Chapter 4

Gibbs sampling for biclustering problems

Clustering techniques provide a global view of the structure of microarray data, revealing genes that share the same expression profile, or grouping experiments according to their associated gene expression values. While the information extracted by these clustering algorithms can be interesting for biologists when they have little idea about the function of the genes under study, or about (when clustering patients, for example) what the classifications of the patients are. However, this type of analysis do not provide insights to those specific questions that biologists ask. One of the most intriguing questions for biologists is, for example, "what are the genes that are related to a particular function (or in a specific pathway) of interest to me?" A more ideal clustering algorithm in this regard should allow input identifying the target genetic function, and direct the clustering result toward the discovery of genes related to the function.

In addition, the increasingly overwhelming amount of data from heterogenous source (besides mRNA level data, e.g. DNA sequence information, protein structural information) calls out the need for analysis that can integrate these biological insights from different viewpoints. To this end, clustering algorithms that are compatible for such integration are favorable.

Clustering algorithms based on Bayesian probabilistic models show promise to be suitable candidates meeting these requirements. First of all, Bayesian probabilistic models inherits general probabilistic models to reveal the fundamental structure that biologists seek in microarray data [67, 66] (see Chapter 3. Secondly, the hierarchical nature in Bayesian inferencing allow the introducing of prior knowledge, which helps biologists to impose specific questions to the analysis of gene expression data. Finally, Bayeisan probabilistic models provide reliable and interpretable platforms for the integration of the analysis of gene expression data with other sources of biological information [68, 28].

Seeing these advantages, we put the biclustering problem of microarray data in the Bayesian context. As we explained in Chapter 3, the biclustering problem that we consider is to identify genes that behave similarly only over a subset of conditions. Or for the other orientation of microarray data, the aim of the biclustering algorithm is to group experiments (e.g., patients) under each of which a sub set of genes have almost the same expression values. To distinguish these two types of biclustering problems, we refer to the former as "biclustering genes", and the latter as "biclustering experiments". We use Bayesian probabilistic models to describe both the data under the bicluster and the rest of the data (i.e., the background data). Gibbs sampling will be used for inferring the parameters of the models as well as the structure of the data.

Gibbs sampling is a technique to draw samples from a join distribution based on the full conditional distributions of all the associated random variables. Though the idea roots back to the work of Hasting (1970) [37], whose focus was on its Markov chain Monte Carlo (MCMC) nature, the Gibbs sampler was first formally introduced by Geman and Geman (1984) [32] to the field of image processing. The work caught the attention of the statistics society (especially boosted by the paper of Gelfand and Smith (1992) [30]). Since then, the applications of Gibbs sampling have covered both the Bayesian world and the world of classical statistics. In the former case, Gibbs sampling is often used to estimate posterior distributions, and in the latter, it is often applied to likelihood estimation [16].

In particular, Gibbs sampling has become a popular alternative to the expectation-maximization (EM) for solving the incomplete-data problem, where the associated random variables of interest include both the hidden variables (i.e., the unobserved data) and the parameters of the model that describe the complete data. The biclustering problem belongs exactly to this category, where the unobserved data is the partition of the microarray data into the bicluster and the background.

To provide answers to the incomplete-data problem, EM is a numerical maximization procedure that climbs in the likelihood landscape aiming to find the model parameters and the hidden variables that maximize the likelihood function. It iterates between the following two steps [21], (1) assuming the model parameters to be known, it calculates the expected value of the hidden variables, (2) with the expected values of the the hidden variables, it finds the model parameter that maximizes the complete likelihood. The produre is guaranteed to converge. However, the obtained solution of EM often gets stuck at the local maximum mode of the likelihood function. To alleviate the problem, multiple runs of EM procedure with independent initializations are often performed, and the solution with the highest likelihood is selected.

In contrast, Gibbs sampling provides the means to estimate the target joint distribution of the hidden variables and the model parameters as a whole, and bases the estimation of the random variables (i.e. both the hidden variables and the model parameters) on the estimated joint distribution, where maximum a posterior (MAP) estimates are often used. Thus, Gibbs sampling suffers less

from the problem of local maxima than EM.

This property makes Gibbs sampling a suitable candidate for solving the model-based problems in bioinformatics, where the likelihood function usually consists of a large amount of modes due to the high complexity of the data. The Gibbs sampling strategy was first introduced to the field of bioinformatics for its applications to the motif-finding problem [50] in DNA sequence analysis, and has become the method-of-choice for the problem [78]. Our idea to apply Bayesian strategy to the biclustering problem of microarray data was inspired by its success in the motif-finding problem.

In this chapter, we first overview the working mechanism of Gibbs sampling. Then, we introduce the concept of Bayesian probabilistic models, and further construct the general Bayesian hierarchical model for the biclustering problem. Finally, we highlight the Gibbs sampling scheme applied to the Bayesian hierarchical model.

