KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

GIBBS SAMPLING ON BAYESIAN MODELS FOR BICLUSTERING MICROARRAY DATA

Promotoren:
Prof. dr. ir. B. De Moor
Prof. dr. ir. Y. Moreau

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Qizheng SHENG

November 2005

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

GIBBS SAMPLING ON BAYESIAN MODELS FOR BICLUSTERING MICROARRAY DATA

Jury:
Prof. dr. ir. G. De Roeck, voorzitter
Prof. dr. ir. B. De Moor, promotor
Prof. dr. ir. Y. Moreau, co-promotor
Prof. dr. ir. J. Vandewalle
Prof. dr. ir. H. Blockeel
Prof. dr. ir. J.A.K. Suykens
Prof. dr. ir. K. Marchal
Dr. J. Dopazo (CIPF, Spain)

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Qizheng SHENG

U.D.C. 519.24 November 2005

D/2005/7515/90

ISBN 90-5682-656-5

Acknowledgment

First of all, I would like to thank my promoter Prof. Bart De Moor for introducing me to the dazzling field of bioinformatics when I came to Belgium five years ago to pursue a higher degree in engineering. I also thank him for his belief in me and his support during the difficult times of my PhD study.

The thesis would have been less complete without the help of Prof. Yves Moreau, who is my co-promoter and daily advisor. I am grateful to him for the ideas that he shared with me, for the research directions that he advised me, and for the many helpful discussions during the development of the methodology presented in this thesis.

Prof. Kathleen Marchal has also been a great support for my PhD research. I thank her especially for all the insightful discussions on the applications of the methodology in systems biology. It is a great pleasure to have her in the jury of my thesis.

I would like to acknowledge Karen Lemmens and Peter Van Loo, two of my dear colleagues in ESAT-SCD-BIOI, for providing biological insights into the validation of the results, and for their useful discussions to improve the methodology.

Dr. Gert Thijs and Dr. Geert Fannes are two great ex-colleagues to whom I am grateful for their knowledge in mathematics and Bayesian statistics that they shared with me, as well as their guidance and lots of helpful advices during my PhD study.

In addition, I would like to thank all the other people who have worked with me in ESAT-SCD-BIOI for the nice working environment that they have created.

I also want to express my gratefulness to the two assessors of this thesis—Prof. Joos Vandewalle and Prof. Hendrik Blockeel—for their valuable feedback on polishing the thesis. In addition, it is an honor to have Prof. Guido De Roeck, Prof. Johan Suykens, and Dr. Joaquin Dopazo in the jury of my thesis. I appreciate Dr. Dopazo's taking the troubles to fly from Spain to fulfill this task.

To come and live in a culture totally different from the Chinese culture has been a challenge for me. My PhD study would not have been carried out smoothly without the help of the many friends I made in Leuven who have created a cozy living environment for me. I would like to thank them all for putting more open-mindedness to different cultures into my character.

Finally, I especially want to thank my parents, who have always believed in me and supported my decisions in every way they can, and whose unconditional love has been the backbone for me to finish this long journey of PhD study.

Abstract

Biclustering of microarray data is gaining increasing attention from researchers both in systems biology and in systems biomedicine. For systems biology, biclustering algorithms have the advantage of discovering genes that are coexpressed in a subset of (instead of all) the measured conditions, compared with conventional clustering methods. Since the emergence of web-based repositories of microarray data such as ArrayExpress and GEO, analysis based on microarray compendia where gene expression levels are measured under a large number of heterogeneous conditions has become more and more popular. Biclustering suits the needs for this type of analysis, especially for discovery of transcriptional modules, which provide essential clues for revealing genetic networks. For systems biomedicine, biclustering concerns the other orientation of microarray data, which is to cluster experiments (e.g., tumor samples) based on a subset of genes for each of which the experiments show consistent expression levels. The pattern of the target bicluster provides a gene expression fingerprint for the classification of the experiments. Therefore, the bicluster can help to reveal genes that are important for the pathology.

In this thesis, we propose a biclustering strategy based on Bayesian modeling of microarray data and Gibbs sampling for the parameterization of the model. Bayesian models give our method the advantage of incorporating prior knowledge so that the resulting bicluster can be directed towards answering the specific questions of the biologist, such as "what are the genes that are involved in this particular function, and what are the working conditions of the function?" In addition, Bayesian models also provide the base for the integration of information extracted from other data sources. Research in bioinformatics has seen growing awareness that data from different sources should not be studied in isolation. This awareness is calling out the need for tools that allow such integration to take place.

Because of the high complexity of the biological process underlying a microarray data set, optimization methods for the clustering problems of microarray data often run into the problem of local maximum solutions. The corresponding clusters are often not interesting for the biologists, or often give an incomplete answer. Gibbs sampling is known for its ability to enhance

the probability to discover the global maximum solutions. We consider this a favorable property for the study of microarray data. We provide several case studies to illustrate the efficiency of our strategy.

Contents

Notations

Mathematical notations

X	scalar random variable
x	realization of random variable X
$\mathbf{X}_m$	set of random variables with set-length equals m
$\mathbf{x}$	realization for the set of random variables $\mathbf{X}_m$
$\mathcal{X}$	set
$p(\cdot)$	density function
$P(\cdot)$	probability distribution
$\mathrm{E}_{p(X)}[X]$	expectation of random variable X based on the probability distribution $p(X)$
$\mathrm{E}[p(X)]$	expectation of the distribution $p(X)$ itself

Fixed symbols

bcl	The subscript denoting that the associated variable is applied to the bicluster
bgd	The subscript denoting that the associated variable is applied to the background
$\mathbf{C}_m$	$\mathbf{C}_m = \{C_1, C_2, \ldots, C_m\}$, set of structural variables for the Bayesian hierarchical model on the biclustering problem.
C_j	A binary variable indicating whether the j^{th} column in the matrix belongs to the bicluster
$\mathbf{c}$	indices of structural variables in $\mathbf{C}_m$ whose values equal 1
$\bar{\mathbf{c}}$	indices of structural variables in $\mathbf{C}_m$ whose values equal 0
$\mathbf{e}$	(When biclustering genes) indices of columns in the data matrix that are assigned to the bicluster
$\bar{\mathbf{e}}$	(When biclustering genes) indices of columns in the data matrix that are assigned to the bicluster
$\bar{\mathbf{c}}$	indices of columns in the data matrix that are assigned to

	the background
$\mathcal{D}$	Microarray data matrix
$\mathcal{D}_R$	Missing data of the biclustering problem—realizations of R
$h(\mathcal{D})$	Counting function
n	Number of rows in a microarray data matrix $\mathcal{D}$
m	Number of columns in a microarray data matrix $\mathcal{D}$
q	Number of conditions in a microarray data set
R	Random variable that indicates whether a row in the matrix belongs to the bicluster or not
$\mathbf{r}$	Indices of rows in the data matrix that are assigned to the bicluster
$\bar{\mathbf{r}}$	Indices of rows in the data matrix that are assigned to the background
s^{α}	User input for the biclustering problem of experiments—scaling factor for adjusting $\boldsymbol{\alpha}$
s^{β}	User input for the biclustering problem of experiments—scaling factor for adjusting B
s^2	Parameter (scale) for the inverse-χ^2 distribution describing σ
$\mathbf{X}_m$	$\mathbf{X}_m = \{X_1, X_2, \ldots, X_m\}$, random variables to which microarray data is mapped. Each X_j is a random variable representing the gene expression level under experiment j.
$\mathbf{Y}_q$	$\mathbf{X}_m = \{Y_1, Y_2, \ldots, Y_q\}$, random variables corresponding to the experimental conditions of microarray data. Each Y_k is random variable representing the gene expression level under condition k.
$\boldsymbol{\alpha}$	Parameter vector for the Dirichlet distribution describing Ψ
B	Parameter matrix for the Dirichlet distributions describing $\boldsymbol{\Phi}$
γ_j^{c}	Odds between the posterior probability that a column belongs to the bicluster and the posterior probability that it does not
γ_i^{r}	Odds between the posterior probability that a row belongs to the bicluster and the posterior probability that it does not
ι	Autocorrelation time
Λ^{c}	Parameter for the Bernoulli distributions of $\mathbf{C}_m$
Λ^{r}	Parameter for the Bernoulli distribution of R
$\boldsymbol{\mu}$	Parameters (means) for the normal distribution describing the microarray data in the problem of biclustering genes
ν	Parameter (degree of freedom) for the inverse-χ^2 distribution describing σ
Ψ	Parameter vector for the multinomial distribution describing the background data in the problem of biclustering experiments, (i.e., model of $\mathbf{X}^{\mathrm{bgd}}$)

$\mathbf{\Phi}$	Parameter matrix for the multinomial distributions describing the background data in the problem of biclustering experiments, (i.e., model of $\mathbf{X}^{\text{bcl}}$)
φ	Parameters (means) for the normal distribution describing μ
σ	Parameters (variance) for the normal distribution describing the microarray data in the problem of biclustering genes
τ^2	Parameters (variance) for the normal distribution describing μ
$\mathbf{\Theta}$	Parameters for the distribution that models $\mathbf{X}_m$
ξ	Parameters for the distribution that models $\mathbf{\Theta}$
ζ^{c}	Hyperparameter for the prior Beta distribution of Λ^{c}
ζ^{r}	Hyperparameter for the prior Beta distribution of Λ^{r}

Acronyms

ALL	acute lymphoblastic leukemia
AML	acute myelogenous leukemia
BIC	Bayesian information score
cDNA	complementary DNA
CPD	conditional probability distribution
EST	expressed sequence tag
EM	expectation–maximization
DAG	directed acyclic graph
GO	gene ontology
HMM	hidden Markov model
IM	ideal mismatch
IQR	interquartile range
MLL	mixed-lineage leukemia
MM	mismatch (probe)
mRNA	messenger RNA
ORF	open reading frame
PCA	principle component analysis
PM	perfect-match (probe)
PME	posterior mean estimate
PRM	probabilistic relational model
RMA	robust multichip average
SB	(biweight) specific background
SOM	self-organizing maps
VSN	variance stabilizing normalization

Publication List

International Journal

Qizheng Sheng, Yves Moreau, and Bart De Moor, Biclustering microarray data by Gibbs sampling, 2003, *Bioinformatics*, **19**, ii196–ii205

Internal Report

Qizheng Sheng, Karen Lemmens, Kathleen Marchal, Bart De Moor, and Yves Moreau, Query-driven biclustering of microarray data by Gibbs sampling, Internal report 05-33, *Department of Electrical Engineering, ESAT-SCD-SISTA, Katholieke Universiteit Leuven* (Leuven, Belgium), 2005.

Book Chapter

Qizheng Sheng, Yves Moreau, Frank De Smet, Kathleen Marchal, and Bart De Moor, Advances in cluster analysis of microarray data, Chapter 10 of *Data Analysis and Visualization in Genomics and Proteomics*, Francisco Azuaje and Joaquin Dopazo (eds.), 2005, John Wiley & Sons Ltd., 153-173.

