be evaluated without integration and the problem disappears. (See Lunn *et al.*[12] for an excellent discussion of this issue.)

Computation of the weighted Kullback-Leibler distance is straightforward for some popular families of distributions (i.e. normal and gamma); its complexity does not depend on the number of observations; and nonlinear models can be handled as efficiently as the linear ones. The weighted Kullback-Leibler distance is a natural measure of closeness between $(K + 1)$- and K-component models (see Sahu and Chang[9] for discussion). The probability distribution of weighted Kullback-Leibler distance between $(K + 1)$- and K-component models characterizes the current state of the Gibbs sampler. In Bayesian statistics, the weighted Kullback-Leibler distance is sometimes used as a measure of the information gain in moving from a prior distribution to a posterior distribution.[13,14]

We suggest to compute the weighted Kullback-Leibler distance between all pairs of components, and to record the smallest of all distances $d^*(k)$ for all components. A large value of $d^*(k)$ indicates that the kth component differs from all other components and the probability to "kill" this component must be small. Inversely, we need to assign large death rates to components that do not differ significantly from other components and hence have small values of $d^*(k)$. Thus, when the Gibbs sampler is at state Q, the death rate for the kth component $\delta_k(Q)$ is assumed to be the inverse of $d^*(k)$. The total death rate $\delta(Q)$ is equal to $\Sigma_{k=1}^{K} \frac{1}{d^*(k)}$. Hence, probability of the death move is $p_d = \frac{\delta(Q)}{\lambda_B + \delta(Q)}$, where λ_B is assumed to be equal to the expected number of components; and probability of the birth move is $p_b = \frac{\lambda_B}{\lambda_B + \delta(Q)}$.

The mixing and convergence of the MCMC can be significantly improved by using the random permutation approach suggested by Sylvia Fruhwirth-Schnatter.[4] At each birth-death step, components of the mixture are randomly permuted, producing a symmetric posterior

distribution of estimated model parameters. The resulting KLMCMC algorithm will be as follows:

1. Run the Gibbs sampler *n* iterations.
2. Based on the Gibbs sampler output, compute the death rate $\delta_k(Q) = 1/d^*(k)$ for individual components, the total death rate $\delta(Q) = \Sigma_{k=1}\delta_k(Q)$, and the probabilities of birth $p_b = \frac{\lambda_B}{\lambda_B + \delta(Q)}$ and death $p_d = \frac{\lambda(Q)}{\lambda_B + \delta(Q)}$.
3. Simulate the birth-death process. Generate a random number $u \sim Uniform\,[0, 1]$.
   a. If $P_b > u$, simulate a point of birth (w_{K+1}, μ_{K+1}) as $w_{K+1} \sim B(1, K+1)$, $\mu_{K+1} \sim Normal\left(\mu_0, \sigma_0^2\right)$. Values of μ_0 and σ_0^2 are estimated as the mean and variance of the distribution of observations, respectively. Weights of the first *K* components should be adjusted as $w_1(1 - w_{K+1})$, $w_2(1 - w_{K+1})$, …, $w_K(1 - w_{K+1})$.
   b. Else, choose a component to die with probability $\frac{\delta_k(Q)}{\delta(Q)}$. This is achieved by computing the cumulative death probability $F(k) = P$ (to kill a component with an index $\leq k$) and generating a random number $u \sim Uniform[0, 1]$. The component k_0 is chosen to die if $F(k_0 - 1) < u \leq F(k_0)$. Weights of the remaining $K - 1$ components should be adjusted as $\frac{w_1}{(1 - w_K)}, \frac{w_2}{(1 - w_K)}, \ldots, \frac{w_{K-1}}{(1 - w_K)}$.
4. Adjust prior distributions for the Gibbs sampler.
5. RPS step. Randomly permute components of the mixture.
6. Repeat steps (1)-(5).

In the next sections, we test the performance of the KLMCMC algorithm on two well-studied time-series datasets: the simulated time series data analyzed by Zhou,[11] and the yeast sporulation dataset studied by Chu *et al.*[15] and Wakefield *et al.*[16]

4. Clustering of Simulated Time Series Data with an Unknown Number of Clusters

We applied our method of choosing the optimal number of components to the simulated dataset suggested by Zhou[11] and compared the performance of KLMCMC and traditional Gibbs sampler, implemented as part of the WinBUGS package. Following Zhou's approach, we generated 50 linear time curves from $K = 3$ clusters:

$$y_{ij} \sim Normal\left(\alpha_i + \beta_i t_j, \sigma_e^2\right), \quad t_j = (-3, -1, 1, 3), \tag{11}$$

where α_i is the intercept and β_i is the slope for a curve *i*. For each generated curve $i \in [1, 50]$, intercept and slope were assumed to arise from the multivariate normal distribution

$$\theta_i = \begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} \sim Normal\left(\mu_{zi}, \Sigma_{zi}\right), \tag{12}$$

where $z_i \in \{1, 2, 3\}$ are cluster membership labels such that $P[z_i = k] = 1/3$.

Parameters of the clusters, namely, cluster centers and covariance matrices, for the three clusters were

$$\begin{aligned}\mu_1&=\begin{pmatrix}8\\1\end{pmatrix},\quad \Sigma_1=\begin{pmatrix}0.1&0\\0&0.1\end{pmatrix},\\ \mu_2&=\begin{pmatrix}10\\0\end{pmatrix},\quad \Sigma_3=\begin{pmatrix}0.1&0\\0&0.1\end{pmatrix},\\ \mu_3&=\begin{pmatrix}12\\-1\end{pmatrix},\quad \Sigma_3=\begin{pmatrix}0.2&0\\0&0.2\end{pmatrix}.\end{aligned} \tag{13}$$

Following Zhou,[11] we chose $\sigma_e = 1.0$. In this section, we will refer to the model below as the Zhou model. Simulated data are shown in Fig. 1.