4.1 Gibbs sampling

Gibbs sampling allow statisticians to avoid the tedious and sometimes non-trivial mathematical calculations of integrals in obtaining the join distribution, by sampling directly from the full conditional distributions. (Because the same mechanism applies to both models for discrete data and models for continuous data, we use the terms "distribution" and "density" interchangeably). Suppose that we want to draw samples for the set of random variables $x_1, x_2, \ldots, x_K$, but that the marginal distributions (and thus the joint distribution) are too complex to directly sample from. Suppose also that the full conditional distributions $p(x_i \,|\, x_j;\, j \neq i)$ (for $i = 1, \ldots, K$) which can easily be sampled from, are available. Starting from initial values $x_1^{(0)}, x_2^{(0)}, \ldots, x_K^{(0)}$, the Gibbs sampler draws samples of the randome variables in the following manner,

$$
\begin{aligned}
x_1^{(t+1)} &\sim p(x_1 \,|\, x_2 = x_2^{(t)}, \ldots, x_K = x_K^{(t)}) \\
x_2^{(t+1)} &\sim p(x_2 \,|\, x_1 = x_1^{(t+1)}, x_3 = x_3^{(t)}, \ldots, x_K = x_K^{(t)}) \\
&\vdots \qquad\quad \vdots \\
x_i^{(t+1)} &\sim p(x_i \,|\, x_1 = x_1^{(t+1)}, \ldots, x_{i-1} = x_{i-1}^{(t+1)}, x_{i+1} = x_{i+1}^{(t)}, \ldots, x_K = x_K^{(t)}) \\
&\vdots \qquad\quad \vdots \\
x_k^{(t+1)} &\sim p(x_K \,|\, x_1 = x_1^{(t+1)}, \ldots, x_{K-1} = x_{K-1}^{(t+1)}),
\end{aligned}
$$

where t denotes the iterations.

Geman and Geman (1984) [32] shows that as $t \to \infty$, the distribution of $(x_1^{(t)}, \ldots, x_k^{(t)})$ converges to $p(x_1, \ldots, x_K)$. Equivalently, as $t \to \infty$, the distribution of $x_i^{(t)}$ converges to $p(x_i)$ (for $i = 1, \ldots, K$).

4.1.1 The Markov chain property

The convergence of samples drawn by the Gibbs sampler relies on the fact that these samples form Markov chains. In other words,

$$\left((x_1^{(1)}, \ldots, x_k^{(1)}), \ldots, (x_1^{(t)}, \ldots, x_k^{(t)})\right)$$

as well as

$$(x_i^{(1)}, \ldots, x_i^{(t)})$$

are Markov chains, where $(x_1^{(t)}, \ldots, x_k^{(t)})$ and $x_i^{(t)}$ are called the states of $(x_1, \ldots, x_k)$ and x_i respectively. The basic property of a Markov chain, take that of x_i for example, is

$$P(x_i^{(t+1)} \,|\, x_i^{(t)}, \ldots, x_i^{(0)}) = P(x_i^{(t+1)} \,|\, x_i^{(t)}), \tag{4.1}$$

which means that the future state of a random variable depends only on its current state but not on its past states. Writing

$$\begin{aligned} \pi_b(t+1) &= p(x_i^{(t+1)} = b) \\ \pi_a(t) &= p(x_i^{(t)} = a) \\ \text{and} \quad p(a \to b) &= p(x_i^{(t+1)} = b \,|\, x_i^{(t)} = a), \end{aligned}$$

we have

$$\pi_b(t+1) = p(a \to b)\pi_a(t). \tag{4.2}$$

$p(a \to b)$ is called the transition probability of going from state a to b (for random variable x_i). The probability transition matrix $\mathbf{P}$ is obtained by listing all the possible states for x_i along the rows and the columns, and fill the matrix with all the transition probabilities. (Note that this implies that each row of $\mathbf{P}$ sums to 1.) Thus to generalize Equation 4.2, we have

$$\pi(t+1) = \mathbf{P}\,\pi(t). \tag{4.3}$$

It is known that if the all the entries of $\mathbf{P}$ are above 0, an evolving Markov chain will reach a stationary distribution π^* after a sufficient amount of time [16], i.e.,

$$\pi^* = \mathbf{P}\,\pi^*. \tag{4.4}$$

Casella and George (1992) [16] gives a simple yet intuitive proof that the stationary distributions of the Markov chains generated by Gibbs sampling are the joint distribution $p(x_1, \ldots, x_k)$ and the marginal distributions $p(x_i)$, and that the probability transition matrices of these Markov chains can be derived from the full conditional distributions.