International Conference

Qizheng Sheng, Gert Thijs, Yves Moreau and Bart De Moor, Applications of Gibbs sampling strategy in bioinformatics, *Workshop on mathematical programming in data mining and machine Learning*, June 1–4, 2005, Hamilton, ON, Canada, submitted for joint publication in *Optimization Methods and Software*.

Chapter 1

Introduction

In this opening chapter of the thesis, we put the main idea of this thesis in a nutshell. We start with a brief introduction of the biological background of the study of bioinformatics, which is followed by a brief explanation of the concept of microarray technology, especially with respect to its role in bioinformatics. We then give a problem statement of what biclustering of microarray data is and why it is an important subject in bioinformatics. After that, we propose a biclustering strategy based on Bayesian modeling and Gibbs sampling for parameter estimation. We introduce the concepts of Bayesian modeling and Gibbs sampling, and provide an explanation of the main advantages of our methodology. Finally, we finish this chapter by an overview of the organization of the thesis.

1.1 Biological background

The study of molecular biology is based on the following central dogma, which was first formulated by Crick (1958) [26]. DNA is known as the carrier of genetic information that is needed to conduct the synthesis of proteins—the workhorses in a living cell. The DNA molecule is composed of two complementary strands, which are made up of four basic units—the nucleotides adenine (A), cytosine (C), guanine (G), and thymine (T), see Figure 1.1. A nucleotide on one strand of the DNA is paired up with the complementary nucleotide at the same position on the other strand by a strict rule of basic pairing, i.e., (guanine (G) can only be paired with cytosine (C), while adenine (A) can only be paired with thymine (T), see Figure 1.1). Genes are the working subunits of DNA molecules that carry such essential information for the construction of proteins and other functional products.

The first step of a protein synthesis procedure is the transcription of its corre-

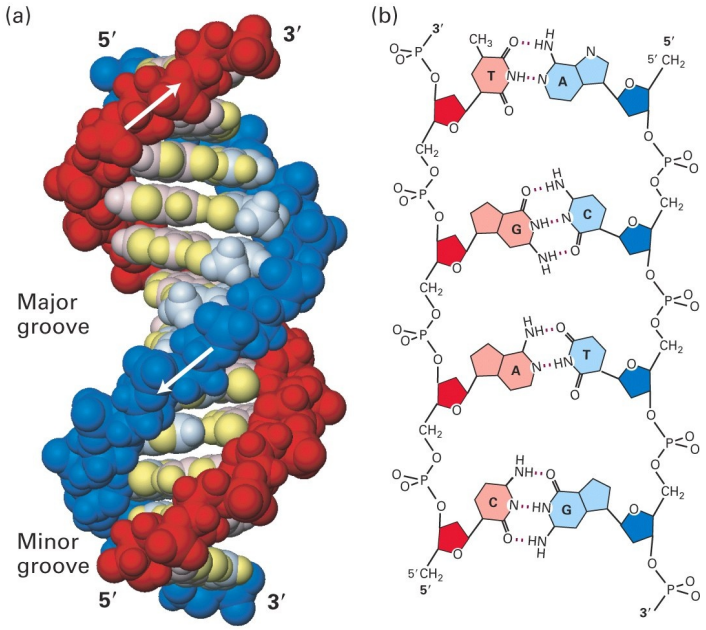


Figure 1.1: (A): 3D illustration of the structure of the DNA molecule. (B) Rule of base pairing for the four nucleotides—adenine (A), cytosine (C), guanine (G), and thymine (T), which are basic components of DNA molecules. Both of the figures illustrate the double helix structure of DNA molecule. At each complementary position on the double helix, the nucleotides are paired according to a strict rule so that guanine (G) can only be paired with cytosine (C), and adenine (A) can only be paired with thymine (T). The figures are obtained from Scott *et al*. (2003).

sponding gene, to a messenger RNA (mRNA), see Procedure 1 in Figure 1.2. This step highly resembles the duplication of DNA molecules. With the help of RNA polymerases, the two strands of DNA are separated at the location of the target gene, and each strand is used as a template from which mRNA molecules are copied (i.e., transcribed). This process is also carried out according to the rule of base pairing. The only difference is that uracil (U) is paired with adenine (and vice versa), because there is no thymine in RNA.

The second step is the translation of the mRNA to the protein, see Procedure 3 in Figure 1.2. This step takes place with the help of ribosomes so that the mRNA is scanned three nucleotides (called a codon) at a time. Each possible combination of a codon (in total 64 possibilities) corresponds to one of the 20 amino acids. (Note that the redundancy of this coding system provides stability to protein synthesis against possible mutations.) In this way, a peptide chain is assembled by the ribosome. The peptide chain is later folded into the resulting protein.

Therefore the detailed residue-by-residue transfer of information is carried out from DNA to RNA to protein. However, this standard pathway of information flow was found to be an oversimplification, and in 1970, the central dogma of molecular biology is modified accordingly by Crick (1970) [27]. The modified information flow is presented in Figure 1.3.

The above is only one part of the story that concerns the guidance of genes in the synthesis of proteins. The other part, however, is related to the regulative roles of proteins in the transcriptions of genes. A transcription process for a gene is only able to start when all the needed transcription factors (which are proteins themselves) bind to the promoter region of the gene (which usually locates upstream, i.e., "in front", of a gene). Consequently, an RNA polymerase binds to the transcription factors and together forms a complex that opens the DNA double helix so that the transcription starts. (A good tutorial book for the beginners of biology is Scott *et al*. (2003) [85].)

The subjects of biological research range from genomics to proteomics and beyond. Looking at the level of genes (i.e., in genomics), biologists are most interested in the functions of the genes and their (regulatory) relation with each other. In this sense, the transcriptional behavior of the genes may provide a clue. Equipped with the newly developed microarray technology, it is possible now to simultaneously monitor the transcriptional behavior of a whole genome, which gives rise to the study of the transcriptome*, which is the main aspect of this thesis. Because proteins are the executors of the cellular functions that genes instruct, proteomics is also an active field of study aiming to associate proteins with different cellular functions. Of course, a cell cannot function without processing metabolites. Metabolomics is an area of study that considers the interactions and dynamics of all the metabolites in a cell.

*The transcriptome refers to the whole set of mRNAs in a cell under the studied circumstance.

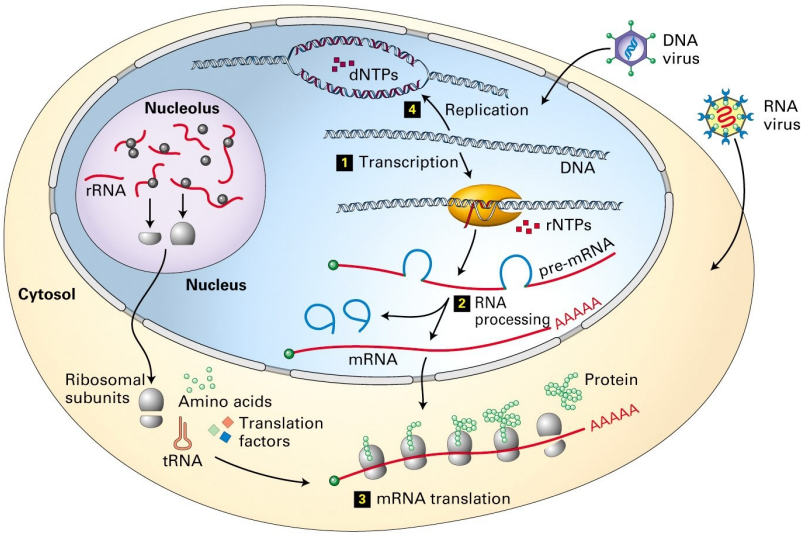


Figure 1.2: Biological processes in a eukaryotic cell. Transcription (Process 1) is the process during which mRNA molecules are made by using DNA molecules as a template. Transcription takes place in the nucleus. Translation (Process 3) refers to the production of proteins from mRNA molecules. This process takes place in the cytosol, and is assisted by both ribosomes and tRNAs. Both transcription and translation are the essential processes that execute the standard sequential information flow from DNA to protein. Other processes depicted in this figure include the replication of DNA (Process 4), and the processing mRNA (Process 2). For eukaryotic cells, an mRNA molecule is often spliced after the transcription takes place, and poly(A) tail is frequently added in the nucleus, and is then transported to the cytosol where the translation occurs. The figure is obtained from Scott *et al*. (2003).

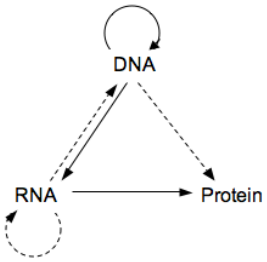


Figure 1.3: The picture depicts the conclusion of Crick (1970), which restated the central dogma of molecular biology. The residue-by-residue transfer of sequential information is represented by the arrows, where a solid arrow represent probable transfers and the dashed arrows represent possible transfers. While the figure confirms the standard information flow of "DNA makes RNA, RNA makes proteins" as well as the duplication of DNAs, it also summarizes other observed exceptions to the standard information flow (denoted by the dashed arrows).

1.2 Technological background

During the past few years, microarray technology [83] has emerged as an effective technique to measure the expression levels of thousands of genes in a single experiment.[†] Nowadays, a microarray chip take a snapshot of the gene expression levels of the whole genome while being no larger than a couple of square centimeters, see Figure 1.4 for an illustration. Putting together data obtained by from microarray experiments under different experimental conditions (which can be different tissues, time points, or environmental conditions), expression profiles are obtained for the genes measured on the microarray chips. Microarray data is often put in a matrix whose rows represent the genes and whose columns represent the experimental conditions, see Figure 1.5. Consequently, each row in a microarray data matrix represents the expression profile of the corresponding gene.

This technology has been become a major attraction for biologists ranging from those interested in gene expressions in yeast [63] to those that are involved in medical research [45], who hope to extract essential functional information about the genes from the expression profiles measured by the technology. However, without the help of powerful computational and statistical techniques, analyzing data in such immense amount and of such complexity is impractical. To begin with, gene expression profiles measured by microarray technology are often complicated by systematic noise introduced by the pitfalls

[†]When a gene is activated and its corresponding mRNA is produced, the gene is said to be expressed in the specific circumstance under discussion. The expression level of a gene refers to the level of abundance of its correspondent mRNA in the cell.

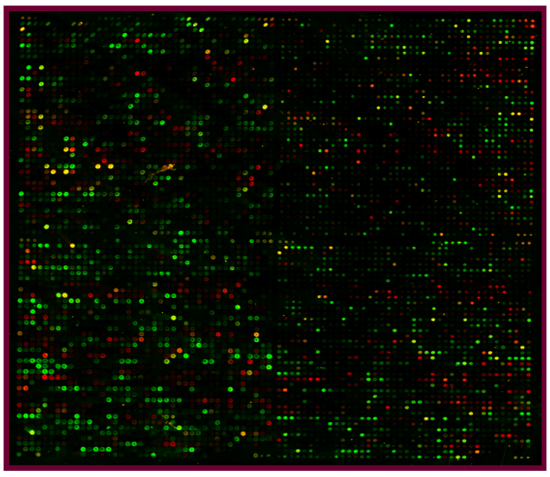


Figure 1.4: A resulting image from a microarray chip (enlarged), where each dot on the chip represents a gene. The color of a dot indicates the level of abundance of the corresponding mRNA of the gene in the cell. The image is obtained by a two-color-channel cDNA microarray technology (see Chapter 2 for an explanation about the technology). Typically, if red indicates that the expression level of the gene is higher under the test condition than under the control condition (i.e., the gene is overexpressed under the test condition), green means the gene is underexpressed in the test condition. Yellow indicates that the gene is expressed under both the test condition and the control condition, and that the levels of expression are similar under the two conditions. On the other hand, if the corresponding color of a gene is black, it means that the gene is not expressed in either of the conditions.