4.1. Model description

1. At the first stage, the model has the form $y_{ij} \sim Normal\left(\alpha_i+\beta_i t_j, \sigma_e^2\right)$ for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, T$.
2. At the second stage, the model describes intercept and slope of individual curves:

$$(\alpha_i,\beta_i)^T \equiv \Theta_i \sim Normal\,(\mu_{z_k},\Sigma_{z_k}), \quad \text{where} \quad k=1,2,\cdots,K.$$

3. At the third stage, we assume that cluster membership labels z follow a categorical distribution $P[z_i = k] = w_k = p_k$, $k = 1, 2, \ldots, K$, such as $\Sigma_k p_k = 1$.
4. At the fourth stage, we specify parameters:

$$\begin{aligned}&(p_1,p_2,\cdots,p_K)\sim \mathcal{D}\,(1_1,\cdots,1)\\ &\mu_k \sim Normal\,(\eta,\mathrm{C})\\ &\Sigma^{-1}_{k\ ij,i=j} \sim A\,(k)\,Normal\,(a,b)\\ &\Sigma^{-1}_{k\ ij,i\neq j}=0,\end{aligned} \tag{14}$$

   where $A(k)$ is a set of cluster-specific constant parameters.
5. At the fifth stage, the model specifies the distribution of the experimental error:

$$\sigma_e^{-2} \sim \Gamma\,(\lambda,\nu).$$

Using the least squares estimation, we calculated weakly informative priors for η and $C{:}\eta = (10.38, -0.24), C=\begin{pmatrix}0.04&0\\0&0.24\end{pmatrix}$. Using WinBUGS with $K = 3$, we let the Gibbs sampler run for 200,000 iterations, discarding the first 100,000 burn-in iterations. For comparison, we analyzed the same model with KLMCMC with and without the random permutation step. We set parameter $\lambda = \lambda_B = 3$ and ran the simulation for 300 hybrid birth-death steps, each containing 5,000 WinBUGS iterations. Distribution of the number of model components K, shown in Fig. 2, attains its maximum at $K = 3$. The Stephens[8] relabeling method was applied to the output of the Gibbs sampler ($p = (p_1, p_2, p_3)$; μ; and σ_e) for the three-component model.

The results for the parameters p, μ, and σ_e are essentially the same for all three methods. All three programs have correctly assigned cluster memberships. On the other hand, the results for

the covariance matrix Σ are considerably improved, as shown in Table 1. This can be explained by the lack of convergence due to the stickiness of the Gibbs sampler. Transdimensional methods, like KLMCMC, can overcome this problem. Better results can be achieved when randomization is performed after each round of Gibbs sampler in the KLMCMC method.

For comparison, we applied traditional clustering methods to this dataset: hierarchical clustering[17] and *K*-means clustering.[18,19] The best of the hierarchical clustering, centroid linkage with city-block distance as a measure of similarity between curves, has correctly assigned 90% of curves to their original clusters; centroid linkage with Pearson correlation coefficient has succeeded in 84% of cases. Conditional on three clusters, *K*-means clustering with the Pearson correlation coefficient as a measure of similarity was able to correctly recover 74% of cluster memberships; *K*-means clustering with the Spearman rank correlation coefficient recovered 78% of cluster memberships; and 96% of cluster membership was recovered with city-block distance as a similarity measure. We believe that, in the case of noisy time series data, valuable clustering information can frequently be recovered only if model-based clustering methods are employed.

5. Brief Review of Clustering Methods in Microarray Analysis

Gene clustering analysis is important for finding groups of correlated (potentially coregulated) genes. A number of clustering methods have been developed specifically with gene expression in mind, in addition to numerous methods adapted from other disciplines. These methods include the following popular approaches: hierarchical clustering,[17] self-organizing maps,[20,21] *K*-means clustering,[18,19] principal component analysis (PCA),[22,23] singular value decomposition (SVD),[24] partitioning around medioids (PAM),[25] model-based clustering,[26-31] tight clustering,[32] and curve clustering.[16]

As was recently demonstrated by Thalamuthu *et al.*,[33] model-based and tight clustering algorithms “consistently overperform other clustering methods.” Tight clustering and model-based clustering make a provision for the existence of a noise set of genes, unaffected by the biological process under investigation. This noise set of genes is not clustered, and thus false-positive cluster members that almost inevitably appeared in traditional methods do not distort the structure of identified clusters. Another improvement of the model-based clustering methods is utilization of a biological model. Traditional measures of similarity between genes, such as correlation coefficients, Euclidean distance, Kendal's tau, and city-block distance, report a single number for a pair of genes, not utilizing the data-rich nature of time series microarray experiments. Model-based clustering methods (e.g. Wakefield *et al.*,[16] Yeung *et al.*,[26] Fraley and Raftery,[27,28] Medvedovic *et al.*,[30,31] McLachlan *et al.*,[29] etc.) are able to take advantage of this situation. The widely used program MCLUST, developed by Fraley and Raftery, provides an iterative expectation-maximization (EM) method for maximum likelihood clustering for parameterized Gaussian mixture models. According to Fraley and Raftery, each gene is represented by a vector in a space of arbitrary dimensions *d*, and interaction between genes is given in the form of a *d*-dimensional covariance matrix. In a more recent publication, Wakefield *et al.*[16] implemented a curve-clustering approach and used a Bayesian strategy to partition genes into clusters and find parameters of mean cluster curves. One crucial difference between the approach of Wakefield *et al.* and the previously developed methods is that neither Yeung[26] nor Fraley and Raftery[27,28] acknowledged the time-ordering of the data.

As was pointed out by Thalamuthu *et al.*,[33] model-based clustering methods enjoy “full probabilistic modeling” and the selection of the number of clusters is statistically justified. Due to the complexity of large-scale probabilistic modeling, they suggest that model-based clustering can be successfully performed as higher-order machinery that can be built upon

traditional clustering methods. In the next section, we discuss KLMCMC as one of the possible higher-order clustering methods.

