BIOINFORMATICS

Vol. 19 Suppl. 2 2003, pages ii196–ii205
DOI: 10.1093/bioinformatics/btg1078

Biclustering microarray data by Gibbs sampling

Qizheng Sheng, Yves Moreau and Bart De Moor*

Department of Electrical Engineering ESAT-SCD, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, Leuven-Heverlee, 3001, Belgium

Received on March 17, 2003; accepted on June 9, 2003

ABSTRACT

Motivation: Gibbs sampling has become a method of choice for the discovery of noisy patterns, known as motifs, in DNA and protein sequences. Because handling noise in microarray data presents similar challenges, we have adapted this strategy to the biclustering of discretized microarray data.

Results: In contrast with standard clustering that reveals genes that behave similarly over all the conditions, biclustering groups genes over only a subset of conditions for which those genes have a sharp probability distribution. We have opted for a simple probabilistic model of the biclusters because it has the key advantage of providing a transparent probabilistic interpretation of the biclusters in the form of an easily interpretable fingerprint. Furthermore, Gibbs sampling does not suffer from the problem of local minima that often characterizes Expectation–Maximization. We demonstrate the effectiveness of our approach on two synthetic data sets as well as a data set from leukemia patients.

Contact: qizheng.sheng@esat.kuleuven.ac.be

INTRODUCTION

The ability of microarrays to monitor transcriptional behavior over a whole genome under different conditions is a major attraction for biologists. Clustering techniques, which discover groups of genes that share similar transcriptional behavior over the conditions in a microarray experiment, play a big role in microarray data analysis. Standard clustering methods, such as hierarchical clustering, K-means, or self-organizing maps, all assume that genes in a cluster behave similarly over all the conditions presented in a microarray experiment. Under this assumption, standard clustering methods produce reliable results for microarray experiments performed on homogeneous conditions. However, when the conditions of a microarray experiment form a heterogeneous compendium, this assumption is no longer appropriate. In this case, biclustering algorithms are preferable, because they can detect those relevant conditions for which the relation between the genes of a potential group exists. Some of the existing biclustering algorithms are based on the idea to perform standard clustering algorithms iteratively on both genes and conditions (Getz *et al.*, 2000; Busygin *et al.*, 2002). Other recent biclustering approaches (Lazzeroni and Owen, 2000; Cheng and Church, 2000; Segal *et al.*, 2001; Ben-Dor *et al.*, 2001; Tanay *et al.*, 2002) rely on a variety of optimization procedures.

To tackle the problem in the Bayesian framework, we present a biclustering strategy based on a simple frequency model for the expression pattern of a bicluster and on Gibbs sampling for parameter estimation. Gibbs sampling has become a method of choice in the discovery of statistically overrepresented subsequences, known as motifs, in DNA and protein sequences data thanks to its high sensitivity (Lawrence *et al.*, 1993; Liu *et al.*, 2000; Thijs *et al.*, 2002). Because finding similar gene behavior across a subset of conditions once the microarray data is discretized resembles the problem of finding subsequences sharing similar alphabetic expressions in sequence data, we have adapted the Gibbs sampling strategy to the biclustering of discretized microarray data. Our approach not only unveils genes and conditions of a bicluster, but also represents the pattern of a bicluster as a probabilistic model described by the posterior frequency of every discretized expression level discovered under each condition of the bicluster. This description provides a transparent interpretation of the bicluster. In addition, the probabilistic model can be exploited as an easily interpretable fingerprint of the genes of the bicluster and be applied in predicting potentially related genes by scanning the candidate genes with the fingerprint. Moreover, the choice of Gibbs sampling avoids the problems of local minima that often characterizes the closely related strategy of Expectation-Maximization.

Although so far we have only introduced our approach as a way to classify genes that share similar behaviors over a subset of conditions, it is easy to understand that the method can also be used to discover biclusters in the other orientation of the microarray data. We will actually demonstrate our method on grouping patients (i.e. conditions) who exhibit similar expression behavior over a subset of genes. As a matter of fact, our Gibbs sampling

*To whom correspondence should be addressed.

strategy applies to the biclustering of discrete data in general and not only from microarray experiments. However, for the clarity of the presentation, when developing our method, we will continue to assume the gene-condition orientation of a microarray data set introduced at the beginning of the article.

PROBABILISTIC MODEL OF A BICLUSTER

Consider a microarray data set that contains n genes and m conditions and assume for the moment that a single bicluster is present in the data. We introduce two vectors

$$g = \begin{bmatrix} g_1 & g_2 & \dots & g_n \end{bmatrix}^T$$
$$\text{and } c = \begin{bmatrix} c_1 & c_2 & \dots & c_m \end{bmatrix}^T,$$

whose elements g_i (for $i = 1, 2, \dots, n$) and c_j (for $j = 1, 2, \dots, m$) are Bernoulli random variables indicating respectively whether the ith gene and the jth condition belong to the bicluster. Hereafter we refer to these vectors as the label vectors and to the Bernoulli random variables that they contain as the labels. (For example, these labels are visually depicted by the outer bars in Fig. 2a).

As mentioned before, we only consider discretized microarray data. In this case, we use multinomial distributions to model the data. Let us assume that the data under study is preprocessed in such a way that the background data (the part of the data that does not belong to the bicluster) is generated by one single multinomial distribution characterized by the following parameters:

$$\begin{gathered} \phi = \begin{bmatrix} \varphi_1 & \varphi_2 & \cdots & \varphi_l \end{bmatrix}^T, \\ 0 \le \varphi_i \le 1, \ \sum \varphi_i = 1, \ \text{for } i = 1, \dots, l \end{gathered} \tag{1}$$

where l is the total number of bins used for discretization. The bicluster that we seek is a subset of the data where the genes behave similarly under each condition. It is important to note this asymmetric nature of the underlying probabilistic model. That is, we ask that the expression level be consistent across the genes of the bicluster for each of the selected conditions, but this expression level maybe different for each condition, (an example of such a data pattern can be seen in Fig. 2d). To put this mathematically, we use a multinomial distribution to model the data under every condition in a bicluster, and we also assume that the multinomial distributions for different conditions of a bicluster are mutually independent. We refer to this model of the bicluster as the pattern Θ, a matrix where each column $\Theta_{\cdot j}$ contains the parameters for the jth independent multinomial distribution

$$\Theta = \begin{bmatrix} \theta_{1,1} & \cdots & \theta_{1,j} & \cdots & \theta_{1,w} \\ \theta_{2,1} & \cdots & \theta_{2,j} & \cdots & \theta_{2,w} \\ \vdots & & \vdots & & \vdots \\ \theta_{l,1} & \cdots & \theta_{l,j} & \cdots & \theta_{l,w} \end{bmatrix} \tag{2}$$
$$0 \le \theta_{i,j} \le 1, \ \sum_i \theta_{i,j} = 1$$
$$\text{for } i = 1, \dots, l; \ j = 1, \dots, w$$

where w denotes the total number of conditions in the bicluster.