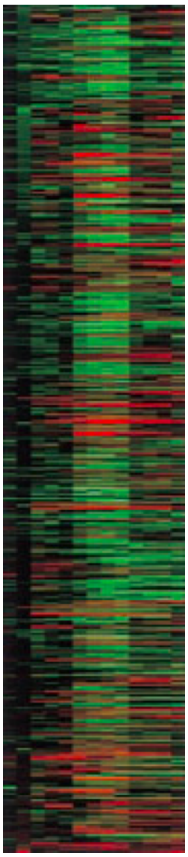


Figure 1.5: Data collected from several microarray experiments are put together in a matrix, where the rows represent the genes and the columns represent the experiments (which are performed on several chips). The values of expression of the genes are represented here by color scales. These values are often derived from the log ratios of the measured gene expression values under the test condition and the control condition (see Chapter 2 for more discussion). Consequently, each row of the matrix represents the expression profile of a gene. Also observe the asymmetry in the dimension of the data—while the number of genes can reach several tens of thousands, the number of conditions is usually up to a few hundred. The figure is obtained from Eisen *et al*. (1998).

of the technology and during the measurement procedure. Therefore, efficient mathematical modeling is needed to correct the systematic noise, a procedure that is often referred to as the normalization. This thesis, however, mainly focuses on a data mining technology that helps biologists to extract essential information from microarray data after normalization is performed. Yet, microarray data have several characteristic features that cannot be corrected during the normalization procedure. First, microarray data contain a huge amount of noise introduced by the underlying biological process as well as the measuring procedure. Secondly, microarray data often form a data matrix with asymmetric dimension. While the number of genes can easily reach tens of thousands, the number of experimental conditions is often no more than a few hundred, see Figure 1.5.

1.3 Biclustering problems for microarray data

A core problem of modern molecular biology research is to unveil the function of the genes. Throughout the years, the goal has evolved from understanding the individual role that a gene plays in the cell by studying the genes in isolation, to the unveiling of the concerted genetic program that is involved in a biological process. By measuring the expression levels of the whole genome under different conditions, microarrays record the activities of the genes in interaction so that information about different functional relationships between the genes is reserved.

For medical applications where the conditions of a microarray study often refer to the different tumor samples from which the mRNA samples are taken, it is reasonable to believe that tumor samples of the same pathological type should have similar expression level for each of those genes that play a responsible role for the pathology. Therefore, we look for algorithms to cluster tumor samples based on their gene expression levels for a subset of genes; and in the meantime, the algorithm should be able to select those genes where the tumor samples of the same cluster show similar expression levels, see Figure 1.6 (B) for an illustration.

For molecular biology, one of the basic assumptions in the functional discovery of genes using microarray data is that coexpressed genes (i.e., genes who share similar expression profiles) often have similar function. This assumption gives rise to the applications of various clustering algorithms on microarray data, aiming to find clusters of genes where the selected genes are coexpressed under *all* the experimental conditions. In the early days of the applications of microarray technology [32, 93], experiments were often conducted under a limited number of homogeneous experimental conditions measured at different time points. In this case, clustering algorithms are a sensible choice because the above assumption is often valid.

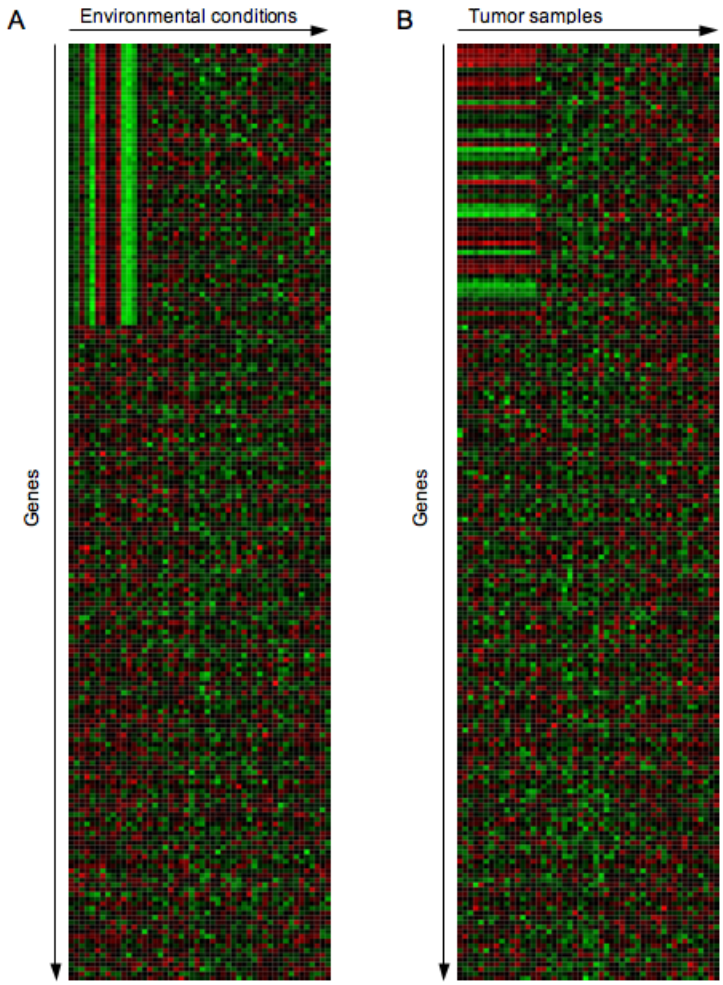


Figure 1.6: (A) Biclustering genes: the problem is to find a set of genes that share similar expression under a subset of conditions. (B) Biclustering experiments: the problem is to find a set of experiments (in this case, tumor samples) that have the same expression levels for each of the selected genes. The resulting biclusters are displayed at the top left corner of the figures. Note that the two problems should be treated differently because of the asymmetry in the dimensionality of microarray data sets—(A) large sample size, but small dimension of the vector space, (B) small sample size, but large dimension of the vector space.

With the maturation of the microarray technologies and of the normalization techniques, the reproducibility of microarray data is improved and comparison between microarray data produced by different labs become more feasible. In addition, with the establishment of a standard for recording and reporting microarray data—minimum information about a microarray experiment (MIAME) [18], it is, nowadays, plausible to retrieve data from publicly available repositories of microarray experiments, such as ArrayExpress [75] and GEO [12], and perform the analysis for a combined microarray data set whose experiments form a heterogeneous compendium. In this case, the assumption for applying conventional clustering techniques no longer holds. Because in this case, genes that share similar functions only exhibit coexpression under their working conditions. Therefore, instead, gene expression profiles should be clustered only under a *subset* of conditions, see Figure 1.6 (A) for an illustration. Biclustering algorithms are introduced to cluster genes and in the mean time to identify the conditions under which genes in the same cluster exhibit similar expression profiles.

However, because of the asymmetry in the dimension of microarray data—much larger number of genes than the number of conditions—the two biclustering problems that we introduced above should be treated individually, which will be explained in more details in the following section. To distinguish them from each other, we refer to them respectively as the biclustering of experiments and the biclustering of genes.

1.4 Bayesian models for microarray data

Probabilistic models have become a popular choice for modeling microarray data because they handle the high level of noise of microarrays in a principled way. Methods based on probabilistic models often treat microarray data as a mixture model of different probability distributions, where each cluster is modeled by a component of the mixture (i.e., one probability distribution). The probability distributions in the mixture model are often in the form of multivariate distributions. For clustering genes, each experiment (i.e., each column of the microarray matrix) is represented by a variate, and the genes are considered as samples from which the multivariate probability distributions are evaluated. However, for the problem of clustering experiments, the variates in question refer to the genes (i.e., rows) in microarray data set, while the experiments are regarded as the samples (see Figure 1.6). Furthermore, in the problem of biclustering, the goal is not only to associate the samples to the different components in the mixture, but also to pick out the relevant variates for each of the probability distributions in the mixture. In the case of biclustering genes (see Figure 1.6 (A)) the number of samples (i.e., genes) available for evaluating the probability distributions are relatively large comparing with the number of variates (i.e. experiments) under consideration. However, in

the case of biclustering experiments, the problem is the other way round—the number of variates (i.e., genes) overwhelms the number of samples (i.e., experiments), see Figure 1.6 (B). This is what we refer to as the asymmetry in the biclustering problems of microarray data.

The likelihood of a mixture model for a microarray data usually contains many modes (i.e., ways of constructing the components), because of the complexity of the underlying biological process. In clustering, these modes correspond to the different clustering results that can be derived from the data. The largest mode (which are easiest to identify) often results in large bicluster that are not the most interesting to the biologist because they correspond to well-known generic biological functions—where few novel findings are to be expected. This lack of sharpness of clustering algorithms has kept clustering algorithms into a vague exploratory role; because for biologists, one of the main questions is always "what are the genes that are related to a *particular* function (or in a *specific* pathway) of interest to me?" Note that patterns discovered for this purpose in microarray data are referred to mathematically as a bicluster, and biologically it is often referred to as a transcriptional module.

In addition, to study the concerted gene activities in a cell and the different relationships between them calls out for the need to integrate different data sources besides microarray data (e.g., DNA sequence information, protein structural information).

Bayesian probabilistic models have shown promise in both answering specific questions of biologists and providing a base for the integration of information from different data sources. Bayesian probability models differ from traditional probabilistic models in their inference procedure, which is summarized by Bayes' rule. Bayesian model can be interpreted as follows,

$$\text{Posterior probability} = \frac{\text{Prior probability} \times \text{Likelihood}}{\text{Evidence}},$$

see Chapter 4 for a discussion. This inference procedure of Bayesian model learning highly resembles that of the human learning process and formalized the practice of inductive reasoning. It allows the introduction of prior knowledge in which form soft queries can be imposed to direct the discovery. The introduction of the prior also provides a systematic base through which information from different sources can be integrated. By introducing a prior, methods based on Bayesian models zoom into the local area of interest of the likelihood landscape, and raise the corresponding area in the posterior according to the Bayes' rule.

1.5 Gibbs sampling for Bayesian models on microarray data

In Bayesian models for microarray data, though the mode that provides answer to the question of interest is raised in the posterior distribution, the other modes in the likelihood function of the models cannot be eliminated. These modes can still be identified by optimization methods, which aim at global maximum solutions in the posterior distribution. When this happens, the optimization method is said to find the local maxima of the posterior distribution.