4.1.2 The Monte Carlo property

Only those samples collected by the Gibbs sampler after the convergence is reached can be used for joint (or marginal) distribution estimation. The Gibbs

sampling procedure performed before the convergence is reached is often referred to as the "burn-in procedure", and the procedure during which samples are collected will be called the "sampling procedure" hereafter. The samples collected in the sampling procedure enable us to calculate the expectation of a function $f(x_i)$ over the distribution $p(x_i)$. This is done by the Monte Carlo integration

$$\mathrm{E}_{p(x_i)}[f(x_i)] = \int f(x_i) \cdot p(x_i)\mathrm{d}x \approx \frac{1}{T}\sum_{t=1}^{T} f(x_i^{(t)}), \tag{4.5}$$

where t indexes the iterations in the sampling procedure, and T is the total number of samples collected. Thus, the expected value of x_i can be calculated as

$$\mathrm{E}_{p(x_i)}[x_i] = \int x_i \cdot p(x_i)\mathrm{d}x \approx \frac{1}{T}\sum_{t=1}^{T} x_i^{(t)}. \tag{4.6}$$

However, as illustrated by Gelfand and Smith (1990) [30] (using the Rao-Blackwell theorem), a more accurate estimate of the expected value of x_i is provided by

$$\mathrm{E}_{p(x_i)}[x_i] = \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}_{p(x_i \mid x_1^{(t)}, \ldots, x_{i-1}^{(t)}, x_{i+1}^{(t)}, \ldots, x_k^{(t)})}[x_i]. \tag{4.7}$$

Similarly, the posterior distribution itself can be approximated by

$$\mathrm{E}[p(x_i)] = \frac{1}{T}\sum_{t=1}^{T} p(x_i \mid x_1^{(t)}, \ldots, x_{i-1}^{(t)}, x_{i+1}^{(t)}, \ldots, x_k^{(t)}). \tag{4.8}$$

With more generality, a better alternative for Equation 4.5 is

$$\mathrm{E}_{p(x_i)}[f(x_i)] = \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}_{p(x_i \mid x_1^{(t)}, \ldots, x_{i-1}^{(t)}, x_{i+1}^{(t)}, \ldots, x_k^{(t)})}[f(x_i^{(t)})]. \tag{4.9}$$

The estimators obtained by Monte Carlo integration are unbiased estimators.

4.1.3 Checking the convergence

A key issue in using Gibbs sampling is to determine when the procedure has essentially converged. The number of iterations needed for the burn-in procedure varies from cases to cases. For a well mixed Markov chain whose samples cover most of the region of the random variable space, the convergence can be reached with in a few iterations. However, a bad starting point plus a multimodal target distribution with some of its probabilities close to zero can result in a poorly mixed chain so that only a small region of the random variable space is sampled for a long period of time. In this case, the number of burn-in iterations can easily reach a few thousand. In general, an optimal starting point close to the center

of the marginal distribution can help in the accelerating the convergence. In addition, using multiple chains starting at independent positions of the random variable space can help to increase the coverage of the samples [31] and thus alleviate the problem of poorly mixed chains. Yet, convergence diagnostics are favorable in assisting the decision. Informal procedures of convergence diagnostics include inspecting the trace plot of the concerned variables or the evolution of likelihood. Various formal procedures has also been proposed. A good review on this issue is provided by Cowles and Carlin (1996) [20]. In the rest of this section, we will focus on one of the important issues for the convergence of a Markov chain—the autocorrelation.

One of the reasons that a Markov chain generated by the Gibbs sampler has a slow convergency is that the samples 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 $\boldsymbol{\omega}^{(t)}$ the sets of parameters obtained at each iteration and by $\bar{\boldsymbol{\omega}} = (1/T)\sum_{t=1}^{T}\boldsymbol{\omega}^{(t)}$ the average set of parameters, the autocorrelation function for a lag of H can be estimated as

$$\hat{\rho}_m = \frac{\mathrm{Cov}(\boldsymbol{\omega}^{(t)}, \boldsymbol{\omega}^{(t+H)})}{\mathrm{Var}(\boldsymbol{\omega}^{(t)})} = \frac{\sum_{t=1}^{T-H}(\boldsymbol{\omega}^{(t)} - \bar{\boldsymbol{\omega}})(\boldsymbol{\omega}^{(t+H)} - \bar{\boldsymbol{\omega}})}{\sum_{t=1}^{T-H}(\boldsymbol{\omega}^{(t)} - \bar{\boldsymbol{\omega}})^2}. \tag{4.10}$$

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.

Another way to reduce the autocorrelation is to use the thinning of the Markov chain. Thinning with a factor J means that each J^{th} element in the chain will be used for the posterior summary statistics (see Equation 4.9). Another computational advantage of using the thinning procedure is that it saves the memory complexity of the Gibbs sampling procedure.