Working with Gibbs sampling will set our model in the Bayesian framework, which means that our probabilistic model is accompanied by prior distributions for its different parameters. According to the definition, we use Bernoulli distributions with parameters λ_g and λ_c as the prior distribution respectively for the row labels g and the column labels c. Further, we use conjugate priors for ϕ, Θ, and the λs. That is, ϕ, Θ, and the λs respectively follow a Dirichlet distribution, a multi-Dirichlet distribution and Beta distributions.

$$\begin{aligned} \phi &\sim \text{Dirichlet}(\alpha), \\ \Theta_{\cdot j} &\sim \text{Dirichlet}(\beta_j), \ \text{for } j = 1, 2, \dots, w \\ \lambda_g &\sim \text{Beta}(\xi_g) \text{ and } \lambda_c \sim \text{Beta}(\xi_c) \end{aligned} \tag{3}$$

where α and β_j are parameter vectors of the Dirichlet distributions, and ξ_g and ξ_c are the parameter vectors for the corresponding Beta distributions. In a practical sense, α, β_j, and the ξs can also be viewed as vectors of pseudocounts, which represent our prior knowledge of the background and the possible pattern, and our prior knowledge on the possibility that a label equals 1.

Multiple biclusters

Our probabilistic model considers only the presence of a single bicluster in the data set, which is not biologically realistic. Several methods can be used to enable the detection of multiple biclusters. We choose (in the case of the gene-condition orientation) to mask the genes selected for the found biclusters and rerun the algorithm on the rest of the data. By masking, we mean that the gene labels of all the found biclusters are permanently set to zero. In this way, genes retrieved for previous biclusters will not further be selected as candidate genes for any future bicluster, while the background model will still be calculated over all the possible positions in the whole data set including the positions of the masked genes or conditions. Note that this choice does allow the unmasked dimension of the biclusters to be selected multiple times. So, in the case of the gene-condition orientation, a condition can be relevant to multiple biclusters to be selected multiple

times. In this way, the algorithm is iterated on a data set until no bicluster can be found for the unmasked part of the data, (see Section 'data without a bicluster' for the decision). If the biclustering takes place in the condition-gene orientation, the same procedure can be applied but then a condition can only belong to a single bicluster, while a gene can be relevant for several biclusters.

Another approach to find multiple biclusters would be to allow the gene and condition labels to take discrete values indicating to which of the several biclusters a gene or condition belongs. We decided against this option because, take the gene-condition orientation for example, a condition can never be relevant to multiple biclusters (which would be an unacceptable biological limitation). This problem could be alleviated by using several binary vectors of labels (as many as the number of biclusters we are looking for), which would then allow biclusters overlapping in both the gene and the condition dimensions. However, the increase in the number of parameters to estimate, together with the need for a procedure for the estimation of the number of biclusters, led us to settle for the simpler masking procedure in a first instance (although we will explore the alternative approach later).

DISCOVERING BICLUSTERS BY GIBBS SAMPLING

Gibbs sampling is one of the best known Markov chain Monte Carlo methods. Suppose we want to draw samples for the random variables x, y, and z, but that the marginal distributions or the joint distribution are too complex to sample directly from them. Suppose also that the conditional distributions $p(x|y,z)$, $p(y|x,z)$, and $p(z|x,y)$, which can easily be sampled from, are available. Starting from initial values $y^{(0)}$ and $z^{(0)}$, the Gibbs sampler draws samples for the three variables in the following manner:

$$\begin{aligned} x^{(t+1)} &\sim p(x|y^{(t)}, z^{(t)}) \\ y^{(t+1)} &\sim p(y|x^{(t+1)}, z^{(t)}) \\ z^{(t+1)} &\sim p(z|x^{(t+1)}, y^{(t+1)}) \end{aligned}$$

for $t = 0, 1, 2, \ldots$. It can be shown that the sequence

$$y^{(0)},\ z^{(0)},\ x^{(1)},\ y^{(1)},\ z^{(1)},\ \ldots,\ x^{(k)},\ y^{(k)},\ z^{(k)}$$

constructs a Markov chain and that, as $k \to \infty$, the distribution of the triplet $(x^{(k)}, y^{(k)}, z^{(k)})$ converges to the true joint distribution $p(x,y,z)$. Furthermore, the sequence $x^{(1)}, x^{(2)}, \ldots, x^{(k)}$ itself is a Markov chain, and the distribution of $x^{(k)}$ converges to its true marginal distribution $p(x)$ as $k \to \infty$. The same can be said for the variables y and z. For an introduction to Gibbs sampling, we refer to (Casella and George, 1992).

Our goal is to draw samples from the joint distribution $p(g, c|D)$ of g and c conditioned on a discretized microarray data set D. In other words, we want to generate samples for every component in g and c from its respective marginal distribution $p(g_i|D)$ or $p(c_j|D)$. In the manner of Gibbs sampling, this can be done by sampling iteratively from the full conditional distributions

$$\begin{aligned} &p(g_i|g_{\bar{i}}, c, D), \text{ for } i = 1, 2, \ldots, n, \\ \text{and } &p(c_j|g, c_{\bar{j}}, D), \text{ for } j = 1, 2, \ldots, m, \end{aligned}$$

where $g_{\bar{i}}$ (or $c_{\bar{j}}$) denotes a label vector with all but the ith gene (or jth condition) label fixed.

Full conditional distributions

The full conditional distributions can be derived based on the fact that

$$\begin{aligned} p(g_i|g_{\bar{i}}, c, D) &\propto p(g_i, g_{\bar{i}}, c, D) = p(g, c, D) \\ \text{and} \quad p(c_j|g, c_{\bar{j}}, D) &\propto p(c_j, g, c_{\bar{j}}, D) = p(g, c, D). \end{aligned}$$

Observe also that $P(g, c, D)$ can be obtained by integrating Θ, ϕ, and λs out of the likelihood function $\mathcal{L}(\Theta, \phi, \lambda_g, \lambda_c|g, c, D)$.