6. Application of KLMCMC to Gene Expression Time Series Analysis

A temporal program of sporulation in budding yeast was studied by Chu *et al.*[15] The dataset consists of seven successive time points (t = 0, 0.5, 2, 5, 7, 9, 11.5 hours). We use the prescreening strategy of Wakefield *et al.*[16] and use the first time point t = 0 to estimate measurement errors. According to the analysis of Chu *et al.*[15] and the false discovery rate filter by Zhou[11] and Wakefield *et al.*,[16] approximately 1,300 of 6,118 genes showed significant changes in mRNA level in the course of the experiment. Wakefield *et al.*[16] assumed a first-order random walk model for gene-specific trajectories. Chu *et al.*[15] identified seven "characteristic profiles" and grouped genes by similarity to these profiles (Fig. 3):

1. Metabolic: induced rapidly and transiently after transfer to sporulation medium.
2. Early (I): detectable after 0.5-hour transfer to sporulation medium and sustained expression through the rest of the time course; role in recombination and chromosome pairing.
3. Early (II): delayed increase in transcription level.
4. Early-middle: initially induced 2 hours after the transfer to sporulation medium and, in addition, at 5 and 7 hours.
5. Middle: expressed between 2 and 5 hours.
6. Mid-late: from 5 to 7 hours; involved in meiotic division.
7. Late: induced between 7 and 11.5 hours; spore wall maturation.

For our analysis, we assume that the temporal patterns of the studied genes can be described by a function $f\left(\Theta_i,t_j\right)=P_i\left(t_j\right)e^{-\delta_i t_j}\eta\left(t_j-\gamma_i\right)$, where $\Theta_i=\{\alpha_i,\beta_i,\delta_i,\gamma_i\}$ is a set of gene-specific parameters and $P_i\left(t_j\right)=\alpha_i+\beta_i t_j$ is a firstdegree polynomial. In order to represent time γ_i when gene i is "turned on", we introduce the step function $\eta\left(t_j-\gamma_i\right)=\begin{cases}0, & \text{if } t_j<\gamma_i\\ 1, & \text{if } t_j\geq\gamma_i.\end{cases}$ Observations are modeled as $y_{ij}\tilde{}Normal\left(f\left(\Theta_i,t_j\right),\sigma_e^2\right)$ for $i\in[1,N]$ and $j\in[1,T]$, where N = 36 is the number of genes, T = 6 is the number of experiments, and $\sigma_e^{-2}\tilde{}G(g,h)$ is the experimental error (with values of g and h estimated from the T = 0 observations). Gene-specific parameters $\{\alpha_i, \beta_i, \gamma_i\}$ are assumed to have a multivariate normal mixture distribution

$$\{\alpha_i,\beta_i,\delta_i\}\tilde{}\sum_{k=1}^{K}w_k\quad Normal\left(\mu_k,\Sigma_k\right), \tag{15}$$

and the "turn-on time" $\gamma_i\tilde{}\sum_{k=1}^{K}Uniform[0, 12]$.

Component weights are assumed to have Dirichlet distribution w_kk ~ *Dirichlet* (α). Other parameters were given appropriate prior distributions. Least squares estimates of gene-specific parameters Θ_i are used to obtain parameters μ_0, Σ_0 of weakly informative priors:

$$\mu_k\tilde{}Normal\left(\mu_0,\Sigma_0\right). \tag{16}$$

To simplify the computation, we assumed that the covariance matrices Σ_k and Σ_0 are diagonal: $\left[\Sigma_M^{-1}\right]_{ll}\Gamma(a_l,b_l)$, for $l \in [1, 3]$, where parameters a_l, b_l of gamma distributions $\Gamma(a_l, b_l)$ are the least squares estimates.

The dimension-changing step consists of three parts:

1. Death rate is estimated as $\delta(Q) = \Sigma_{k=1} 1/d^*(k)$ (3), where d^* is the smallest weighted Kullback-Leibler distance. It is computed analytically in the case of normal distributions and evaluated by Gibbs sampler based on model parameters. Birth or death move is chosen with probabilities $p_b = \frac{\lambda_B}{\lambda_B + \delta(Q)}$ and $p_d = \frac{\delta_{(Q)}}{\lambda_B + \delta(Q)}$;
2. Labels of components are randomly permutated; and
3. Initial values of parameters for the next round of Gibbs sampler are calculated from the permuted output of the previous run.

As was previously reported by Wakefield *et al.*[16] and Zhou,[11] the number of clusters is highly sensitive to the choice of prior distributions. Hence, biological and experiment-specific insight should be used to determine informative priors. Based on the biological analysis by Chu *et al.*,[15] we set parameter $\lambda_B = 7$ (the expected number of clusters) and ran the simulation for 1,000 hybrid birth-death steps, each containing 5,000 WinBUGS iterations; the first 4,000 WinBUGS iterations were discarded as "burn-in". Stephens' relabeling method[8] was used to determine parameters of individual clusters and cluster membership.

The distribution of the number of mixture components (clusters) K is shown in Fig. 4. From this distribution, it is easy to see that the Markov chain spent most (34%) of the time in the state $K = 7$. Hence, we use the parameters estimated for $K = 7$ to describe KLMCMC profiles for individual clusters (Tables 2 and 3). It is remarkable that the KLMCMC approach was able to reproduce the results of Chu *et al.*,[15] who partitioned genes involved in the yeast sporulation process into seven groups. KLMCMC efficiently dealt with observation errors and gene-specific variances.

KLMCMC profiles are shown in Fig. 5. Figure 6 shows observations grouped into seven clusters along with KLMCMC profiles and cluster mean curves, defined as the average expression value at each time for all cluster members. We can see that the KLMCMC profile produces smoother and more biologically meaningful curves, as compared to cluster mean curves.

Clusters obtained by the KLMCMC process can be further extended using the entire collection of microarray data. For example, "late" sporulation genes are described by the KLMCMC profile as $f_{late}(t) = (-0.305 + 1.668t)e^{0.1566(t-7.555)} \times \eta(t - 7.555)$. All unclustered genes are ordered by Pearson's correlation coefficient of expression profiles and compared to the cluster in a sequential manner. After an addition of each gene, parameters of the cluster are recomputed. If a parameter distribution becomes too vague upon an inclusion of an extra gene, this gene is not added to the cluster. When we added 15 cluster members, the distribution of cluster parameters became $f_{15}(t_j) = (-0.322 + 0.141t_j e^{0.09(t_j - 7.72)}\eta(t_j - 7.72)$. In this manner, we can add as many as 71 genes without distorting the initial parameter distributions $f_{71}(t_j) = (-0.21 + 0.12t_j)e^{0.04(t_j - 6.1)}\eta(t_j - 6.1)$ (Fig. 7). If we compare this approach with the traditional nearest neighbor clustering based on Pearson's correlation coefficient of expression profiles, the cluster mean for the top 71 genes with correlation coeffcient >0.9 will be shifted towards negative expression values (Fig. 8).