Gibbs sampling [19] is known as one of the techniques enhanced the probability to find the mode that corresponds to the maximum probability in a posterior mixture model. Gibbs sampling is an empirical method to sample from a posterior distribution, when the analytical form of the posterior distribution is not trivial to get, and when the conditional distributions of all the concerned variates are available. It is a Markov chain Monte Carlo (MCMC) method. The Gibbs sampling procedure is carried out by sampling iteratively from the conditional distributions of each of the involved variates. Samples collected by Gibbs sampling are guaranteed to converge to the joint distribution by the Markov chain property. Then Monte Carlo integration is applied to these samples to evaluate the target distribution. In brief, the Gibbs sampling procedure produces samples that picture the posterior distribution as a whole, and consequently the mode (or an approximation thereof) of the posterior distribution that corresponds to the global maximum solution is decided by Monte Carlo integration of the samples.

1.6 Organization of the thesis

This thesis is organized as follows (also see Figure 1.7). In Chapter 2, we overview the various microarray technologies and the pitfalls that lie in these technologies. This is then followed by a review of the main quality control measures and normalization techniques that help to minimize the systematic noise in microarray data. We summarize the chapter by enumerating the characteristics of normalized microarray data. These characteristics require full awareness when designing data analysis tools for normalized microarray data.

Since the emergence of microarray technology, clustering techniques have been recognized as a useful tool for the analysis of microarray data. Standard clustering methods, such as hierarchical clustering, *K*-means, and self-organizing maps (SOM), were applied directly to microarray data and dominated the early papers for microarray data analysis [32, 93, 3, 108, 97, 101, 95]. With growing experience, it became clear that tailored clustering algorithms are required to improve the analysis. Throughout the years, numerous techniques have been

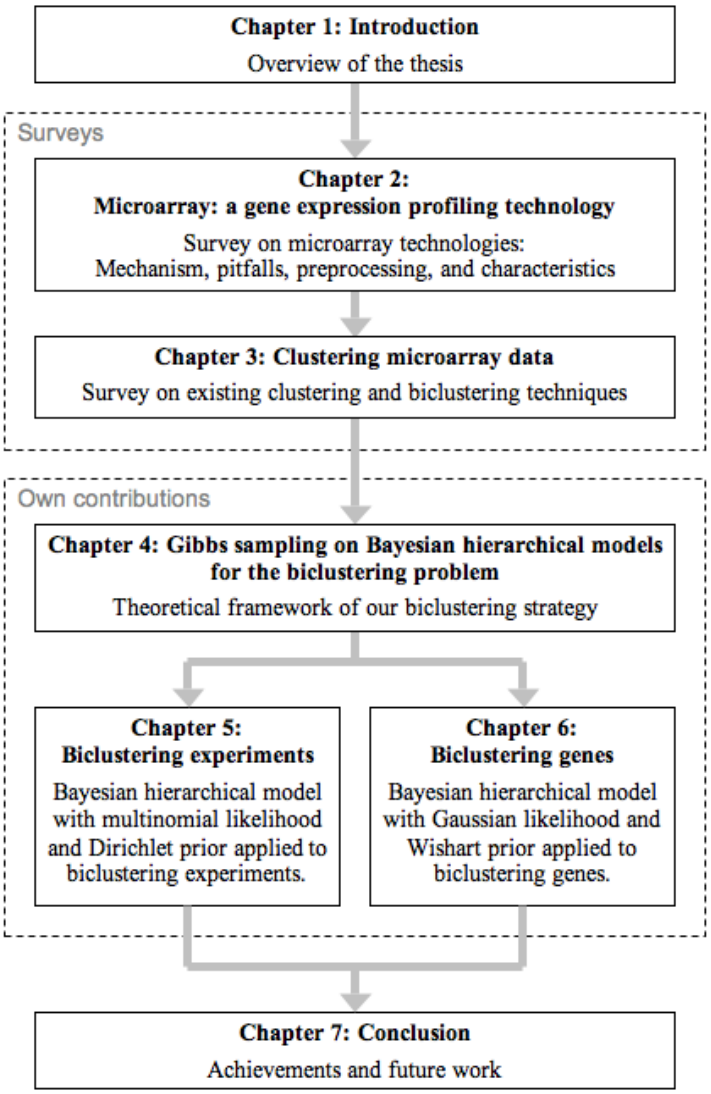


Figure 1.7: Organization of the thesis.

developed for clustering microarray data. Furthermore, this still remains an active field of research. In Chapter 3, we review several popular clustering techniques, and further introduce the need for biclustering algorithms. We also account for several existing biclustering algorithms. The chapter is concluded by a checklist for evaluating cluster quality.

Our biclustering strategy is fully explained in Chapter 4, where the concepts of Gibbs sampling and Bayesian models are first introduced separately, and then combined together to address the biclustering problem. This chapter focuses on the general framework of our methodology. The technical details for carrying out such an analysis are filled in the following two chapters.

Chapter 5 explains the application of our methodology to the problem of biclustering experiments. We discuss the application in two scenarios. The first one is the global pattern discovery in microarray data, which is suitable when no prior knowledge is available about the class of the experiments (e.g., tumor samples). In the second scenario, we consider biclusters of tumor samples whose shared genotype is fingerprinted by a weaker expression pattern that is overwhelmed by a dominant bicluster embedded in the data. We discuss the use of a set of seed tumor samples from which we extract information to construct prior knowledge for bicluster. We demonstrate the effectiveness of our algorithm on two data sets of leukemia patients.

In Chapter 6, we put the problem of biclustering genes in the context of gene regulatory module discovery. We first give more biological background for the purpose of such study. We then explain in detail how to transform information from the seed genes into the prior knowledge for the Bayesian model. We illustrate the usefulness of our algorithm in regulatory module discovery by applying the method on a combined data on *Saccharomyces cerevisiae*.

Finally, in Chapter 7, we conclude our work and propose some challenges for further research on this topic.

1.7 Achievements

Our main contribution can be summarized as follows.

Introduction of prior knowledge and integration of information from other data sources into biclustering

Bayesian models provide a systematic base for the introduction of prior knowledge and the integration of other data sources. We illustrate in Chapter 6 the usefulness of our method in cooperation with other methods to discover gene regulatory modules in the study of systems biology. Our biclustering results reveal highly coexpressed genes under a subset of biological conditions that are

highly correlated to the working conditions of the governing regulatory program. We also illustrate (in Chapter 5) the same methodology to incorporate information from a small number of patient samples to direct the discovery of bicluster toward the finding of gene expressional fingerprints of subtle traits.

Robust results

The choice of Gibbs sampling for the parameterization of the Bayesian models provides our method a high frequency to find the global maximum solution of the posterior probability mixture. We demonstrate such ability of our methodology in Chapter 5 and Chapter 6. In addition, we illustrate that the final biclusters discovered by our algorithm often only differ in a few genes or a few conditions.

Handling missing values in the data in a natural way

Because of the use of probabilistic models, missing values in the microarray values are handled in the most natural way by assuming that they are generated equally likely by the background component and by the bicluster component of the mixture model.

Allowing genes to belong to different biclusters

Another advantage of our strategy in contrast to conventional clustering algorithms is its ability to include one gene in different biclusters. This is also a desirable property based on the fact that a gene can have multiple functions.

Chapter 2

Microarray: a gene expression profiling technology

In this chapter, we provide a survey of popular microarray technologies applied to gene expression profiling. We start with an overview of different technologies that are used to manufacture microarrays. This overview not only explains the working mechanisms of microarrays, but also provides a better understanding of the pitfalls and noise present in microarray data. Then, we make a survey of various preprocessing methods that help to remove the systematic noise introduced during the manufacturing procedure. We conclude the chapter by reminding the readers about the characteristics of preprocessed microarray data, which should be taken into account when designing clustering algorithms for such data.

2.1 Introduction

A microarray is a chip (i.e. array) on the surface of which single-stranded DNAs (called probes) are bound in grid. When exposed to an RNA or cDNA sample obtained from a certain biological study, a microarray is able to capture a snapshot of the transcription levels (i.e., the mRNA levels) of tens of thousands of genes (nowadays, even a whole genome) under the experimental condition. By performing microarray experiments under different conditions, biologists can simultaneously monitor the behavior of the genes at the transcriptional level. The transcriptional behavior of a gene is thus described by its expression profile, which is made up of the expression levels of the gene under different experimental conditions.

Besides gene expression profiling, other applications of microarrays include

revealing genome-wide location of DNA-bound proteins [80], genome-wide analysis of DNA sequence copy number variation (the specific microarray technology is called comparative genomic hybridization) [76], and monitoring alternative splicing of pre-mRNA on a genome scale [114], among the others. However, we will limit our discussion to the application of microarray in gene expression analysis to keep within the scope of this thesis.

There are different technologies available for the making of microarray chips. However, the main mechanism for the measurement of mRNA abundance in the cells is the same for all the technologies. Microarrays used for gene expression profiling contain probes representing target genes for the study. mRNA samples in the studied biological process are extracted from the cells. They are then amplified, and sometimes reverse transcribed to complementary DNAs (cDNAs), which are less easy to degrade than the mRNA samples. The mRNA or the single-stranded cDNA is then labeled, usually by fluorescent dyes, and finally exposed to the chip. The measurement of the expression level of a gene relies on the binding (called array hybridization) of its corresponding (i.e., complementary) labeled mRNA or cDNA to the probe(s) representing the gene on the chip. Once the hybridization is finished, the unhybridized materials are washed away, and the chip is scanned so that the intensity of the fluorescence for each probe is read out, which should reflect the abundance of the corresponding mRNA in the cell.

However, from the building of the chips and the preparation of the mRNA samples, to the array hybridization and the final scanning procedure, every step involved in a microarray experiment introduces noise and artifacts to the readout data, which is faraway from the absolute measurement of mRNA abundance in a cell under the studied biological process. Thus, the raw data obtained from a microarray experiment needs to go under various preprocessing procedures that removes the systematic noise before any further analysis can be carried out.

2.2 Microarray technologies

The mainstream microarray technologies can be classified into two categories – spotted arrays, and *in situ* synthesized arrays. In spotted arrays, pre-synthesized DNA probes, which are typically oligonucleotides (i.e., short DNA sequences, usually of 50 to 80 bases in length) or cDNAs, are attached to glass or nylon slides. On the contrary, single stranded DNAs are synthesized directly on slide surface in *in situ* synthesized arrays. Because oligonucleotides are typically used as probes for *in situ* synthesized arrays, these arrays are often referred to as oligonucleotide arrays. Two dominant technologies in the market are cDNA arrays (a type of spotted arrays) and Affymetrix GeneChip® (a type of *in situ* synthesized arrays). Our following discussion will be based

on these two types of arrays.

2.2.1 cDNA microarrays

The probes on a cDNA microarray are cDNAs fragments genes. These cDNAs are typically of 100 to 5000 bases long. While the cDNAs were prepared (reverse transcribed from mRNAs) by individual labs in the early days of cDNA microarray technology, nowadays presynthesized cDNA clones are commercially available and are usually derived from reference banks of expressed sequence tags (ESTs), each of which is documented and, if possible, associated with a gene. When making cDNA microarrays, a robot fetches cDNA probes by its pins (fixed on its arm) from wells in a microtiter plate, and spots the probes onto a glass (or nylon) slide. Each spot on the microarray contains one cDNA probe representing one gene.