Given the background model ϕ, the pattern model Θ, and the λs, the likelihood of the complete data (which includes the observed data D and the labels g and c) is

$$\begin{aligned} &\mathcal{L}(\Theta, \phi, \lambda_g, \lambda_c|D, g, c) \\ =&P(D, g, c|\Theta, \phi, \lambda_g, \lambda_c) \\ =&P(D|g, c, \Theta, \phi) \cdot p(g|\lambda_g) \cdot p(c|\lambda_c) \\ =&\phi^{\boldsymbol{c}(\boldsymbol{b}(g,c))} \prod_{j=1}^{w} \Theta_{\cdot j}^{\boldsymbol{c}(\boldsymbol{P}_{\cdot j}(g,c))} \\ &t{\lambda_g}^{v}(1-\lambda_g)^{n-v} \cdot {\lambda_c}^{w}(1-\lambda_c)^{m-w}, \end{aligned}$$

where v denotes the number of genes that belong to the bicluster. In addition, we use $\boldsymbol{b}(x,y)$ (or $\boldsymbol{P}(x,y)$) to denote the part of the data where the background (or pattern) model is evaluated, with x and y (which are respectively gene and condition label vectors) providing information for selecting the data points for the evaluation. We define $\boldsymbol{c}(\cdot)$ as a counting function that returns a vector of length l indicating the number of times each discretized value is observed at the data points specified in the bracket. We also define for the vectors $r = [r_1, \cdots, r_k]^T$ and $s = [s_1, \cdots, s_k]^T$ that $r^s = r_1^{s_1} \cdots r_k^{s_k}$, and $\Gamma(s) = \Gamma(s_1) \cdots \Gamma(s_k)$ where $\Gamma(\cdot)$ is the gamma function generalizing the factorial.

Integrating the model parameters out, we have,

$$
\begin{aligned}
& P(D, g, c) \\
= & \iint [P(D, g, c|\Theta, \phi, \lambda_g, \lambda_c) \cdot p(\Theta) \cdot p(\phi) \\
& \cdot p(\lambda_g) \cdot p(\lambda_c)] \, \mathrm{d}\Theta \, \mathrm{d}\phi \, \mathrm{d}\lambda_g \, \mathrm{d}\lambda_c \\
\propto & \int_{\phi} \phi^{\boldsymbol{c}[\boldsymbol{b}(g,c)]} \phi^{\alpha-1} \, \mathrm{d}\phi \cdot \int_{\Theta} \prod_{j=1}^{w} \Theta_{\cdot j}^{\boldsymbol{c}[\boldsymbol{P}_{\cdot j}(g,c)]} \Theta_{\cdot j}^{\beta_j - 1} \, \mathrm{d}\Theta \\
& \cdot \int_{\lambda_g} {\lambda_g}^{v} (1-\lambda_g)^{n-v} {\lambda_g}^{\xi_{g1}-1} (1-\lambda_g)^{\xi_{g2}-1} \, \mathrm{d}\lambda_g \\
& \cdot \int_{\lambda_c} {\lambda_c}^{w} (1-\lambda_c)^{m-w} {\lambda_c}^{\xi_{c1}-1} (1-\lambda_c)^{\xi_{c2}-1} \, \mathrm{d}\lambda_c \\
\propto & \frac{\Gamma\{\boldsymbol{c}[\boldsymbol{b}(g,c)] + \alpha\}}{\Gamma\big(\sum \{\boldsymbol{c}[\boldsymbol{b}(g,c)] + \alpha\}\big)} \\
& \cdot \prod_{j=1}^{w} \frac{\Gamma\{\boldsymbol{c}[\boldsymbol{P}_{\cdot j}(g,c)] + \beta_j\}}{\Gamma\big(\sum \{\boldsymbol{c}[\boldsymbol{P}_{\cdot j}(g,c)] + \beta_j\}\big)} \\
& \cdot \frac{\Gamma(v + \xi_{g1}) \, \Gamma(n - v + \xi_{g2})}{\Gamma(n + \xi_{g1} + \xi_{g2})} \\
& \cdot \frac{\Gamma(w + \xi_{c1}) \, \Gamma(m - w + \xi_{c2})}{\Gamma(m + \xi_{c1} + \xi_{c2})},
\end{aligned}
$$

where ξ_{g1} and ξ_{g2} denote respectively the first and the second element in ξ_g, (same for ξ_c); and for a vector s, the notation $\sum(s)$ denotes the sum of all the elements in the vector.

By writing the concerned label separately from the rest of the labels in the above equation and working further on the reduction, we will finally arrive at the following parameterization of the full conditional distributions.

The two terms that characterize the Bernoulli posterior distribution of gene label g_i are

$$
\begin{aligned}
P(g_i = 1|D, g_{\bar{i}}, c) = & \frac{1}{Z_g} \prod_{j=1}^{w} \hat{\eta}[\boldsymbol{P}_{\cdot j}(g_{\bar{i}}, c)]^{\boldsymbol{c}[\boldsymbol{P}_j(\delta_i, c)]} \\
& \cdot (v_{\bar{i}} + \xi_{g1}) \\
P(g_i = 0|D, g_{\bar{i}}, c) = & \frac{1}{Z_g} \hat{\eta}[\boldsymbol{b}(g_{\bar{i}}, c)]^{\boldsymbol{c}[\boldsymbol{P}(\delta_i, c)]} \\
& \cdot (n - 1 - v_{\bar{i}} + \xi_{g2})
\end{aligned} \tag{4}
$$

where δ_i denotes a gene label vector whose ith entry is 1 and the other entries are 0, $v_{\bar{i}}$ is the number of gene labels that are 1 in $g_{\bar{i}}$, and Z_g is a normalization term such that $P(g_i = 1|g_{\bar{i}}, c, D) + P(g_i = 0|g_{\bar{i}}, c, D) = 1$. The $\hat{\eta}(\cdot)$ in the above equation stands for a function for calculating the posterior mean of the specified data points. We obtain thus the byproduct of our method – the posterior pattern model evaluated at the currently assigned biclustering positions, and the posterior background model evaluated at the currently assigned background positions.

$$
\begin{aligned}
\hat{\Theta}_{\cdot j} = \hat{\eta}[\boldsymbol{P}_{\cdot j}(g_{\bar{i}}, c)] &= \frac{\boldsymbol{c}[\boldsymbol{P}_{\cdot j}(g_{\bar{i}}, c)] + \beta_j}{\sum\{\boldsymbol{c}[\boldsymbol{P}_{\cdot j}(g_{\bar{i}}, c)] + \beta_j\}} \\
\hat{\phi} = \hat{\eta}[\boldsymbol{b}(g_{\bar{i}}, c)] &= \frac{\boldsymbol{c}[\boldsymbol{b}(g_{\bar{i}}, c)] + \alpha}{\sum\{\boldsymbol{c}[\boldsymbol{b}(g_{\bar{i}}, c)] + \alpha\}}.
\end{aligned} \tag{5}
$$

Intuitively, by fixing all the other labels to the values sampled in previous Gibbs sampling steps, the possibility that gene i contributes to the pattern is associated with the likelihood that the data of the gene under the currently assigned bicluster conditions is generated by the pattern; while the possibility that the gene belongs to the background is related to the likelihood that those data points are drawn from the background model.