4.2 Bayesian probabilisitc models for biclustering microarray data

While the probability in classical probability and statistics theory measures the physical property of the chance that a random event would happen (e.g. the "head side" of a coin faces upwards after being tossed), the probability in the Bayesian probabilistic and statistics theory represents rather our *belief* that this certain random event would happen [38]. The purpose of the classical statistics

is to estimate this physical probability. But Bayesian statistics concerns more about how our belief changes after we obtain a certain amount of information.

This updating procedure of our belief—called Bayensian inference—is expressed mathematically in the following equation,

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta) \cdot p(\theta)}{p(\mathcal{D})}, \tag{4.11}$$

where θ is the random variable of interest, $p(\theta)$ represents our prior knowledge (i.e., our belief before observing the data); $\mathcal{D}$ represents the data, $p(\mathcal{D} \mid \theta)$ is the likelihood of data; and $p(\mathcal{D})$ is called the evidence that provides the normalization in order to quantitize our posterior belief $p(\theta \mid \mathcal{D})$ between 0 and 1. The evidence can be obtained by

$$p(\mathcal{D}) = \int p(\mathcal{D} \mid \theta) \cdot p(\theta) \mathrm{d}\theta. \tag{4.12}$$

In this way, the Bayesian inference captures casual relations presented in the data by the term for likelihood and combines it with the prior knowledge. Thus, Bayesian inference is a learning procedure that is very close to the human learning process.

A simple Bayesian hierarchical model comprises of three levels—from bottom up, the data, the model of the data, and the prior of the model. In a more complicated case, more levels of hierarchy can be built, and a network (called Bayesian network) can be thus formed, with each higher level modelling the random variables at its immediate lower level (for a tutorial on Bayesian networks, see [38]). But for our problem, the simple model of three layers suffices. Both the data and the model parameters are treated as random variables. But the priors of the model are treated as hyperparamters, which means that the parameters of the prior will not be inferred and are used as user input in the implementation of the method.

As we mentioned, biclustering can be applied to both orientations of a microarray data matrix—i.e., biclustering the genes and biclustering the experiments. In this chapter, we generalize the problem by treating both of the problems as biclustering the rows of a matrix, so as to illustrate the frame of the Data model and the Gibbs sampling scheme. That is to say, in the case for biclustering experiments, we transpose the matrix, and the generalized problem is to find a set of rows in a matrix, whose data entries under each selected column (for the bicluster) are similar (see Figure 4.1 for an illustration).

The biclustering problem belongs to the incomplete data problem. The microarray data are the observed data, and the unobserved data are the positions of the biclusters. To begin with, we assume that there is only one bicluster in the data, and we refer to the data at the rest of the positions as the background. Given the position of the bicluster, the microarray data (denoted as $\mathbf{D}$ here after) can be modelled by a bicluster model ($\mathcal{M}^{\mathrm{bcl}}$) for the data in the bicluster and a background model ($\mathcal{M}^{\mathrm{bgd}}$). $\mathcal{M}^{\mathrm{bcl}}$ and $\mathcal{M}^{\mathrm{bgd}}$ and their priors will be discussed in more details in Chapter 5 and Chapter 6.

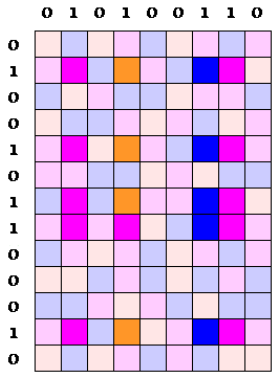


Figure 4.1: Conceptual plot of the biclustering problem: colors in the matrix represent different values. The data of the bicluster is highlighted. Binary labels are used to indicate if the row or the column belongs to the bicluster.

For the unobserved data, we use binary random variables (called labels here after) to describe the association of the rows and columns to the bicluster—i.e., a "1" indicates that the row (or the column) belongs to the bicluster and a "0" otherwise. The binary labels are divided into two sets, $\{\mathcal{R} \,|\, r_i,\, i = 1 \dots N\}$ for the rows (where N is the number of rows), and $\{\mathcal{C} \,|\, c_j,\, j = 1 \dots M\}$ for the columns (where M is the number of columns).

The natural choice for the distributions of the labels are the Bernoulli distributions

$$\begin{aligned} p(r_i) &\sim \text{Bernoulli}(\lambda_\text{r}) \quad r_i \in \mathcal{R}, \\ p(c_j) &\sim \text{Bernoulli}(\lambda_\text{c}) \quad c_i \in \mathcal{C}, \end{aligned} \tag{4.13}$$

where λ_r and λ_c are the parameters for the Bernoulli distributions (respectively for the row labels and column labels). The labels are assumed to be independently distributed given the Bernoulli parameters. Conjugate priors are used for the Bernoulli distributions, which are Beta distributions

$$\begin{aligned} p(\lambda_\text{r}) &\sim \text{Beta}(\xi_{\text{r}0},\, \xi_{\text{r}1}), \\ p(\lambda_\text{c}) &\sim \text{Beta}(\xi_{\text{c}0},\, \xi_{\text{c}1}). \end{aligned} \tag{4.14}$$