In a cDNA microarray experiment, mRNA samples (or their corresponding cDNAs) derived from two experiment conditions are hybridized to one microarray. One condition is used as the reference condition, and the set of mRNA/cDNA samples derived from this condition is called the reference sample. The other condition is the experimental condition of interest, which is referred to as the test condition. The set of mRNA/cDNA samples obtained in this experimental condition is called the test sample. The reference sample is labeled with the fluorescent dye Cy3, and the test sample is labeled with the fluorescent Cy5, or vice versa. After the hybridization, the chip is scanned at the wavelengths for Cy3 and Cy5. The ratio between the signal intensities of the two wavelengths measured at each spot on the array is reflects the expression level for the corresponding gene. Note that the application of two differentially labeled samples effectively removes the array-to-array variability in cDNA microarray technology [110]. Figure 2.1 provides an overview for the whole measuring procedure using cDNA microarray.

2.2.2 Affymetrix GeneChip

Affymetrix uses a combined technology of photolithography and combinatorial chemistry to synthesize nucleotides to the multiple growing chains of oligonucleotides on the surface of the chip. Figure 2.2 illustrate the manufacturing procedure of CeneChip.

Instead of using one probe for one target mRNA as is the case for cDNA microarrays, Affymetrix GeneChip uses a probe set to represent one transcript. A probe set usually contains 11 to 16 *probe pairs*, which identifies different regions of the target gene. The choices for the sequences of the probes are based on the predicted hybridization properties of the oligonucleotides, and are further filtered for specificity. Each probe pair consists of a perfect-match (PM) probe and a mismatch (MM) probe, where the PM probe is perfectly complementary to the target mRNA sequence, while the MM probe differs

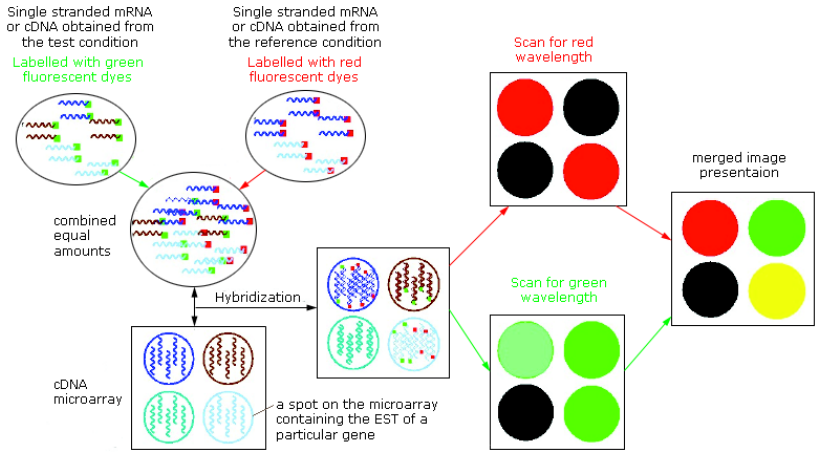


Figure 2.1: Measuring gene expression values by cDNA microarray. Note that the detected signal intensities of the fluorescence dyes (typically Cy3 and Cy5) are often converted by software into a red-and-green-dye presentation for the readout microarray data.

from the PM probe by only a single base in the center of the oligonucleotide. All the probes (i.e., oligonucleotides) are 25 bases long. Figure 2.3 illustrates the probe design strategy of Affymetrix GeneChip.

The PM/MM probe strategy originates from the consideration that it is unavoidable for mRNAs other than the target to bind to the PM probe. The MM probe is introduced with the intention to measure the non-specific binding of the corresponding PM probe. With this technology, only one mRNA is required, and the gene expression is measured as absolute value instead of ratios.

2.2.3 Comparison between spotted arrays and *in situ* synthesized arrays

A main drawback of spotted arrays is the big array-to-array variation. In addition, any deficiency in the synthesis and purification of the biomolecules to be spotted, or any contamination in the source plate will greatly affect the array quality. On the contrary, better precision in array manufacturing can be achieved for the *in situ* synthesized arrays because the technology relies merely on the source sequence information of oligonucleotides and synthesis chemistry, and thus provides a better base for between array, even between batch comparison.

Because of the involvement of photolithographic masks in the manufacturing procedure, Affymetrix CeneChip are expensive, while spotted arrays are usu-

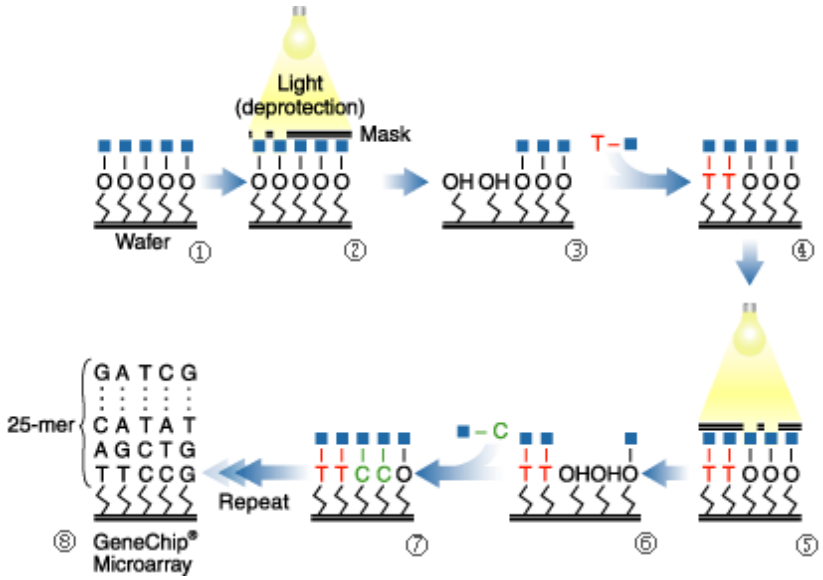


Figure 2.2: The manufacturing of Affymetrix CeneChip (picture source from Affymetrix http://www.affymetrix.com/technology/manufacturing/index.affx). (1) Linker molecules that can be activated by ultraviolet light are attached to the surface of a chip. (2)(5) A photo-protected mask with windows open for the desired oligonucleotides is placed over the surface of the chip, and ultraviolet light is shone over the mask. (3)(6) Linker molecules at the unprotected areas are activated. (4)(7) The surface is flushed with a solution containing a single nucleotide, and the nucleotide attaches to the oligonucleotides with activated ends. (8) The procedure is repeated to add all the four types of nucleotides: adenosine, thymine, cytosine and guanine, and is continued until the probes reach their full length, usually 25 bases long.

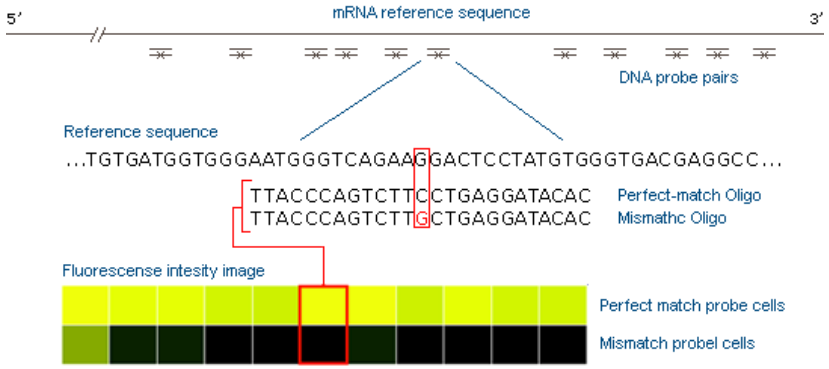


Figure 2.3: Probe design strategy of Affymetrix GeneChip. Affymetrix GeneChip uses a probe set to represent one transcript. A probe set usually contain 11 to 16 *probe pairs*, which identifies different regions of the target gene. Each probe pair consists of a perfect-match probe (PM) and a mismatch (MM) probe, where the PM probe is perfectly complementary to the target mRNA sequence, while the MM probe differs from the PM probe by only a single base in the center of the oligonucleotide. All the probes (i.e., oligonucleotides) are 25 bases long.

ally more affordable for small labs. Besides, spotted arrays are more flexible in customized design, for example, the users can decide the set of genes that are more relevant for the study and spot only these genes on the microarrays. However, an alternative *in situ* technology provided by NimbleGen® is said to provide more flexibility in array design as well as lower price.

2.3 Noise and artifacts in microarray data

In spite of their best efforts, all the existing microarray technologies cannot prevent noise and artifacts from being introduced into the data in every step of the biological and technical procedures involved.

The first thing needed to be pointed out for the discussion of this section is that the biological variations for different mRNAs make the comparison between expression levels measured for different genes on the same microarray worthless. Examples include the variation in the abilities of different mRNAs to be amplified (after being extracted from the cell culture) or to be hybridized to the chip. Thus, gene expression levels measured from a microarray is only meaningful when compared to those from (an)other microarray(s).

Besides biological variation, noise and artifacts are introduced because of the lack of effective controls in handling the mRNA samples – that is, from the

isolation of mRNAs to the array hybridizations.

While the idea of a microarray experiment is to measure the gene expression levels in a single cell under the studied condition, in reality, it is often difficult to separate the cells, and instead the mRNA levels in a population of cells are extracted for further measurement. The worst case scenario is that the extracted mRNA sample could come from different tissues. Another source of artifacts in the isolation of mRNA samples is caused by the subtle changes in the experimental condition during the procedure, where stress responses of the genes and mRNA degradation can easily happen if the cell cultures are handled without much caution [8].

When it comes to the array hybridization, and because the binding of mRNA molecules to the probes depends to a great extent on their three dimensional features, it happens that some mRNAs may bind to unintended probes. Furthermore, it is also possible for the free fluorescent dyes in the solution to land on the probes.

Speaking of fluorescent dyes, for cDNA microarrays (and other two-channel microarray technologies in general), the difference between the labeling efficiencies of Cy3 and Cy5 introduces another artifact, and can be effectively corrected by using two microarrays where the dyes for the test sample and the reference sample are swapped.

A third source of noise and artifacts lies in the pitfalls of the manufacturing of microarrays and the data readout technology. For the cDNA microarrays, a contaminated plate is one example, and a blocked or worn-out spotting pin is another. In addition, combined with an inappropriate scanning method, unevenly spotted probes on the chip will cause some areas of the chip to have a "brighter" background than the rest. For Affymetrix GeneChip, the variation among different probes in the same probe set should also be taken into account.