When it comes to the label of the jth condition, we finally have a Bernoulli posterior distribution described by

$$
\begin{aligned}
P(c_j = 1|g, c_{\bar{j}}, D) = & \frac{1}{Z_c} \cdot \frac{\Gamma\{\boldsymbol{c}[\boldsymbol{b}(g, c_{\bar{j}})] + \alpha\}}{\Gamma\big(\sum \{\boldsymbol{c}[\boldsymbol{b}(g, c_{\bar{j}})] + \alpha\}\big)} \\
& \cdot \frac{\Gamma\{\boldsymbol{c}[\boldsymbol{P}(g, \delta_j)] + \beta_j\}}{\Gamma\big(\sum \{\boldsymbol{c}[\boldsymbol{P}(g, \delta_j)] + \beta_j\}\big)} \cdot (w_{\bar{j}} + \xi_{c1}) \\
P(c_j = 0|g, c_{\bar{j}}, D) = & \frac{1}{Z_c} \cdot (m - w_{\bar{j}} - 1 + \xi_{c2}) \\
& \cdot \frac{\Gamma\{\boldsymbol{c}[\boldsymbol{b}(g, c_{\bar{j}})] + \boldsymbol{c}[\boldsymbol{P}(g, \delta_j)] + \alpha\}}{\Gamma\big(\sum \{\boldsymbol{c}[\boldsymbol{b}(g, c_{\bar{j}})] + \boldsymbol{c}[\boldsymbol{P}(g, \delta_j)] + \alpha\}\big)}
\end{aligned} \tag{6}
$$

where δ_j is a label vector whose jth entry is 1 and other entries are 0, $w_{\bar{j}}$ is the number of condition labels that are 1 in $c_{\bar{j}}$, and Z_c is a normalization term such that $P(c_j = 1|g, c_{\bar{j}}, D) + P(c_j = 0|g, c_{\bar{j}}, D) = 1$. Intuitively, the first term in Equation 6 assumes the current prediction of the background and extends the current bicluster by treating the jth condition as one of the biclustering conditions, while the second term in Equation 6 adds the jth condition to the currently assigned background.

The algorithm

To summarize, the Gibbs biclustering procedure is

1. Initialization: Randomly assign gene labels and condition labels to either 1 or 0.
2. Fix the labels of the conditions. For every gene i, ($i = 1, 2, ..., n$), fix the labels for all the other genes, and
   (1) Calculate the Bernoulli distribution for the gene as described in Equation 4.
   (2) Draw a label for gene i from the distribution.
3. Fix the labels of the genes. For every condition j, ($j = 1, 2, ..., m$), fix the labels of all the other conditions, and

(1) Calculate the Bernoulli distribution for the condition as described in Equation 6.
(2) Draw a label for condition j from the distribution.

4. Go to Step 2 for a predefined number of iterations

From samples to the final pattern

We stated in the beginning of the section that Gibbs sampling is a Markov chain Monte Carlo method. However, so far we have only discussed the Markov chain property inherited by Gibbs sampling, which is demonstrated by its sampling procedure. The Monte Carlo property, on the other hand, comes from the way by which Gibbs sampling evaluates statistics, such as mean and variance, of the target marginal distribution by using the population quantities of the samples. Furthermore, even the marginal distribution itself can be simulated by the Monte Carlo method. For our problem, the mean of the marginal distribution of a label can be estimated by

$$\begin{aligned} \mathrm{E}[g_i|D] &= \frac{1}{T}\sum_{t=1}^{T}\mathrm{E}[g_i|g_{\bar{i}}^{(t)}, c^{(t)}, D] \\ \text{and} \quad \mathrm{E}[c_j|D] &= \frac{1}{T}\sum_{t=1}^{T}\mathrm{E}[c_j|g^{(t)}, c_{\bar{j}}^{(t)}, D], \end{aligned} \tag{7}$$

and the marginal distribution itself can be approximated by

$$\begin{aligned} \hat{p}(g_i|D) &= \frac{1}{T}\sum_{t=1}^{T}p(g_i|g_{\bar{i}}^{(t)}, c^{(t)}, D) \\ \text{and} \quad \hat{p}(c_j|D) &= \frac{1}{T}\sum_{t=1}^{T}p(c_j|g^{(t)}, c_{\bar{j}}^{(t)}, D), \end{aligned} \tag{8}$$

where t indexes the iterations and T is the total number of iterations. The final positions of the bicluster are selected as the ones where the expectations of both the gene label and the condition label are above a threshold. Then, the final pattern model is calculated as the posterior mean of the bicluster.

It is important to realize that samples simulated before the Gibbs sampler reaches its convergence should not be regarded as samples drawn from the target distribution. We examine the convergence of the Gibbs sampling procedure by monitoring the likelihood of the data. Once convergence is reached, the Gibbs sampler is run for another desired numbers of iterations to collect samples for evaluating the properties of the target posterior distributions, namely using Equations 7 and 8. To distinguish the additional sampling procedure performed after convergence is reached from the one that prepares for the convergence, we call the latter burn-in while we refer to the former as sampling.

Another important issue is that the samples produced by the Gibbs sampler at successive iterations are not independent. This dependency implies that the variance (i.e. the accuracy) of the model obtained by averaging the parameters may be much higher than if the samples were independent. The autocorrelation time is the sum of the autocorrelation values for all positive lags and its square root gives the factor by which we must increase the number of iterates of the autocorrelated estimates to obtain the same accuracy as with independent estimates. Denoting by ω_t the sets of parameters obtained at each iteration and by $\bar{\omega} = (1/T)\sum_{t=1}^{T}\omega_t$ the average set of parameters, the autocorrelation function for a lag of k can be estimated as

$$\hat{\rho}_k = \frac{\mathrm{Cov}(\omega_t, \omega_{t+k})}{\mathrm{Var}(\omega_t)} = \frac{\sum_{t=1}^{T-k}(\omega_t - \bar{\omega})(\omega_{t+k} - \bar{\omega})}{\sum_{t=1}^{T-k}(\omega_t - \bar{\omega})^2}. \tag{9}$$

In the frequent case where the autocorrelation function can be described as an autoregressive process, the autocorrelation time $\tau = \sum_{k=1}^{\infty}\hat{\rho}_k$ can be simplified to $\tau = (1 + \hat{\rho}_1)/(1 - \hat{\rho}_1)$. Such an estimate can be easily computed at the hand of the iterates of a run of the algorithm.