4.3 Gibbs sampling for the biclustering problem

The whole set of involved random variables of the biclustering problem are hence the microarray data $\mathbf{D}$, the parameters of the model of the data $\mathcal{M}^\text{bcl}$ and $\mathcal{M}^\text{bgd}$, the labels $\mathcal{R}$ and $\mathcal{C}$, and the Bernoulli parameters for the labels λ_r and λ_c. The task of the Gibbs sampling procedure is to infer the joint distribution of the unobserved data and the parameters of the involved models from the observed

data, i.e. the posterior joint distribution

$$p(\mathcal{R},\, \mathcal{C},\, \lambda_{\mathrm{r}},\, \lambda_{\mathrm{c}},\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}},\, |\, \mathbf{D}). \tag{4.15}$$

To carry out the Gibbs sampling strategy, full conditional distributions of the labels and the model parameters need to be derived. We will look into the details of the full conditional distributions for $\mathcal{M}^{\mathrm{bcl}}$ and $\mathcal{M}^{\mathrm{bgd}}$ in Chapter 5 and Chapter 6 because they are problem dependent. In this chapter, we consider the full conditional distribution of the labels $\mathcal{R}$ and $\mathcal{C}$, which turn out to have some common interpretations for both of the biclustering problems (i.e., for biclustering genes and for biclustering experiments).

First of all, using Gibbs sampling to obtain the target joint distribution in Equation 4.15 means that we also have to sample from the complete conditional distributions of λ_{r} and λ. Because conjugate priors are used for these two parameters, the posterior distributions (i.e., the full conditional distributions) will be in the same form as the prior—Beta distributions. To simplify the Gibbs sampling procedure, we show in the following that λ_{r} and λ_{c} can be integrated out of the target distribution, based on the observation that

$$\begin{aligned} p(\mathcal{R},\, \mathcal{C},\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}}\,|\,\mathbf{D}) = & \int\int p(\mathcal{R},\, \mathcal{C},\, \lambda_{\mathrm{r}},\, \lambda_{\mathrm{c}},\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}}\,|\,\mathbf{D})\,\mathrm{d}\lambda_{\mathrm{r}}\,\mathrm{d}\lambda_{\mathrm{c}} \\ = & \frac{1}{p(\mathbf{D})} \cdot p(\mathbf{D}\,|\,\mathcal{R},\, \mathcal{C},\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}}) \\ & \cdot p(\mathcal{M}^{\mathrm{bcl}}) \cdot p(\mathcal{M}^{\mathrm{bgd}}) \\ & \int \cdot p(\mathcal{R}\,|\,\lambda_{\mathrm{r}}) \cdot p(\lambda_{\mathrm{r}})\,\mathrm{d}\lambda_{\mathrm{r}} \cdot \int p(\mathcal{C},\, |\,\lambda_{\mathrm{c}}) \cdot p(\lambda_{\mathrm{c}})\lambda_{\mathrm{c}}. \end{aligned}$$

Consequently, the target joint distribution becomes

$$p(\mathcal{R},\, \mathcal{C},\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}},\, |\, \mathbf{D}). \tag{4.16}$$

Aiming at Equation 4.16, the full conditional distribution of a row label is

$$\begin{aligned} & p(r_i = 1\,|\,\mathcal{R}_{\bar{i}},\, \mathcal{C},\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}}, \mathbf{D}) \\ = & \frac{p(\mathbf{D},\, r_i = 1,\, \mathcal{R}_{\bar{i}},\, \mathcal{C}\,|\,\mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}},\, \lambda_{\mathrm{r}},\, \lambda_{\mathrm{c}})}{p(\mathbf{D},\, r_i = 1,\, \mathcal{R}_{\bar{i}},\, \mathcal{C}\,|\,\mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}}) + p(\mathbf{D},\, r_i = 0,\, \mathcal{R}_{\bar{i}},\, \mathcal{C}\,|\,\mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})} \\ = & \frac{\gamma_i^r}{1+\gamma_i^r} \quad i = 1 \ldots N \end{aligned} \tag{4.17}$$

where $\mathcal{R}_{\bar{i}} = \{r_k\,|\,k \neq i \wedge r_k \in \mathcal{R}\}$ (which is the set of row labels except r_i), and γ_i^r is the likelihood ratio between the case that the i^{th} row is generated by $\mathcal{M}^{\mathrm{bcl}}$ and the case when it is generated by $\mathcal{M}^{\mathrm{bgd}}$,

$$\gamma_i^r = \frac{p(\mathbf{D},\, r_i = 1,\, \mathcal{R}_{\bar{i}},\, \mathcal{C}\,|\,\mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})}{p(\mathbf{D},\, r_i = 0,\, \mathcal{R}_{\bar{i}},\, \mathcal{C}\,|\,\mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})} \quad i = 1 \ldots N. \tag{4.18}$$