2.4 Preprocessing of microarray data

Because of the high level of noise in microarray data, it is essential to assess the quality of the data, and remove as much as possible the systematic noise that might obscure the biological variation, before any analysis of microarray data can be carried out. Therefore, preprocessing procedures are designed to check and remove as much as possible the systematic noise (such as array effect, plate effect and pin effect for cDNA microarrays, and probe effect for the Affymetrix CeneChip) in the raw microarray data, so that in the ideal world, the variation in the data is only explained by biology. The main assumption for most of the preprocessing measures to work is that the expression levels of most of the genes are not differentially expressed under different experimental condition. Therefore, looking from the population level, when we segment the obtained expression data by any means (e.g., according to the array from

which the expression levels are obtained, according to the plate from which the correspondent probes are drawn, according to the pins by which the correspondent probes are plotted, or according to the day when the experiment is performed), the expression levels should exhibit the same distribution for different slots.

2.4.1 Quality assessment

The first step is to decide if the data obtained for a microarray is beyond correction and should better be removed from further analysis. There are many ways for quantitatively assess the quality of microarray data. However, the threshold for this type of quality assessment usually lacks consensus among different analyzers, and are quite arbitrary in most of the cases. A simpler and more intuitive way for assessing microarray data is to use visualization techniques.

For example, a first glance at the obtained image of a cDNA microarray can reveal spatial non-uniformity (due to such as damage or contamination on the surface of the microarray, plate effects, and/or pin effects), low contrast between the foreground and the background, and abnormality in the size and shape of spots. In the case of Affymetrix GeneChip, a plot of the log-intensity of the raw microarray data serves the same purpose to check spatial non-uniformity. (The reason to use log is because the largest values in the data are often orders of magnitude larger than the bulk of the data.) Figure 2.4 shows two cases of log-intensity plots from a cDNA microarray, indicating possible contamination on the surface of the microarray and a dominating pin effect. Another useful plot for checking the array effect, plate effect, pin effect is a box plot, (see Figure 2.5). While plate effects and pin effects are removable by normalization methods, array effects due to severe contamination or damage on the surface of microarray are often beyond correction.

2.4.2 Background correction

Before we calibrate microarray data, one might want to subtract background noise from the measured values to purify the signal. The motivation for background adjustment is the belief that a spot's measured intensity includes a contribution not specifically due to the hybridization of the target to the probe, but due to the non-specific hybridization and optical noise [112].

Most of the image processing softwares accompanying cDNA microarray facilities produce spot-specific background fluorescence signal intensities, which is measured from the surrounding areas of each spot [112]. The assumption for such measurement is that the signal intensity measured at the surrounding area of a spot represent the optical noise and noise due to non-specific binding to the spot. Affymetrix GeneChip, instead, use the MM probes to measure the non-specific binding fluorescence intensities.

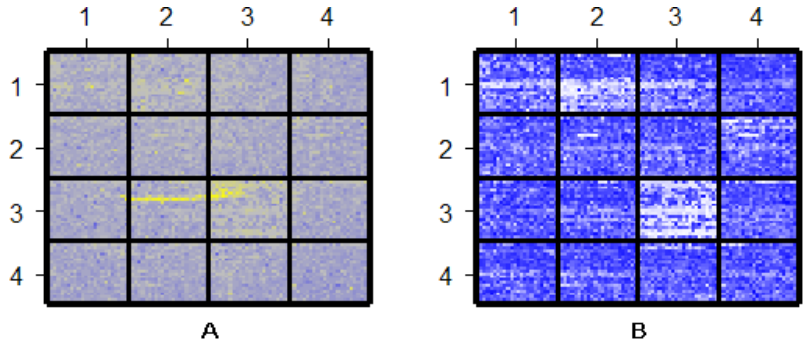


Figure 2.4: Log-intensity plots of a cDNA microarray (i.e., array 81 in the "swirl" data from BioConductor package "marray"). (A) log-intensity ratios between the two channels for each spot on the array, a color toward yellow indicates a higher value while a color toward blue indicates otherwise. The figure indicates that there might be a contamination on the surface of the microarray (the yellow line starting in (1,3) and ends in (3,3)). (B) Added log-intensities of the two channels for each spot on the array. The plot is segmented into 16 areas according to the pins. The plot indicates that there might be an abnormality associated with Pin (3,3).

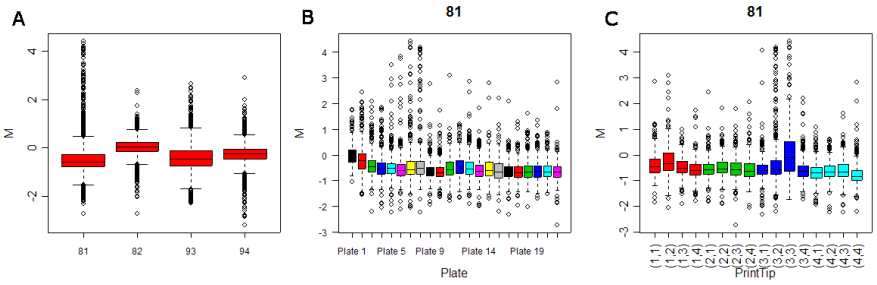


Figure 2.5: Boxplots of the "swirl" data (from BioConductor package "marray"). (A) Boxplot of the log-intensity ratios between the two channels (y-axis) of the spots, calculated for each array (x-axis) in the data. (B) Boxplot of the log-intensity ratios between the two channels (y-axis) of the spots on array 81, calculated for each plate (x-axis). The plot indicates that Plate 1 and 2 might suffer from some plate effects. (C) Boxplot of the log-intensity ratios between the two channels (y-axis) of the spots on array 81, calculated for each pin (x-axis). The plot indicates that there might be some deficiency associated with Pin (3,3).

However, the subtraction of these measured background signals provided by either of the platforms has been under debate. There is evidence showing that the subtraction of the spot-specific background signals measured for a cDNA introduces greater variability around the low-intensity spots than the case when no background subtraction is performed at all [112, 2]. As for the Affymetrix GeneChip, it turned out that the MM probes might be measuring signals as well as non-specific binding, because for data from a typical array, as many as 30% of MM probes have intensities higher than their corresponding PM probes [74]. Furthermore, evidence shows that after subtracting the MM intensity, the information on expression level provided by the different probes for the same gene are still highly variable, and that the variation due to probe effects is larger than variation due to the arrays [67].

While whether to perform the background subtraction of cDNA microarray data remains a personal choice, popular alternatives to subtracting the MM probe intensities include the use of ideal mismatch (IM) [1] and a model based approach, which only uses the PM values, described as the background adjustment in the robust multichip average (RMA) approach [54].

The IM intensity is designed by Affymetrix as a corrected MM intensity, which is guaranteed to be smaller than the corresponding PM intensity. To obtain the IM intensities for a probe set, first a biweight specific background (SB), which is a robust average over the log-ratios between the corresponding PMs and MMs in the probe set, is calculated. If the SB is big (decided by a threshold), it means that the values from the probe set are generally reliable, and if the MM intensity for one of the probe pairs is larger than the corresponding PM intensity, the SB is used to construct the IM, which replaces the MM for the probe pair. On the contrary, if the SB for the probe set is small, Affymetrix smoothly degrades the PM value to calculate the IM value. See [1] for more details.

The background adjustment of the RMA method assumes that the observed PM value is composed of two terms, one generated from a normal distribution, which explains the background noise, and the other being an exponential signal component. The normal distribution is truncated at zero to avoid negative background signals. The model is fit by the expression levels obtained by the PM probes. See [54] for more details.

2.4.3 Normalization

Normalization procedures are designed mainly to calibrate microarray data so as to remove as much as possible systematic noise on each array. At this stage, it is common practice to transform the data to its logarithm. This is especially suited for dealing with expression ratios (coming from two-channel cDNA microarray experiments, using a test and reference sample), since expression ratios are not symmetrical [77] in the sense that upregulated genes have expression ratios between one and infinity, while downregulated genes

have expression ratios squashed between one and zero. Taking the logarithms of these expression ratios results in symmetry between expression values of up- and down-regulated genes. Further more, it is observed that the variance of microarray data increase proportional to the expression level [20]. Taking logarithm makes the noise of microarray data additive.

Various methods have been developed for the normalization of microarray data. Most of them are platform dependent. For the cDNA microarrays, systematic noise that needs to be taken care of include array effects, pin effects, plate effects, and dye effects. Nonlinear normalization methods are found to outperform the linear normalization methods (e.g., total intensity normalization [78], rank invariant methods [104], ratio statistics [20], and analysis of variance models [60]), in terms of correcting systematic biases mainly caused by dye effects [78, 113]. It is observed that the log-ratios between the red and green channels of the data are intensity dependent, and the dependency is often nonlinear. Such effects can be visualized by an MA-plot (see Figure 2.6A). One of the most popular normalization methods for cDNA microarrays is the lowess normalization [113], which is a nonlinear normalization method. To remove the intensity-dependent dye effects, the method performs a lowess fit (i.e., robust locally weighted regression) [23] to the MA-plot, and subtracts the obtained fit from the log-ratios of the intensities. By performing lowess normalization locally to data from the same printing pin, for example, one can calibrate data from different printing pins so that each pin group has zero mean, and thus remove the spatial variation on a chip due to pin effects. We can further regulate the variance of the data by rescaling the variation of the log-ratios, so that data printed by different pins have the same stretch on the log-ratios. (Figure 2.6B, 2.6C and 2.6D show the result after the normalization procedure described above is performed on the example data.) Of course, the same method can be applied to data from different arrays, or different plates to remove the array effects and the plate effects.

For Affymetrix CeneChip, the task of normalization is to remove the array effects and the probe effects. Both linear and nonlinear methods exist for the normalization of Affymetrix. Similar to the case for cDNA microarrays, the nonlinear normalization methods tend to outperform the linear ones (such as the scaling method used in the software of Affymetrix [1]). A popular nonlinear normalization method for Affymetrix CeneChip data is quantile normalization [17], whose goal is to impose the same empirical distribution of intensities to each array. The empirical distributions of the intensities are represented by the quantiles. The algorithm first ranks the probe intensities from the lowest to highest for each array, so that each rank represents a quantile. Then, the average intensity value across all the arrays is calculated within each quantile. Finally, the measured intensity in a given quantile in an array is replaced by the calculated average intensity for that quantile. After the normalization procedure, the data measured by different probe sets need to be summarized to produce one measure for the expression level of each gene on each chip.