Data without a bicluster

To decide that a data set does not or no longer contain a bicluster, we check the number of genes or conditions that belong to the bicluster after Step 2 and Step 3 of the addressed algorithm. If either of the numbers equals zero, we reinitialize the algorithm and perform Gibbs sampling again. However, if after a predefined number of reinitializations (for example, 50 in our implementation) the algorithm still does not succeed to reach convergence, we terminate the algorithm and consider that the data set does not contain a bicluster.

RESULTS

Synthetic data

Data with one embedded bicluster. We embedded a pattern of 25 rows by 8 columns (see Fig. 2d) into a data set of size 100 by 30 (see Fig. 2b). The pattern was described by eight sharp multinomial distributions, while the background was generated from a multinomial distribution close to a uniform distribution.

We ran the algorithm for 500 iterations on the data set. The average autocorrelations (see Equation 9) for $\mathrm{E}[g_i|g_{\bar{i}}^{(t)}, c^{(t)}, D]$ and $\mathrm{E}[g_i|g_{\bar{i}}^{(t)}, c^{(t)}, D]$ are 0.0807 and 0.0404 respectively. So we decide that the number of samples is sufficient for evaluating the models. From the trace of the likelihood and from monitoring the expected values of the labels (i.e. $\mathrm{E}[g_i|D]$, and $\mathrm{E}[c_j|D]$), we observed that convergence had been reached by the end of

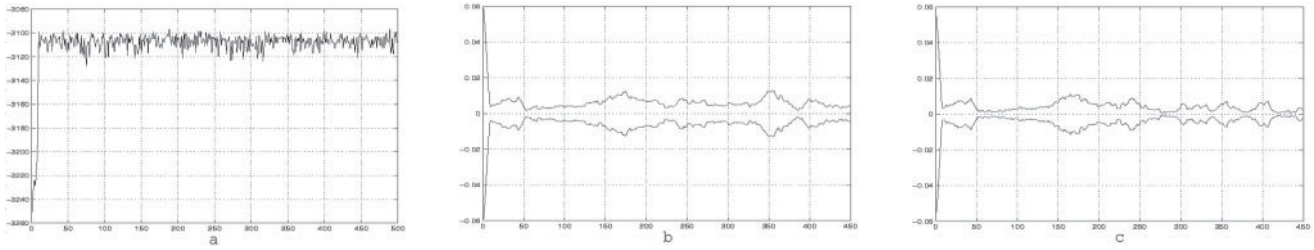

Fig. 1. (a) Trace of the log likelihood of the synthetic data evaluated at the end of each iteration during the whole Gibbs sampling procedure. (b) and (c) reflects the evolution of the expected values of respectively the row labels and the column labels. For every label, we estimated $\mathrm{E}[g_i|D]$ or $\mathrm{E}[c_i|D]$ over every possible window of 50 iterations to obtain the trace of the expected value (every trace contains thus 451 points); then we centered each trace around the mean of its last 100 points; finally we examined the variance of these centered traces across the whole set of the row (gene) labels or the column (condition) labels. Shown in (b) and (c) are the plus and minus one standard deviation.

the first 50 iterations, (see Fig. 1). Thus, we used samples drawn from the last 450 iterations to simulate the posterior distributions of the labels.

Figure 2a shows the posterior probability for each position in the data matrix that it belongs to the bicluster. The posterior probability is reflected by the brightness associated to every position in the plot, where the two extremes, white and black, imply respectively probability 1 and 0. The inner bars around the main plot in Figure 2a indicate the expected values of the labels, which can be calculated from Equation 7. The outer bars mark the embedded positions of the bicluster by a white tag. A further examination of the expected values showed that the row labels are separated into two groups, one of which includes labels whose expected values are larger than 0.7, and the other contains labels whose expected values are less than 0.4. We thus use 0.7 as the threshold for the row labels, and likewise 0.8 as the threshold for the column labels, and consider the positions the positions of the target bicluster to be the ones that possess higher expected values than the thresholds at both dimensions. The final pattern of the bicluster revealed by our algorithm is shown in Figure 2c. From these pictures we see that all the columns where the embedded pattern locates were correctly found, and most of the embedded rows were recovered.

A more detailed look showed that there was quite a variability for the biclusters retrieved at each iteration. However, these biclusters overlapped with each other most frequently at the positions of our final decision, which is reflected by Figure 2a. This is a typical characteristic of Gibbs sampling, which presents targets in terms of distributions rather than deterministic values. In this way, Gibbs sampling also avoids the problem of local maxima that often hinders Expectation–Maximization.

Data with multiple biclusters. To examine the ability of the algorithm to find multiple biclusters, especially when overlap between biclusters are present, we embedded three biclusters into a noisy background described by a distribution close to uniform distribution. The data set was of size 200 rows by 40 columns, and the three embedded biclusters are of the following size – 40 by 7 for Bicluster 1, 25 by 10 for Bicluster 2, and 35 by 8 for Bicluster 3. Figure 3 shows the data and the result, where the rows and the columns of the data set and the found biclusters are reordered for the convenience of display. (The algorithm was performed on the original data set, where the biclusters are scattered around.)

As can be seen in the main plot of Figure 3a, Bicluster 1 (located at the bottom left of the figure) overlapped with the Bicluster 2 (the middle bicluster in the figure) at two columns, and Bicluster 3 (the most top right one among the three) overlapped with Bicluster 2 at five rows and three columns.