Similarly, the conditional distribution of a column label is also in the form of likelihood ratio,

$$p(c_j = 1 \,|\, \mathcal{C}_{\bar{j}},\, \mathcal{R},\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}}, \mathbf{D}) = \frac{\gamma_j^c}{1+\gamma_j^c} \quad c_j = 1 \dots M \tag{4.19}$$

where $\mathcal{C}_{\bar{j}} = \{c_k \,|\, k \neq j \,\wedge\, c_k \in \mathcal{C}\}$, and

$$\gamma_j^c = \frac{p(\mathbf{D},\, \mathcal{R},\, c_j = 1,\, \mathcal{C}_{\bar{j}} \,|\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})}{p(\mathbf{D},\, \mathcal{R},\, c_j = 0,\, \mathcal{C}_{\bar{j}} \,|\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})} \quad j = 1 \dots M \tag{4.20}$$

is the likelihood ratio between the case that the column is generated by $\mathcal{M}^{\mathrm{bcl}}$ against the case that the column is generated by $\mathcal{M}^{\mathrm{bgd}}$.

Observe that, for Equation 4.18

$$\begin{aligned}
\gamma_i^r =& \frac{p(\mathbf{D} \,|\, r_i = 1,\, \mathcal{R}_{\bar{i}},\, \mathcal{C}\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})}{p(\mathbf{D} \,|\, r_i = 0,\, \mathcal{R}_{\bar{i}},\, \mathcal{C}\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})} \\
& \cdot \frac{\int p(r_i = 1 \,|\, \lambda_{\mathrm{r}})\, p(\mathcal{R}_{\bar{i}} \,|\, \lambda_{\mathrm{r}})\, \mathrm{d}\lambda_{\mathrm{r}} \cdot \int p(\mathcal{C} \,|\, \lambda_{\mathrm{c}})\, \mathrm{d}\lambda_{\mathrm{c}}}{\int p(r_i = 0 \,|\, \lambda_{\mathrm{r}})\, p(\mathcal{R}_{\bar{i}} \,|\, \lambda_{\mathrm{r}})\, \mathrm{d}\lambda_{\mathrm{r}} \cdot \int p(\mathcal{C} \,|\, \lambda_{\mathrm{c}})\, \mathrm{d}\lambda_{\mathrm{c}}} \\
=& \frac{p(\mathbf{D} \,|\, r_i = 1,\, \mathcal{R}_{\bar{i}},\, \mathcal{C}\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})}{p(\mathbf{D} \,|\, r_i = 0,\, \mathcal{R}_{\bar{i}},\, \mathcal{C}\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})} \cdot \frac{\int p(r_i = 1 \,|\, \lambda_{\mathrm{r}})\, p(\mathcal{R}_{\bar{i}} \,|\, \lambda_{\mathrm{r}})\, \mathrm{d}\lambda_{\mathrm{r}}}{\int p(r_i = 0 \,|\, \lambda_{\mathrm{r}})\, p(\mathcal{R}_{\bar{i}} \,|\, \lambda_{\mathrm{r}})\, \mathrm{d}\lambda_{\mathrm{r}}}.
\end{aligned} \tag{4.21}$$

With the rest of the labels (i.e. all but the i^{th} row label and all the column labels) fixed

$$\begin{aligned}
& \int p(r_i = 1 \,|\, \lambda_{\mathrm{r}})\, p(\mathcal{R}_{\bar{i}} \,|\, \lambda_{\mathrm{r}})\, \mathrm{d}\lambda_{\mathrm{r}} \\
=& \int_0^1 \lambda_{\mathrm{r}} \cdot \lambda_{\mathrm{r}}^{v_{\bar{i}}} \cdot (1-\lambda_{\mathrm{r}})^{(N-1-v_{\bar{i}})} \cdot \frac{\lambda_{\mathrm{r}}^{\xi_{\mathrm{r}1}-1}(1-\lambda_{\mathrm{r}})^{\xi_{\mathrm{r}0}-1}}{B(\xi_{\mathrm{r}0}, \xi_{\mathrm{r}1})}\, \mathrm{d}\lambda_{\mathrm{r}} \\
=& \int_0^1 \frac{\lambda_{\mathrm{r}}^{1+v_{\bar{i}}+\xi_{\mathrm{r}1}-1}(1-\lambda_{\mathrm{r}})^{N-1-v_{\bar{i}}+\xi_{\mathrm{r}0}-1}}{B(\xi_{\mathrm{r}0}, \xi_{\mathrm{r}1})}\, \mathrm{d}\lambda_{\mathrm{r}} \\
=& \frac{B(v_{\bar{i}} + 1 + \xi_{\mathrm{r}1},\, N - 1 - v_{\bar{i}} + \xi_{\mathrm{r}0})}{B(\xi_{\mathrm{r}0}, \xi_{\mathrm{r}1})},
\end{aligned} \tag{4.22}$$

where $B(a, b)$ stands for the Beta function

$$\begin{aligned}
B(a,b) =& \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \\
=& \int_0^1 x^{a-1}(1-x)^{b-1}\, \mathrm{d}x,
\end{aligned} \tag{4.23}$$

and $v_{\bar{i}}$ is the number of rows in the current bicluster. The position of the current bicluster is accessed upon the fixed labels.