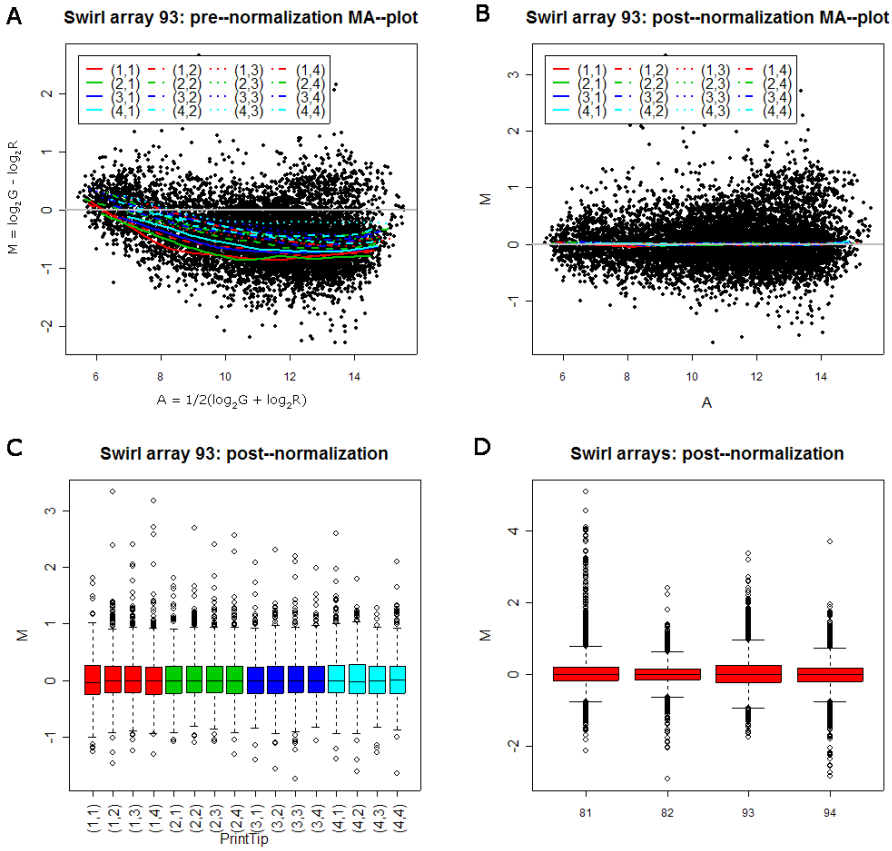


Figure 2.6: Lowess normalization of the "swirl" data (provided by BioConductor package "marray") (A) MA-plot showing the intensity dependent effects on the log-ratios obtained for Array 93. The x-axis presents the intensity of the genes by the A (add) values, which are the added log-intensities of the red (R) and green (G) channels. The y-axis gives the M (minus) values of the genes, which are the log-ratios of the two channel intensities. Different line types illustrate the lowess fit for different printing pins. (B) MA-plot of Array 93 after the normalization. (C) Box-plot of post-normalized data for Array 93, grouped according to different printing pins. (D) Box-plot of post-normalized data from different chips.

Common summarization methods include average difference (which simply computes the average difference between PM and MM intensities over all the probe sets of a gene), one-step Tukey's biweight estimate [1], median polish fit [105] to a linear model describing the log-intensities as a three-term-addition (with one term being the true log expression level, another one describing the probe effect and the third one for normally distributed noise) [54].

Besides the platform dependent normalization methods mentioned above, the variation stabilization normalization (VSN) [52, 30], a normalization method that works both for cDNA microarrays and Affymetrix CeneChip is receiving increasing attention. It combines the background correction, the nonlinear transformation, and the normalization procedures together. The motivation of the method is to solve the problem of the multiplicative noise feature of raw microarray data. While the assumption of the proportionality between the noise and the intensity of microarray data holds for genes with a relatively large intensity, it does not continue down to the genes that are unexpressed at all because the proportionality would imply zero measurement noise for those genes [81]. While this problem can roughly be solved by eliminating the observed intensities that are close to the background intensities, and take the logarithm on the rest of the data, the VSN model is proposed [81] to transform the microarray data incorporating a nonlinear term that approximates the natural logarithm for large intensities and a linear term for intensities that are around 0 [30]. Therefore, by using VSN, the variance of the intensities becomes approximately independent of the mean [52], and data in the whole range can be preserved for further analysis, which is certainly more favorable. There exist several ways for specifying the VSN transformation, which is associated with different names such as the general logarithm [82] and the *arcsinh* function [52]. However, the basic form of these models (i.e., transformations) is the same,

$$f(y) = \log\left(a(y) + \sqrt{a(y)^2 + 1}\right), \tag{2.1}$$

where y is the measured intensity, and a is specified individually by different realizations of the method [52, 30, 31]. In addition, these different realizations are all motivated by the model that describes the raw microarray data (y) by two terms [81], one for the normally distributed background intensities, and the other for the real intensities of the genes with an exponential error,

$$y = \alpha + \epsilon + \mu e^{\eta}, \tag{2.2}$$

where α is the mean background intensity, ϵ is the additive noise, μ is the true expression level, and η is the multiplicative noise. Both ϵ and η are independently and normally distributed with mean zero. The resulting variance of the observed intensities by applying this model is a quadratic function of the mean of the intensities,

$$\mathrm{Var}(y|\mu) = s_\eta^2 \mu + s_\epsilon^2, \tag{2.3}$$

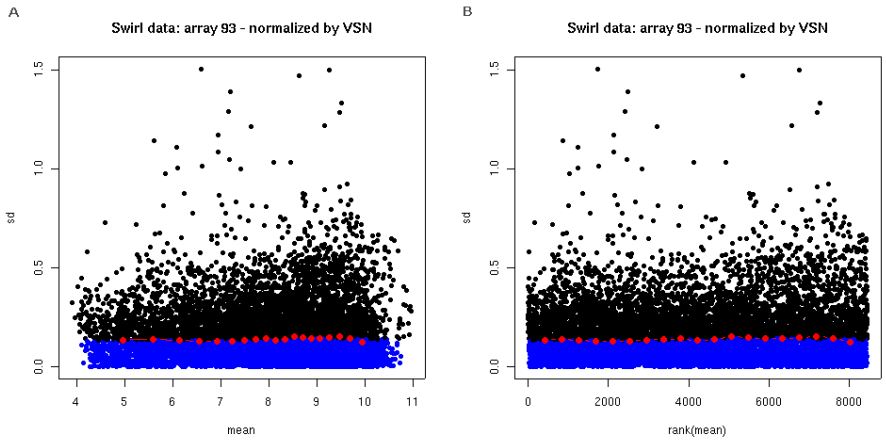


Figure 2.7: Result of VSN on the swirl data for Chip 93. The red dots depict the running median estimator. If there is no variance-mean dependence, the line formed by the red dots should be approximately horizontal. Each dot in the plots represents a gene. In each plot, the y-axis represents the standard deviation of the normalized intensity of the red and the green channels, and the x-axis represents (A) the average (i.e., mean) of the normalized intensity of the red and the green channels (B) the rank of the mean.

where s^2_η is the variance of e^η, and s^2_ϵ is the variance of ϵ. Figure 2.7 illustrates the result of VSN on the example data.

The normalization methods mentioned above are only a small portion of the various approaches available that are motivated by different problems embedded in the microarray technologies and the experimental designs. The choice of the normalization method should be data specific and is related to the specific biological question under study. Efforts in investigating the optimal normalization method are well deserved for each study.

2.5 Specific characteristics of microarray data

To conclude this chapter, it is necessary to mention the unique characteristics of (preprocessed) microarray data, which should be taken into account for any further analysis.

First of all, microarray data typically have an asymmetric dimensionality. The preprocessed microarray data is usually put in a matrix, where the genes are represented as rows and the experiments are listed in the columns. Each row (i.e., the expression levels of a gene across different experiments) is the expres-

sion profile of the gene. While the number of rows of the matrix can contain tens of thousands of genes, the dimension of the columns is much smaller. The current cost for a microarray chip limits the number of experiments in a study. Most labs can afford studies up to a few tens of experiments. A relatively large study can use up to a couple of hundred of chips, which is still a small number comparing to the number genes.

Secondly, although a proper preprocessing procedure removes as much as possible systematic noise from the data, the noise presented in the resulting data for further statistical analysis is still non-negligible.

Thirdly, microarray experiments often contain missing values. These values typically comes from probes that encounter complications in the measuring procedure (see Section 2.3) and whose values are beyond correction. Simple replacements such as a replacement by zero or by the average of the expression profile often disrupt these profiles. Indeed, replacement by average values relies on the unrealistic assumption that all expression values are similar across different experimental conditions. More advanced techniques of missing value replacement (which use the *k*-nearest neighbor method or the singular value decomposition) have been described [103] and take advantage of the rich information provided by the expression patterns of other genes in the data set. A more favorable way, however, is to obviate the need for missing value replacement by devising the statistical tools for microarray data analysis so that only the measured values are used.

Finally, the biological process under scrutiny in a microarray study is assumed to be a complicated process, which involves concerted gene reactions in different pathways. While some genes can even be involved in more than one pathway, some others, however, might not be relevant to the biological process. These genes usually show little variation over the different experiments under study. Genes that show little variation over the different experiments are called constitutive with respect to the biological process studied. Constitutive genes often contribute to a large proportion of the whole population of the genes included in a microarray study. A conventional way to handle these genes is to remove the gene expression profiles from the data that do not satisfy some simple criteria [32]. Commonly used criteria include (1) a minimum threshold for the standard deviation of the expression values in a profile and a threshold on the maximum percentage of missing values, and (2) a minimum threshold on the interquartile range (IQR) of a gene across the experiments.

Chapter 3

Clustering microarray data

This chapter gives a survey of (bi)clustering algorithms that have been applied to microarray data. We dedicate the first main section of this chapter to the overview of three traditional clustering techniques in the machine learning world—hierarchical clustering, K-means clustering, and SOM—which are popular choices in the early practice of clustering microarray data. Then, we focus our discussion of existing model-based algorithms for clustering microarray data as a base to motivate our application of Bayesian models (which will be explained in the three following chapters). Our discussion is then further extended to biclustering algorithms. In this regard, we first list different types of biclustering algorithms, and point out the type of biclustering algorithm that is the focus of this thesis. Then, we provide a brief survey about existing biclustering algorithms that fall in the same category as ours. Finally, we give a review of various methods to validate the clustering results.

3.1 Introduction

The first level of interest for molecular biologists is to identify genes whose expression level is significantly changed under different experimental conditions. Basic statistical techniques can be applied to solve this problem efficiently [7, 106, 94, 92]. However, such an analysis treats the genes as individuals rather than exploring their relation with each other. On the other hand, for every gene, the detailed information about its expression profile as a whole over all the experiments under study is neglected in this first-level analysis. To make better use of the full-scale information provided by microarray experiments, the next level of insight is provided by clustering genes into biological meaningful groups according to their pattern of expression. Comparing with the full data itself, such groups of related genes are much more tractable for

further studies including gene function and regulation.

Based on the assumption that expressional similarity implies functional similarity of the genes (and vice versa), the challenge of finding genes that might be involved in the same biological process is thus transformed to the problem of clustering genes into groups based on their similarity in expression profiles. Genes that have similar expression profiles are said to be coexpressed.

The first generation of clustering algorithms (e.g., hierarchical clustering [32], *k*-means [46] and self-organizing maps (SOM) [62]) applied to gene expression profiles were mostly developed outside biological research. Although encouraging results have been produced [93, 97, 95], some of their characteristics often complicate their use for clustering expression data [90]. They require, for example, the predefinition of one or more user-defined parameters that are hard to estimate by a biologist (e.g., the predefinition of number of clusters in *k*-means and SOM – this number is almost impossible to predict in advance). Moreover, changing these parameter settings will often have a strong impact on the final result. These methods therefore need extensive parameter fine-tuning, which means that a comparison of the results with different parameter settings is almost always necessary – with the additional difficulty that comparing the quality of the different clustering results is hard. Another problem is that the first-generation algorithms often force every data point to belong to a cluster. In general, a considerable number of genes included in the microarray experiments do not contribute to the studied biological process, and these genes will therefore have seemingly constant or even random expression profiles rather than having similar expression profiles with the other genes. Including these "noisy" genes in one of the clusters will contaminate their content and make these clusters less suitable for further analysis.