By masking the rows of every discovered bicluster, the algorithm succeeded in finding three biclusters. The bars in Figure 3a indicate the expected values of the labels for evaluating the final positions of the biclusters. From the outer one to the inner one, the bars show the expected values of the labels in the first run, the second run (i.e. after the rows of the first bicluster found was masked), and the third run. The first bicluster found (whose pattern is shown in Fig. 3b) contains all the columns of Bicluster 3, and most of its rows. Only two rows that were embedded for Bicluster 3 were missing in the bicluster found. However, another two rows that were not designed as part of Bicluster 3 were added to the bicluster, because the patterns at these two rows happened to match the one that characterized the rest of the bicluster found. The second bicluster revealed by the algorithm (see Fig. 3c) corresponded to Bicluster 1. Again, all the columns were found back, and the rows that were neglected by by the discovered bicluster are often among the most noisy rows in Bicluster 1. The third discovered bicluster (see Fig. 3d) contained 18 out of 19 rows that were still available for Bicluster 2, (the first bicluster found included all the

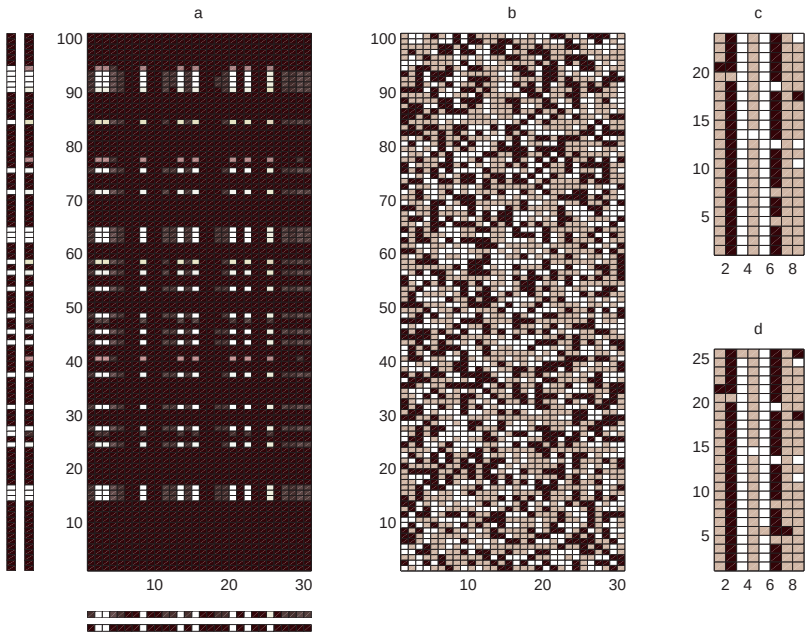

Fig. 2. Results from the synthetic data set. (a) Main plot: The posterior probability that a position of the data matrix belongs to the bicluster. Inner bars: expected values of the labels. Outer bars: positions of the embedded pattern. (b) The data matrix. (c) Pattern of the bicluster revealed by the Gibbs sampling algorithm. (d) Pattern of the embedded bicluster.

five rows at the overlapping area of Bicluster 2 and 3, in addition, it also picked up one of the rows that was designed to belong to Bicluster 2). Like the two biclusters found earlier, this final bicluster also recovered all the columns designed for Bicluster 2.

Leukemia data

We have also applied our algorithm on a leukemia data set, see (Armstrong *et al.*, 2002) for a detailed description of the data. In their paper, Armstrong *et al.* showed that differences in gene expression are robust enough to classify leukemias correctly as mixed-linkage leukemia (MLL), acute lymphoblastic leukemia (ALL) or acute myelogenous leukemia (AML). We want to explore the possibility to use our algorithm to find fingerprints of gene expression profiles for the three patient groups.

The data set consists of expression data from Affymetrix chips for 12 600 genes collected from 72 leukemia patients, of which 28 were diagnosed with ALL, 20 were MLL patients, and 24 were AML patients. In contrast with the emphasis in the theoretical presentation, the task here was to identify patients that share similar expression behavior over a subset of genes.

Because data points with low values are noisy and non-reproducible, a threshold of 100 was put on the original data. A ceiling of 1600 was also placed because of saturation effects. Next, the variation of each gene along all the patients was examined. Since the genes that have consistent behavior over all the patients are not of much interest, only the first 15 percent of genes with the highest standard deviation were selected for further analysis. In this way, the size of the data set was reduced to 1887 genes by 72 patients. This reduced data set was then discretized according to the equal frequency principle. That is, for every gene, we first put its expression data over all the patients in an ascending order, and then divided the data points into a desired number of bins, (which is 3 in the case presented below), in a way such that the number of data points in every bin is the same. Note that the use of the equal frequency principle enables the application of the one-multinomial background introduced in Equation 1. We use data from the last three patients of every category to construct a test data set. Data from the rest of the patients were used as a training set.

By masking the patients found after each run, the algorithm succeeded in discovering three biclusters one after another for the training data set. The algorithm stopped after discovering three biclusters one after another for the training data set. Figure 4 demonstrates the ability of our algorithm to group patients based on their expression behavior over a subset of genes. Furthermore, the patients collected in every bicluster came from the same category. More specifically, (a) the first bicluster selected 19 patients all of whom are out of the 25 AML

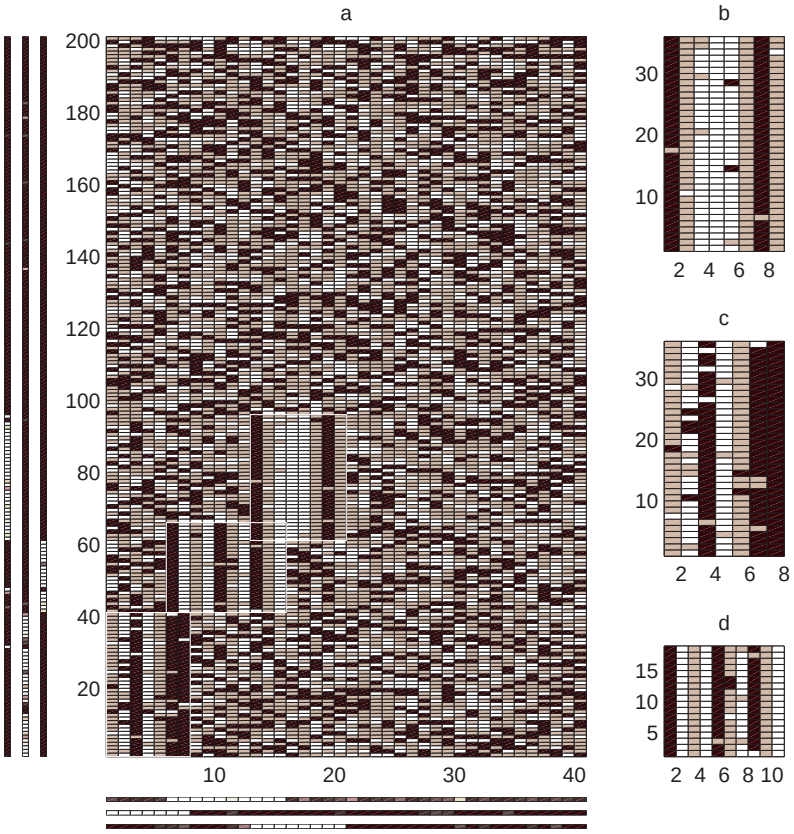

Fig. 3. (a) Main plot: The data set where three biclusters are embedded. The bars indicate the expected values of the labels for deciding the positions of the biclusters discovered. (b), (c) and (d) Three biclusters found by masking the rows of a found bicluster.

patients in the training set, and 80 genes; (b) the second bicluster included 18 (out of 21) ALL patients, and 87 genes; (c) the third bicluster consisted of 14 (out of 17) MLL patients and 62 genes.