Similarly,

$$\begin{aligned}
&\int p(r_i = 0 \,|\, \lambda_{\mathrm{r}})\, p(\mathcal{R}_{\bar{i}} \,|\, \lambda_{\mathrm{r}})\, \mathrm{d}\lambda_{\mathrm{r}} \\
=&\int_0^1 (1-\lambda_{\mathrm{r}}) \cdot \lambda_{\mathrm{r}}^{v_{\bar{i}}} \cdot (1-\lambda_{\mathrm{r}})^{(N-1-v_{\bar{i}})} \cdot \frac{\lambda_{\mathrm{r}}^{\xi_{\mathrm{r}1}-1}(1-\lambda_{\mathrm{r}})^{\xi_{\mathrm{r}0}-1}}{B(\xi_{\mathrm{r}0}, \xi_{\mathrm{r}1})}\, \mathrm{d}\lambda_{\mathrm{r}} \\
=&\frac{B(v_{\bar{i}} + \xi_{\mathrm{r}1},\, N - v_{\bar{i}} + \xi_{\mathrm{r}0})}{B(\xi_{\mathrm{r}0}, \xi_{\mathrm{r}1})}.
\end{aligned} \tag{4.24}$$

Thus,

$$\begin{aligned}
\frac{\int p(r_i = 1 \,|\, \lambda_{\mathrm{r}})\, p(\mathcal{R}_{\bar{i}} \,|\, \lambda_{\mathrm{r}})\, \mathrm{d}\lambda_{\mathrm{r}}}{\int p(r_i = 0 \,|\, \lambda_{\mathrm{r}})\, p(\mathcal{R}_{\bar{i}} \,|\, \lambda_{\mathrm{r}})\, \mathrm{d}\lambda_{\mathrm{r}}} =& \frac{B(v_{\bar{i}} + 1 + \xi_{\mathrm{r}1},\, N - 1 - v_{\bar{i}} + \xi_{\mathrm{r}0})}{B(v_{\bar{i}} + \xi_{\mathrm{r}1},\, N - v_{\bar{i}} + \xi_{\mathrm{r}0})} \\
=& \frac{\Gamma(v_{\bar{i}} + 1 + \xi_{\mathrm{r}1})\Gamma(N - 1 - v_{\bar{i}} + \xi_{\mathrm{r}0})}{\Gamma(v_{\bar{i}} + \xi_{\mathrm{r}1})\Gamma(N - v_{\bar{i}} + \xi_{\mathrm{r}0})} \\
=& \frac{v_{\bar{i}} + \xi_{\mathrm{r}1}}{N - 1 - v_{\bar{i}} + \xi_{\mathrm{r}0}}.
\end{aligned} \tag{4.25}$$

Putting Equation 4.25 into Equation 4.21, we have

$$\gamma_i^r = \frac{p(\mathbf{D} \,|\, r_i = 1,\, \mathcal{R}_{\bar{i}},\, \mathcal{C}\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})}{p(\mathbf{D} \,|\, r_i = 0,\, \mathcal{R}_{\bar{i}},\, \mathcal{C}\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})} \cdot \frac{v_{\bar{i}} + \xi_{\mathrm{r}1}}{N - 1 - v_{\bar{i}} + \xi_{\mathrm{r}0}} \quad i = 1 \ldots N. \tag{4.26}$$

From exactly the same reasoning, for Equation 4.20, we have

$$\gamma_j^c = \frac{p(\mathbf{D} \,|\, \mathcal{R},\, c_j = 1,\, \mathcal{C}_{\bar{j}}\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})}{p(\mathbf{D} \,|\, \mathcal{R},\, c_j = 0,\, \mathcal{C}_{\bar{j}}\, \mathcal{M}^{\mathrm{bcl}},\, \mathcal{M}^{\mathrm{bgd}})} \cdot \frac{w_{\bar{j}} + \xi_{\mathrm{r}1}}{M - 1 - w_{\bar{j}} + \xi_{\mathrm{r}0}} \quad j = 1 \ldots M, \tag{4.27}$$

where $w_{\bar{j}}$ is the number of columns in the current bicluster.