7. Conclusions

KLMCMC is a novel method for time-series observation clustering that can be applied after some initial data filtering and preclustering analyses. It is a transdimensional method: it determines the optimal number of clusters for a given dataset and selected model. In addition to cluster membership, KLMCMC produces a meaningful and easy-to-interpret biological profile. KLMCMC requires careful selection of the model for fitting; hence, it is best suited for refinement of some preliminary clustering. The number of clusters is sensitive to the choice of prior distributions; hence, biological and experiment-specific insight should be used to determine informative priors. Our method can be successfully used for cluster refinement and construction of models of cellular processes. Since KLMCMC does not involve likelihood evaluation, it is more computationally efficient as compared to traditional methods (BDMCMC and reversible jump MCMC) which, in the case of nonlinear mixture models, require multiple numeric evaluations of integrals.

Acknowledgments

This work was supported in part by National Institute of Health Grants P41-EB001978 and R01-GM068968.

Biographies

Tatiana Tatarinova is a Visiting Professor at the Department of Mathematics, Loyola Marymount University, USA. She received her Diploma in Theoretical Physics from the Moscow Engineering Physics Institute, Russia; her M.Sc. degree in Physics from the University of Utah, USA; and her Ph.D. in Applied Mathematics from the University of Southern California, USA. She is working in the area of computational biology; and her main research interests are pharmacokinetics, microarray data analysis, prediction of patterns in genomes, and software development.

John Bouck is Director of Information Technology at Ceres, Inc., a bioenergy company based in Thousand Oaks, California, USA. He graduated from the University of Wisconsin-Madison in Computer Science, and received his Ph.D. in Molecular Biology and Genetics from the University of Pennsylvania. He led the Bioinformatics effort at the Baylor College of Medicine Human Genome Sequencing Center through the publication of the human draft sequence. He also led the Bioinformatics, Cheminformatics, and Library Sciences group for the mid-sized biotechnology company UCB Pharma. His research interests include comparative genomics, functional annotation of genomes, and pathway analysis.

Alan Schumitzky is a Professor of Mathematics at the University of Southern California, Los Angeles, CA, USA. He received his Ph.D. degree in Mathematics from Cornell University, Ithaca, NY. He has taught at the University of California, Berkeley, CA; and was a Visiting Scholar in the Department of Biomathematics, University of California, Los Angeles, CA. He has been a Visiting Professor at Victoria University, Wellington, New Zealand, and at the University of Washington, Seattle, WA, USA. His research interests include control theory

and applications, estimation theory and applications, complex analysis, applied pharmacokinetics, population pharmacokinetics theory and applications, and software development. He is a member of the Laboratory of Applied Pharmacokinetics at the University of Southern California, and is a member of the American Mathematical Society.

References

1. Müller, P. Technical Report. Purdue University; West Lafayette, IN: 1990. A generic approach to posterior integration and Gibbs sampling.
2. Tatarinova, T. Doctoral dissertation. University of Southern California; Los Angeles, CA: 2006. Bayesian analysis of linear and nonlinear mixture models.
3. Celeux G, Hurn M, Robert CP. Computational and inferential difficulties with mixture posterior distributions. J Am Stat Assoc 2000;95:957–970.
4. Fruhwirth-Schnatter S. Markov chain Monte Carlo estimation of classical and dynamic switching and mixture models. J Am Stat Assoc 2001;96(453):194–209.
5. Stephens M. Bayesian analysis of mixture models with an unknown number of components — An alternative to reversible jump methods. Ann Stat 2000;28(1):40–74.
6. Stephens M. Contribution to the discussion of paper by Richardson and Green. JR Stat Soc Ser B 1997;59:768–769.
7. Stephens, M. Ph.D. thesis. University of Oxford; Oxford, UK: 1997. Methods for mixtures of normal distributions.
8. Stephens M. Dealing with label switching in mixture models. J R Stat Soc Ser B 2000;62:795–809.
9. Sahu SK, Cheng RCH. A fast distance based approach for determining the number of components in mixtures. Can J Stat 2003;31(1):3–22.
10. Casella G, George EI. Explaining the Gibbs sampler. Am Stat 1992;46:167–174.
11. Zhou, C. Ph.D. thesis. University of Washington; Seattle, WA: 2003. A Bayesian model for curve clustering with application to gene expression analysis.
12. Lunn DJ, Best N, Thomas A, Wakefield J, Spiegelhalter D. Bayesian analysis of population PK/PD models: General concepts and software. J Pharmacokinet Pharmacodyn 2002;29:217–307.
13. Dowe, DL.; Baxter, RA.; Oliver, JJ.; Wallace, CS. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD98). Vol. Vol. 1394. Lecture Notes in Artificial Intelligence, Springer-Verlag; Berlin: 1998. Point estimation using the Kullback- Leibler loss function and MML; p. 87-95.
14. Fitzgibbon, LJ.; Dowe, DL.; Allison, L. Technical Report 107. School of Computer Science and Software Engineering, Monash University; Clayton, Victoria, Australia: 2002. Message from Monte Carlo.
15. Chu S, DeRisi J, Eisen M, Mulholland J, Botstein D, Brown PO, Herskowitz I. The transcriptional program of sporulation in budding yeast. Science 1998;282(5389):699–705. [PubMed: 9784122]
16. Wakefield JC, Zhou C, Self C. Modeling gene expression over time: Curve clustering with informative prior distribution. Bayesian Stat 2003;7:721–732.
17. Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci USA 1998;95:14863–14868. [PubMed: 9843981]
18. MacQueen JB. Some methods for classification and analysis of multivariate observations. Proc Fifth Berkeley Symp Math Stat Probab 1967;1:281–297.
19. Hartigan JA, Wong MA. A *K*-means clustering algorithm. Appl Stat 1979;28:126–130.
20. Kohonen T. The self-organizing map. Proc IEEE 1990;78:1464–1480.
21. Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. Proc Natl Acad Sci USA 1999;96:2907–2912. [PubMed: 10077610]
22. Raychaudhuri S, Stuart JM, Altman RB. Principal components analysis to summarize microarray experiments: Application to sporulation time series. Pac Symp Biocomput 2000:455–466. [PubMed: 10902193]