In Figure 5, we illustrate the idea that the pattern of the genes of a bicluster discovered by our algorithm provides a signature for identifying patients of the particular patient class featured by the bicluster. The scores in the figure are calculated as the log ratio of the likelihood that the data points at those genes are generated by the pattern model versus the likelihood that they are drawn from the background model. One can see that all the patients of the category characterized by the bicluster (including those that were not detected by the bicluster) had higher scores than those of the rest of the patients, with one exception for the score of the 38th patient obtained under pattern (c). More importantly, in the test data set, patients of the correct category were also associated with much higher scores than the patients of the other categories. This result shows that the patterns discovered by our algorithm can be used directly for prediction.

To test the significance of the found patterns, we performed the algorithm on 100 permutated data sets of the test data. Tests were done under three sets of pseudocounts (see Equation 3), representing the prior knowledge of three levels (i.e. high, medium, and low) of noisiness in the desired pattern. No pattern was found for any of the data sets under any setting of the pseudocounts. By this we mean that for every iteration in the tests, a small bicluster (often consisting of only one patient and several genes) was sampled at most iterations but that, if we look at the evolution of the bicluster throughout all the iterations, the revealed biclustering positions ambled around and did not have a consistent core. This result demonstrates that the patterns found by Gibbs biclustering are statistically highly significant.

DISCUSSION

We have introduced Gibbs sampling method to the biclustering problem of discretized microarray data and have demonstrated the effectiveness of our approach on two synthetic data sets and a real-life data set of leukemia pa-

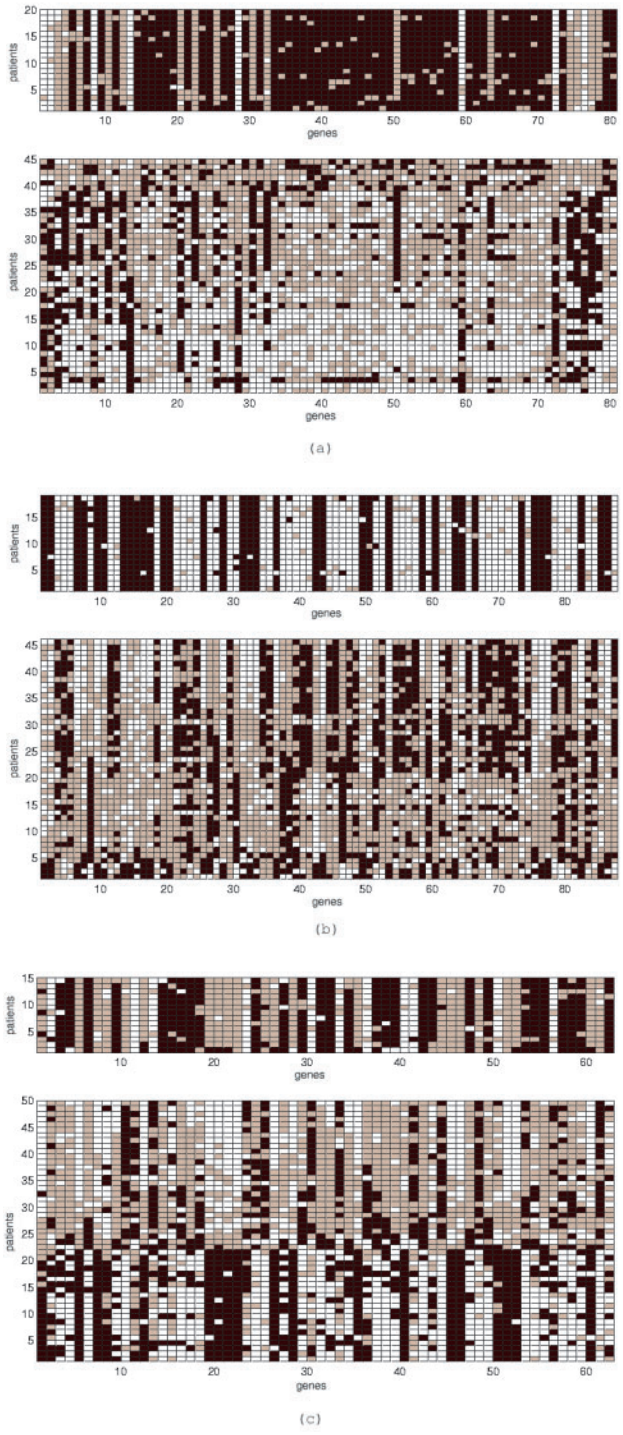

Fig. 4. Three biclusters featuring (a) AML patients, (b) ALL patients, and (c) MLL patients. Top: Patterns of the bicluster. Bottom: Discretized data of the genes in the bicluster from the patients who were not selected by the bicluster.