4.4 The final Gibbs sampling scheme

To summarize, the Gibbs sampling scheme for the biclustering problem is as follows.

1. Initialization: Randomly assign the row labels and the column to either 1 or 0. Initialize the parameters in $\mathcal{M}^{\mathrm{bcl}}$, and $\mathcal{M}^{\mathrm{bgd}}$.
2. Sample the parameters of $\mathcal{M}^{\mathrm{bcl}}$ and $\mathcal{M}^{\mathrm{bgd}}$ according to their conditional distributions (see Chapter 5 and Chapter 6 for detail).
3. Fix the labels of the columns. For each row i, $(i = 1, \ldots, N)$, fix the labels for all the other rows, and
   (a) Calculate the Bernoulli distribution for the row as described in Equation 4.26.

(b) Draw a label for r_i from the distribution.

4. Fix the labels of the rows. For every column j, $(j = 1, 2, ..., M)$, fix the labels of all the other columns, and
   (a) Calculate the Bernoulli distribution for the column as described in Equation 4.27.
   (b) Draw a label for c_j from the distribution.
5. Go to Step 2, and iterate for a predefined number of iterations

4.5 From samples to the final pattern

To reduce the memory complexity of the algorithm, we only collect the samples of the labels to determine the final position of the bicluster. The parameters for the model and the background—if necessary—will be obtained by evaluating the sample statistics of these parameters according to the determined position of the bicluster. According to Equation 4.7, the MAP estimate of a label is.

$$\begin{aligned}
\mathrm{E}_{p(r_i \mid \mathbf{D})}[r_i] &= \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}[r_i \mid \mathcal{R}_{\bar{i}}^{(t)}, \mathcal{C}^{(t)}, (\mathcal{M}^{\mathrm{bcl}})^{(t)}, (\mathcal{M}^{\mathrm{bgd}})^{(t)}, \mathbf{D}] \quad r_i \in \mathcal{R}, \\
\mathrm{E}_{p}(c_j \mid \mathbf{D})[c_j] &= \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}[c_j \mid \mathcal{R}^{(t)}, \mathcal{C}_{\bar{j}}^{(t)}, (\mathcal{M}^{\mathrm{bcl}})^{(t)}, (\mathcal{M}^{\mathrm{bgd}})^{(t)}, \mathbf{D}] \quad c_j \in \mathcal{C},
\end{aligned} \tag{4.28}$$

Similarly, according to Equation 4.8, the marginal distribution of a label is

$$\begin{aligned}
\mathrm{E}[p(r_i \mid \mathbf{D}] &= \frac{1}{T}\sum_{t=1}^{T} p(r_i \mid \mathcal{R}_{\bar{i}}^{(t)}, \mathcal{C}^{(t)}, (\mathcal{M}^{\mathrm{bcl}})^{(t)}, (\mathcal{M}^{\mathrm{bgd}})^{(t)}, \mathbf{D}) \quad r_i \in \mathcal{R}, \\
\mathrm{E}[p(c_j \mid \mathbf{D}) &= \frac{1}{T}\sum_{t=1}^{T} p(c_j, \mid \mathcal{R}^{(t)}, \mathcal{C}_{\bar{j}}^{(t)}, (\mathcal{M}^{\mathrm{bcl}})^{(t)}, (\mathcal{M}^{\mathrm{bgd}})^{(t)}, \mathbf{D}) \quad c_j \in \mathcal{C},
\end{aligned} \tag{4.29}$$

where t indexes the iterations and T is the total number of iterations in the sampling procedure. To determine the final position of the bicluster, we can either put a threshold on the MAP estimate of the labels, and select only the rows and columns for whom the MAP estimate are above that threshold for the bicluster. Or, another more meaningful way is to put a threshold on the, say 95% quantile, of the estimated marginal distribution of the labels. In the examples that we give in Chapter 5 and Chapter 6, we use the latter method as the criterion for the determination of the position of the bicluster.

In our implementation, we use 500 iteration as default for performing the Gibbs sampling procedure (including both the burn-in and the sampling procedure). The autocorrelation (see Equation 4.10) between the samples is monitored for the determination of convergence, and additional iterations are added if necessary.

4.6 Multiple biclusters

The probabilistic model that we discussed above 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 (for both the biclustering of genes and the biclustering of experiments) to mask the experiments selected for the found biclusters and rerun the algorithm on the rest of the data. By masking, we mean that the labels of the experiments in all the found biclusters are set permanently to 0, and are not included in the next round of the Gibbs sampling procedure. In this way, experiments retrieved for previous biclusters will not further be selected as candidate genes for any future bicluster, while the data under these experiments will be included for the evaluation of the background model for the next bicluster. Note that this choice does allow the genes 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 4.6.1 for the decision).

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 a gene 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.

4.6.1 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 3 and Step 4 of the algorithm (see Section 4.4). 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.