23. Yeung, KY.; Ruzzo, WL. Technical Report UW-CSE-00-11-03. Dept. of Computer Science and Engineering, University of Washington; Seattle, WA: 2000. An empirical study on principal component analysis for clustering gene expression data.
24. Alter O, Brown PO, Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. Proc Natl Acad Sci USA 2000;97:10101–10106. [PubMed: 10963673]
25. Kaufman, L.; Rousseeuw, P. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley; New York: 1990.
26. Yeung KY, Fraley C, Murua A, Raftery AE, Ruzzo WL. Model-based clustering and data transformations for gene expression data. Bioinformatics 2001;17:977–987. [PubMed: 11673243]
27. Fraley, C.; Raftery, AE. Technical Report. Department of Statistics, University of Washington; Seattle, WA: 2007. MCLUST: Software for model-based clustering, density estimation and discriminant analysis.
28. Fraley C, Raftery AE. Model-based clustering, discriminant analysis, and density estimation. J Am Stat Assoc 2002;97:611–631.
29. McLachlan GJ, Bean RW, Peel D. A mixture model-based approach to the clustering of microarray expression data. Bioinformatics 2002;18:413–422. [PubMed: 11934740]
30. Medvedovic M, Sivaganesan S. Bayesian infinite mixture model based clustering of gene expression profiles. Bioinformatics 2002;18:1194–1206. [PubMed: 12217911]
31. Medvedovic M, Yeung KY, Bumgarner RE. Bayesian mixture model based clustering of replicated microarray data. Bioinformatics 2004;20:1222–1232. [PubMed: 14871871]
32. Tseng GC, Wong WH. Tight clustering: A resampling-based approach for identifying stable and tight patterns in data. Biometrics 2005;61:10–16. [PubMed: 15737073]
33. Thalamuthu A, Mukhopadhyay I, Zheng X, Tseng GC. Evaluation and comparison of gene clustering methods in microarray analysis. Bioinformatics 2006;22:2405–2412. [PubMed: 16882653]

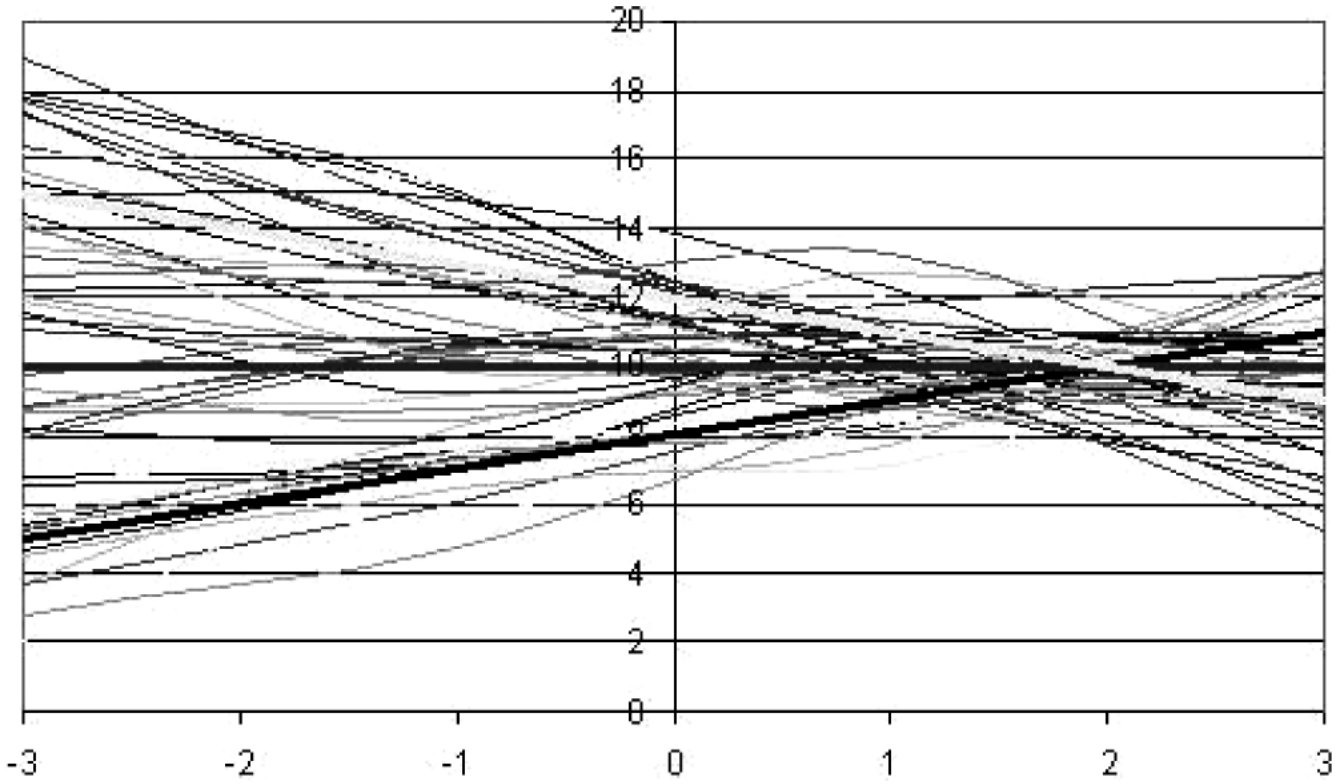

Fig. 1.
Simulated time series. Plot of 50 simulated curves and three cluster means (thick lines).

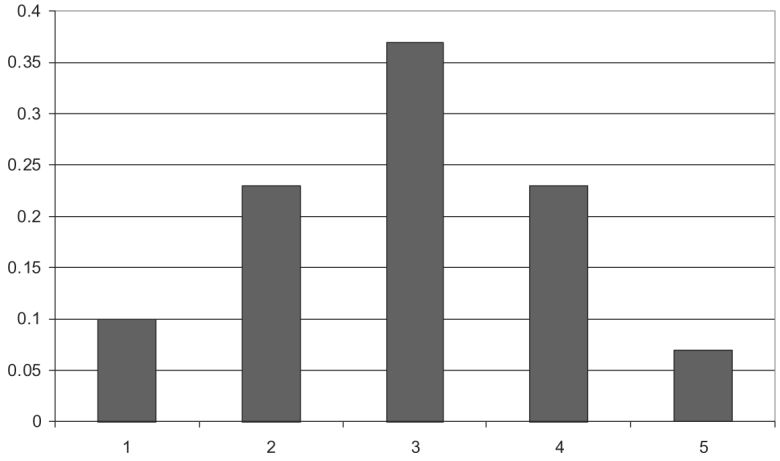

Fig. 2.