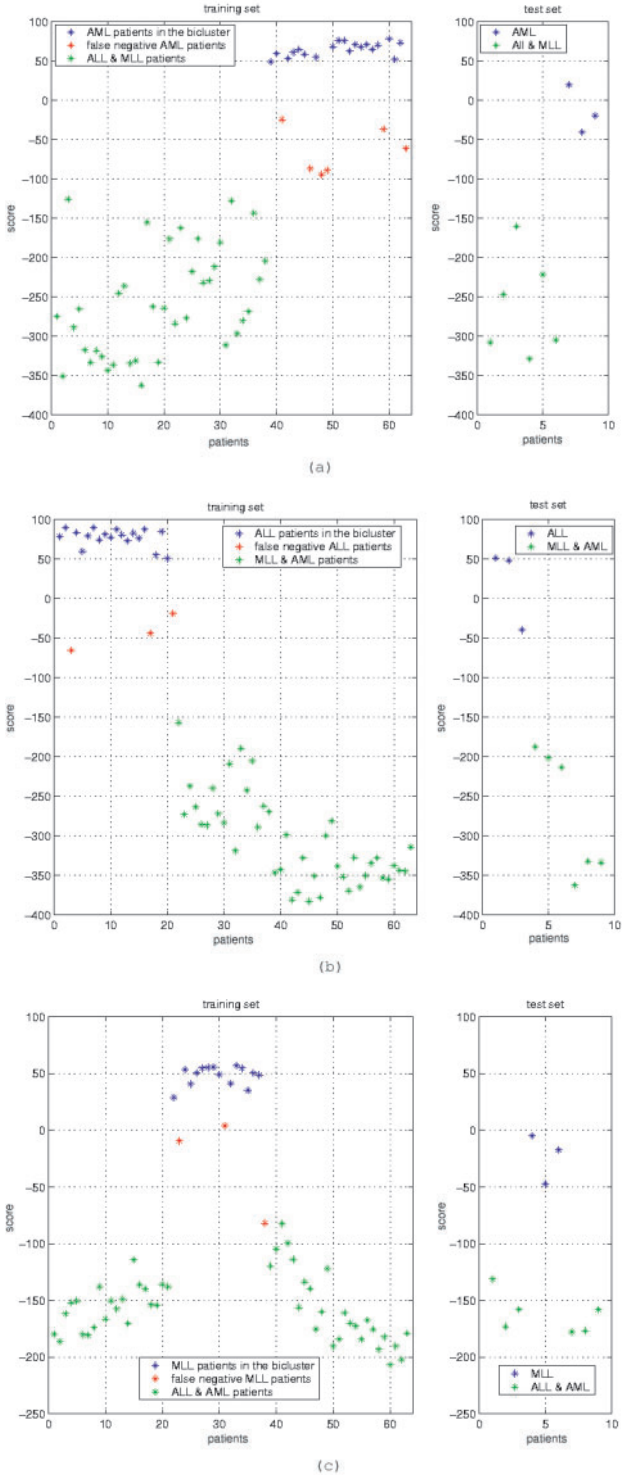

Fig. 5. Scores of the patients in both the training data set and the test data set. The scores are calculated using the pattern models of the biclusters characterizing (a) AML patients, (b) ALL patients, and (c) MLL patients.

tients. We showed that our method detects biclusters that are statistically contrasted with the noisy background. We examine this contrast at every iteration by evaluating the model of the pattern as the posterior mean of the data points in a bicluster, (see Equation 5), which infers that the prior knowledge (represented by the pseudocounts) as well as the sample mean of the data points in the bicluster participate in the evaluation of the pattern model. Our method fits in the Bayesian framework so that the power of the prior knowledge fades as the number of data points in the bicluster increases. This explains why our algorithm discovers relatively large biclusters. Small biclusters with a consistent pattern (e.g. a bicluster of the size 2×2 out of a data set of the size 100×20) are neglected (because of the regularization caused by the pseudocounts), which protects our algorithm from specious false-positive biclusters.

We have developed our approach for discretized microarray data, which is an acceptable trade off given the high level of noise observed on microarray (this noise decreases the importance that must be given to detailed measurement values). In Section Results, we used three discretization bins for all the data sets. However, this does not mean that three is the value of choice for the number of bins to be used in discretization. Users can change it to a value suitable for their data set. For the leukemia data set we analyzed, however, we found that the final indices of the biclusters was not much affected by different choices of discretization levels. A generalization for the approach to accommodate continuous data is possible, but the choice of continuous distribution (e.g. Gaussian noise model) should be carefully investigated and this is a further topic of our research.

We introduced the background model as a single multinomial distribution (see Section 'model and assumptions', Equation 1), which is suitable for data sets where the background data shares the same distribution under every condition. However, if distribution of the background data under each condition differs significantly from each other, it might be better to use several multinomial distributions, each of which describes the background under an individual condition. The background model will then look like the pattern model described in Equation 2. Extending the formulae of the algorithm to accommodate a background described by several multinomial distributions is possible. Nevertheless, for the patient-gene orientation that was introduced in Section 'leukemia data', the condition for applying the one multinomial background model can be easily achieved by applying the equal frequency principle to the discretization of the data under each gene.

Our algorithm has a computational requirement on the ratio of the row size and the column size of the data set being analyzed. Here the row dimension refers to either the gene dimension in the gene-condition orientation (introduced at the beginning of the paper) or the patient dimension in the patient-gene orientation (as addressed in Section 'leukemia data'). Observe that the row dimension is actually the dimension along which the multinomial models are evaluated, while the column dimension is where feature selection is performed. In general, our algorithm works well on the gene-condition orientation where the row dimension is usually much larger than the column dimension. However, when working on the patient-gene orientation, the algorithm will meet computational difficulties if the number of patients is too low compared to the number of genes. An example is that the algorithm could not perform correctly on a data set of 22 patients with some 3000 genes, while the problem no longer appeared once the number of patients was increased to 60.

REFERENCES

Armstrong,S.A., Staunton,J.E., Silverman,L.B., Pieters,R., den Boer,M.L., Minden,M.D., Sallan,S.E., Lander,E.S., Golub,T.R. and Korsmeyer,S.J. (2002) MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nat. Genet.*, **30**, 41–47.

Ben-Dor,A., Friedman,N. and Yakhini,Z. (2001) Class discovery in gene expression data. *Proc. 5th Annual Intl. Conf. on Computational Biology*, **5**, 31–38.

Busygin,S., Jacobsen,G. and Krämer,E. (2002) Double conjugated clustering applied to leukemia microarray data. *2nd SIAM ICDM, Workshop on Clustering High Dimensional Data*.

Casella,G. and George,E.I. (1992) Explaining the Gibbs sampler. *J. Amer. Stat.*, **46**, 167–174.

Cheng,Y. and Church,G.M. (2000) Biclustering of expression data. *ISMB 2000 proceedings*, 93–103.

Getz,G., Levine,E. and Domany,E. (2000) Coupled two-way clustering analysis of gene microarray data. *Proc. Natl Acad. Sci. USA*, **97**, 12079–12084.

Lawrence,C.E., Altschul,S.F., Boguski,M.S., Liu,J.S., Neuwald,A.F. and Wooton,J.C. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignments. *Science*, **262**, 208–214.

Lazzeroni,L. and Owen,A. (1999) Plaid models for gene expression data. *Technical report, Stanford University, Statistics*.

Liu,J.S., Neuwald,A.F. and Lawrence,C.E. (1995) Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *J. Amer. Stat.*, **90**, 1156–1170.

Segal,E., Taskar,B., Gasch,A., Friedman,N. and Koller,D. (2001) Rich probabilistic models for gene expression. *Bioinformatics*, **17**, 243S–249S.

Tanay,A., Sharan,R. and Shamir,R. (2002) Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, **18**, 136S–144S.

Thijs,G., Marchal,K., Lescot,M., Rombauts,S., De Moor,B., Rouzé,P. and Moreau,Y. (2002) A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *J. Comput. Biol.*, **9**, 447–464.