Simulated time series. Posterior distribution of the number of components for the simulated time series problem, KLMCMC with $\lambda = \lambda_B = 3$.

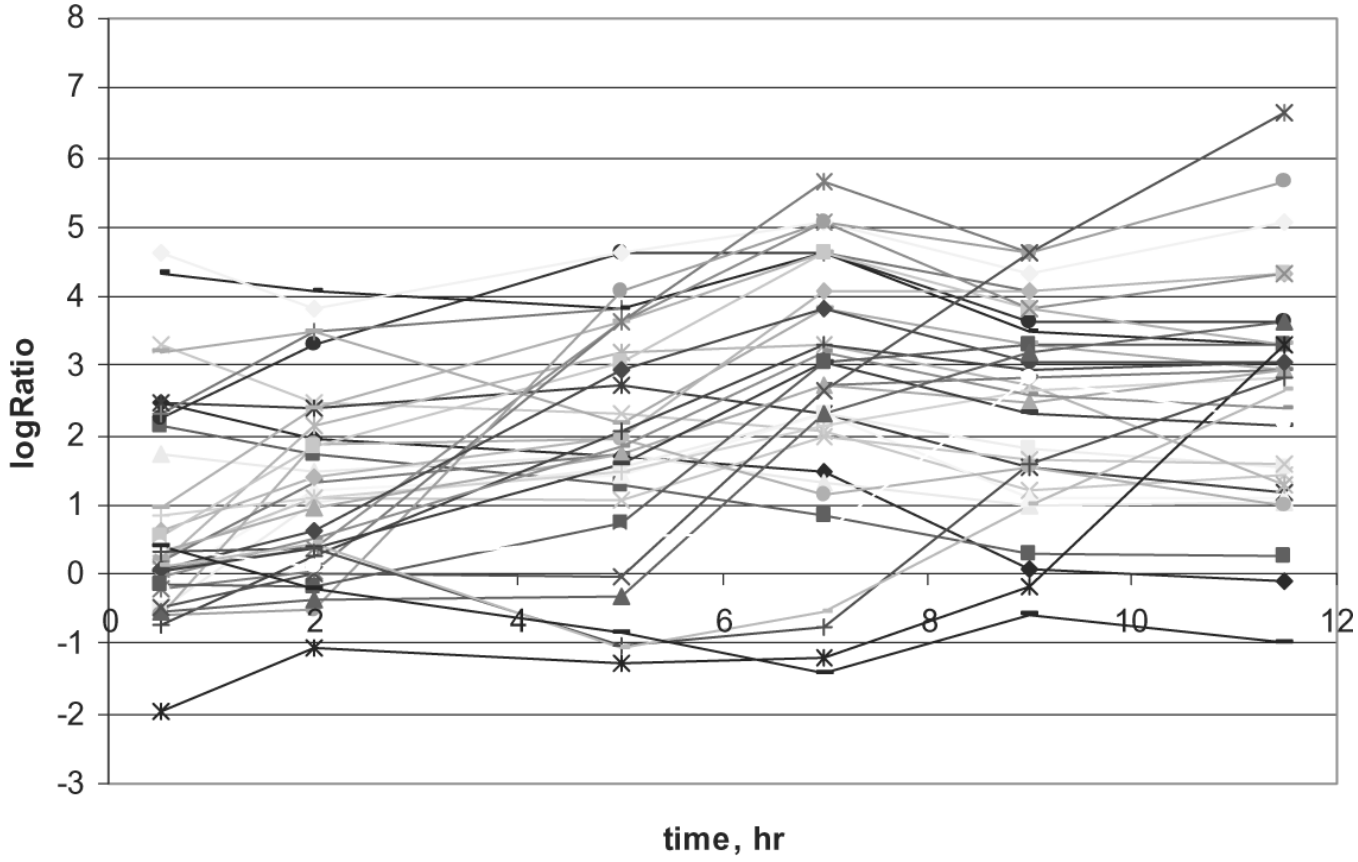

Fig. 3.
Gene expression time series of sporulation in budding yeast, corresponding to seven characteristic profiles, as identifield by Chu *et al.*[15]

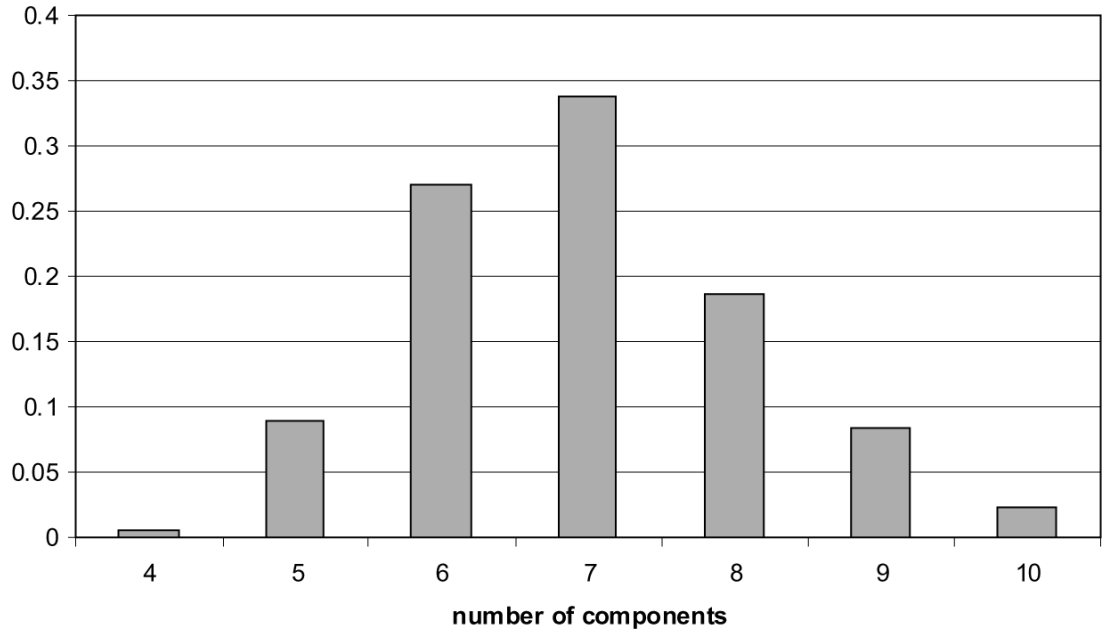

Fig. 4.
Distribution of the number of components *K* for sporulation in budding yeast time series.

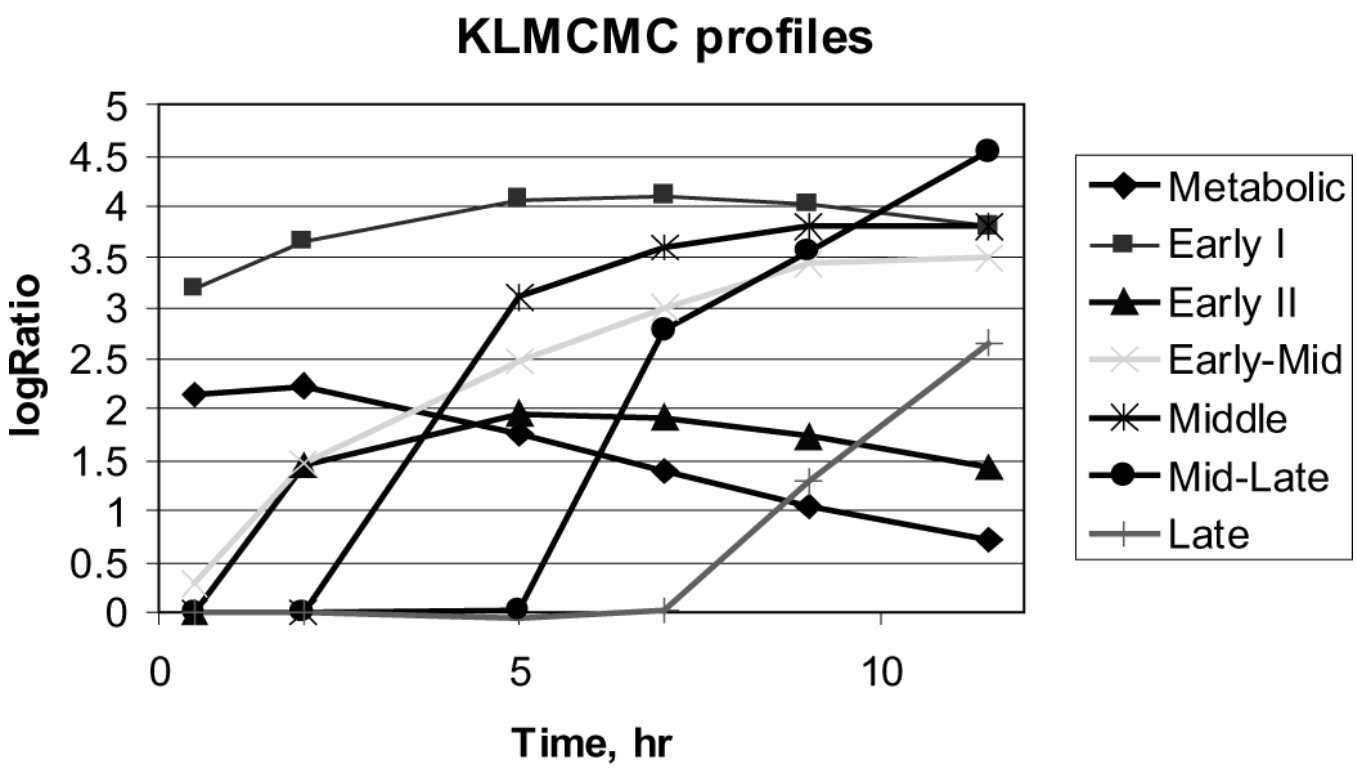

Fig. 5.
Cluster profiles of temporal patterns of sporulation in budding yeast, as defined by KLMCMC.

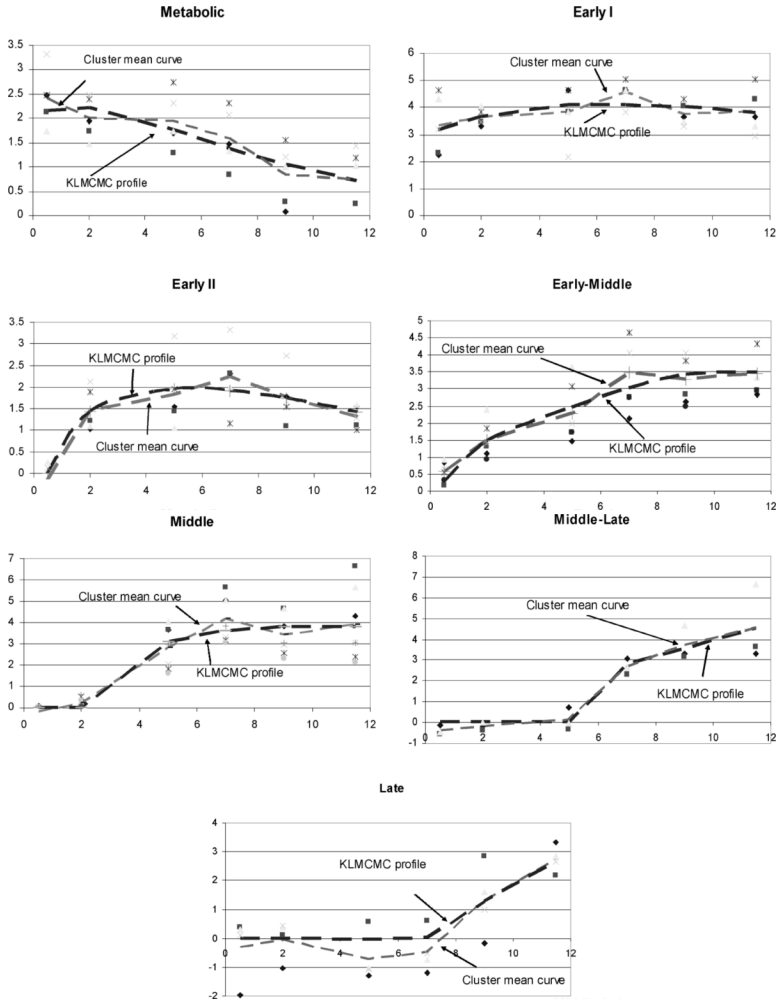

Fig. 6.
Seven clusters of temporal patterns of sporulation in budding yeast, as defined by KLMCMC.

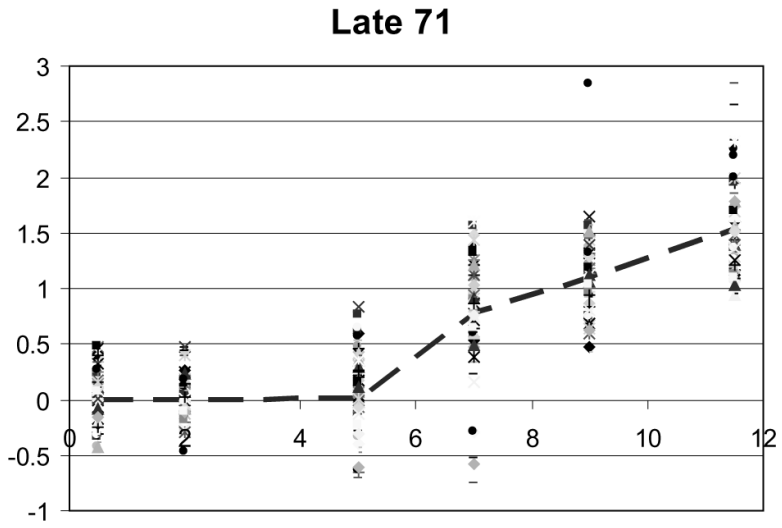

Fig. 7.
Extended “late” cluster with 71 members.

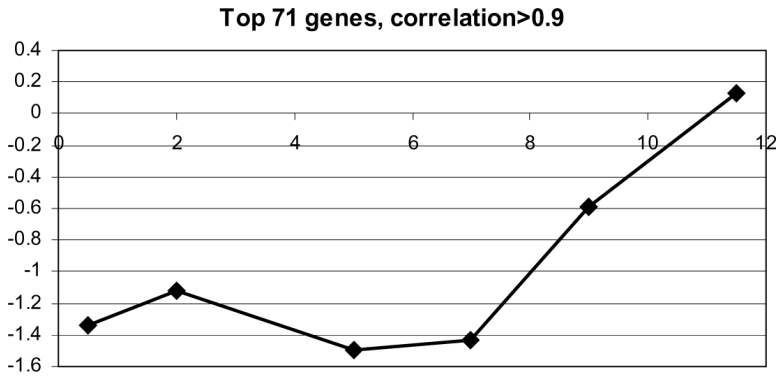

Fig. 8.
Nearest neighbor clustering results for the top 71 genes, with expression correlated to the members of the “late” sporulation cluster.

Table 1

Simulated time series. Posterior estimates for the elements of the covariance matrix for traditional Gibbs sampler, KLMCMC, and KLMCMC with RPS, conditional on $K = 3$.

Method	Σ_{11}^1	Σ_{11}^2	Σ_{11}^3	Σ_{11}^1	Σ_{11}^2	Σ_{11}^3
Gibbs sampler	0.18	0.18	0.28	0.15	0.23	0.13
KLMCMC	0.1	0.1	0.1	0.1	0.1	0.1
KLMCMC + RPS	0.1	0.11	0.19	0.1	0.11	0.19
Original values	0.1	0.1	0.2	0.1	0.1	0.2

Table 2

Yeast sporulation times series parameter estimates for KLMCMC with RPS, conditional on $K = 7$.

Cluster	**1**	**2**	**3**	**4**	**5**	**6**	**7**
μ_1	1.9540	3.034	0.1588	0.2886	-0.01791	-0.0074	-0.305
μ_2	0.6325	0.5189	0.7532	0.6336	0.7208	0.3987	0.1668
μ_3	0.2105	0.06239	0.1678	0.04974	0.09469	-0.00197	-0.1566
μ_4	0.2585	0.2526	1.174	0.7561	3.483	5.958	7.555
Σ_{11}	3.391	8.974	0.9738	0.8739	1.035	1.292	1.357
Σ_{22}	0.7824	0.7367	0.811	0.7402	0.8183	0.6485	0.6438
Σ_{33}	0.2621	0.2597	0.2619	0.2593	0.2592	0.2583	0.2608

Table 3

Yeast sporulation time series KLMCMC profiles conditional on $K = 7$.

Cluster	Equation
Metabolic	$f_{\text{metabolic}}(t) = (1.1954 + 0.6325\,t)e^{-0.2105(t-0.2585)}\eta(t - 0.2585)$
Early I	$f_{\text{earlyI}}(t) = (3.034 + 0.5189\,t)e^{-0.06239(t-0.2526)}\eta(t - 0.2626)$
Early II	$f_{\text{earlyII}}(t) = (0.1588 + 0.7532\,t)e^{-0.1678(t-1.174)}\eta(t - 1.174)$
Early-Middle	$f_{\text{early-mid}}(t) = (0.2886 + 0.6336\,t)e^{-0.04974(t-0.7561)}\eta(t - 0.7561)$
Middle	$f_{\text{middle}}(t) = (-0.01791 + 0.7208\,t)e^{-0.09469(t-3.489)}\eta(t - 3.489)$
Mid-Late	$f_{\text{mid-late}}(t) = (-0.0074 + 0.3987\,t)e^{0.00197(t-5.958)}\eta(t - 5.958)$
Late	$f_{\text{late}}(t) = (-0.305 + 1.668\,t)e^{0.1566(t-7.555)}\eta(t - 7.555)$