In addition to the above limitations of first-generation clustering algorithms, the specific characteristics of microarray data, such as the high dimensionality, the highly noisy measurements, the complex biological processes hidden behind, in general have created the need for clustering methods to be tailored to these specific requirements. Accordingly, desired features for microarray data cluster analysis include fast calculation speed, robustness, easy interpretation, and so on.

A second generation of clustering algorithms has started to tackle some of the limitations of the earlier methods, while seeking to meet those specific requirements for microarray data. These algorithms include model-based algorithms [116, 44, 71], the self-organizing tree algorithm [50], quality-based algorithms [51, 91], simulated annealing [69], the cluster affinity search technique [13], and biclustering algorithms [21, 89, 47, 11]. Also, some procedures were developed that could help biologists to estimate some of the parameters needed for the first generation of algorithms (such as the number of clusters present in the data [44, 69, 116]).

In particular, algorithms based on probabilistic models have become a popular

choice for the analysis of microarray data [116, 71, 87] because of its ability to handle the complex nature of the data. A proper probabilistic model can capture the essential information presented by the data, but in the same time, it allows variability in the biological system.

In this chapter, we first give a survey of the use of classical clustering methods, namely hierarchical clustering, k-means clustering, and SOM, on microarray data, in order to discuss the need of tailored clustering techniques for microarray data. We then particularly focus the discussion of clustering algorithms for microarray data analysis on the model-based algorithms to see their advantages and disadvantages. Finally, we provide an overview on the current status of researches on biclustering algorithms for microarray data as a comparison with our method. To conclude this chapter, a list of popular methods to assess the quality of the clusters for microarray data is provided.

3.2 Standardization of gene expression profiles

Before we dive into the detailed discussion of various clustering algorithms, we would like to remind the readers of the importance of the standardization of gene expression profiles for the clustering problems of microarray data. Biologists are mainly interested in grouping gene expression that have the same relative behavior; i.e., genes that are up- and downregulated together. Genes showing the same relative behavior but with diverging absolute behavior (e.g., gene expression profiles with a different baseline or a different amplitude but going up and down at the same time) will have a relatively high Euclidean distance (see the next section for details). Cluster algorithms based on this distance measure will therefore wrongfully assign the genes to different clusters. This effect can largely be prevented by applying standardization or rescaling to the gene expression profiles to have zero mean and unit standard deviation. Gene expression profiles showing the same relative behavior will have a small(er) Euclidean distance after rescaling [77].

3.3 Classical clustering methods

Classical clustering methods originated in the machine learning society have shown success in mining microarray data. Some of them, e.g., hierarchical clustering, still remain as popular choices for microarray data analysis. We review three classical clustering algorithms here, and then provide a discussion on their deficiency with regard to the special characteristics of microarray data, and give a list of desired features for clustering algorithms for microarray data.

3.3.1 Distance metrics

All the three classical clustering analysis that we discuss here measure the similarity between objects (i.e. in our case, gene expression profiles, or when clustering experiments, the expression levels of genes in each experiments) by means of distance metrics. A cluster contains data points whose pair wise distance between one another is lower than a threshold value. Here, the problem arises from how the distance should be defined. Hereunder, we give a brief overview of common distance metrics applied for clustering microarray data.

1. *Pearson correlation:* This is the most commonly used distance metric. It is often denoted by r. Pearson correlation is the cosine of the angle between two vectors. This means that it measures the similarity in the shapes of two profiles, while not taking the magnitude of the profiles into account. Therefore, Pearson correlation suits well the intuition of biologists for what they mean when saying that two expression profiles are "coexpressed" [32].
2. *Squared Pearson correlation:* This is the squared product of Pearson correlation. Therefore, squared Pearson correlation r^2 neglects the sign in front of a Pearson correlation measure and considers two vectors pointing to the exact opposite directions to be perfectly similar (i.e., in this case, $r = -1$ while $r^2 = 1$). This character makes the squared Pearson correlation able to capture inverse relationships among gene expression profiles, which might also be interesting for biologists.
3. *Euclidean distance:* Euclidean distance measures the absolute distance between two points in space. That is, it is the length of the straight line connecting the two points. As mentioned previously, the Euclidean distance measures the similarity between the absolute behaviors of genes, while the biologists are more interested in their relative behaviors. Thus, when using Euclidean distance metric, a standardization procedure is needed before clustering can be applied to the gene expression profiles. Note here that after the standardization, the Euclidean distance is related to the Pearson correlation between two points x and y by $|x - y|^2 = 2(1 - r)$ [3].

3.3.2 Hierarchical clustering

Hierarchical clustering was first applied in biology was for the construction of phylogenetic trees. Early applications of the method to gene expression data analysis [32, 93] have proved its usefulness.

Hierarchical clustering has almost become the *de facto* standard for gene expression data analysis, probably because of its intuitive presentation. The whole clustering process is presented as a tree called a dendrogram, the original data are often reorganized in a heatmap demonstrating the relationships between genes or conditions.

Two approaches to hierarchical clustering are possible: divisive clustering (a top-down approach as is used in [3]), and agglomerative clustering (a bottom-up approach, see for example [32]). In agglomerative clustering [32], each expression profile is initially assigned to one cluster; at each step, the distance between every pair of clusters is calculated and the pair of clusters with the minimum distance is merged; the procedure is carried on iteratively until a single cluster containing all the expression profiles is obtained. Divisive clustering works the other way round, initially, all the gene expressional profiles are treated as belonging to one cluster; in each step, a cluster is divided so that the resulting clusters are as far away from each other as possible. Different techniques for dividing the clusters are available for divisive hierarchical clustering [46, 57]. In general, the stepwise computational complexity is simpler in agglomerative clustering than the divisive clustering, but the latter is more useful when one is more interested in the main structure of the data [57] or when the number of clusters (say, k) presented in the data is known in advance [29] (since the stopping criteria for the algorithm can then be modified so that the splitting procedure is no longer performed when k clusters are produced).

However, this advantage of the divisive approach is not often helpful in the analysis of gene expression data, because for the visualization of the resulting reorganized microarray data, see Figure 3.1, a full dendrogram is usually desired by biologists. Therefore, the structure of the dendrogram remains an important problem, because although the dendrogram itself does not determine the clusters for the users, a good ordering of the leaves can help the users to identify and interpret the clusters. A heuristic approach aiming to find a good solution was developed [32] by weighting genes using combined source of information, and then placing the genes with lower average weight earlier in the final ordering. Further, [10] reported a dynamic programming method that helps to reduce the time and memory complexities for solving the optimal leaf-ordering problem.

After the full tree is obtained, the determination of the final clusters is achieved by cutting the tree at a certain level or height, which is equivalent to putting a threshold on the pair wise distance between clusters. Note that the final cluster partition is thus rather arbitrary.

Whether agglomerative or divisive, hierarchical clustering in general has several drawbacks. Hierarchical clustering can never repair a decision (to merge in agglomerative clustering, and to split in the divisive one) made in previous steps [57]. It is, after all, based on a stepwise optimization procedure rather than finding k optimal clusters globally. Another disadvantage of hierarchical clustering is that the nature of the hierarchical clustering (where data points are forced into a strict hierarchy of nested subsets) fits more into the context of building a phylogenetic tree [95] rather than that of grouping expression profiles.

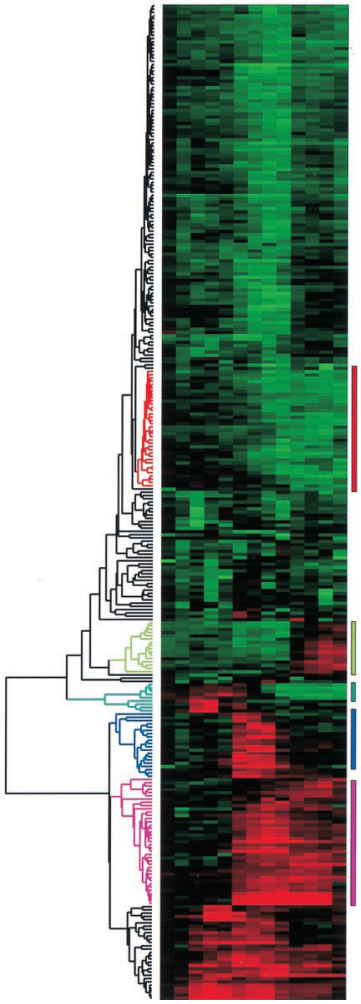


Figure 3.1: Visualization of the results of hierarchical clustering. A heatmap presenting the gene expression data, with a dendrogram to its side indicating the relationship between genes (or experimental conditions) is the standard way to visualize the result of hierarchical cluster analysis on microarray data. The length of a branch in the dendrogram is proportional to the pair wise distance between the clusters. Importantly, the leaves of the dendrogram, and accordingly the rows of the heatmap, can be swapped (without actually changing the information contained in the tree) so that the similarity between adjacent genes are maximized, and hence the patterns embedded in the data become obvious in the heatmap. The figure is obtained from Eisen *et al*. (1998).

Distance measure between two clusters

Because agglomerative clustering is more popular in microarray data analysis, here we take a deeper look at the algorithm. As we mentioned, in every step of agglomerative clustering, the two clusters that are closest to each other will be merged. Here comes the problem of how we define the distance between two clusters. There are four common options:

1. *Single linkage:* The distance between two clusters is the distance between the two closest data points in these clusters (each point taken from a different cluster).
2. *Complete linkage:* The distance between two clusters is the distance between the two furthest data points in these clusters.
3. *Average linkage:* Both single linkage and complete linkage are sensitive to outliers [29]. Average linkage provides an improvement by defining the distance between two clusters as the average of the distances between all pairs of points in the two clusters.
4. *Ward's method:* At each step of agglomerative clustering, instead of merging the two clusters that minimize the pair wise distance between clusters, Ward's method [107] merges the two clusters that minimizes the "information loss" for the step. The "information loss" is measured by the change in the sum-of-squared-error of the clusters before and after the merge. In this way, Ward's method assesses the quality of the merged cluster at each step of the agglomerative procedure.

These methods yield similar results if the data consist of compact and well-separated clusters. However, if some of the clusters are close to each other or if the data have a dispersed nature, the results can be quite different [29]. Ward's method, although less well known, often produces the most satisfactory results.

3.3.3 *K*-means clustering

K-means clustering [46] is a simple and widely used partitioning method for data analysis. Its helpfulness in discovering groups of coexpressed genes has been demonstrated [97].

The number of clusters k in the data is needed as an input for the algorithm. The algorithm then initializes the mean vector for each of the k clusters either by hard assignment (e.g., from the input, or by random generation). These initial mean vectors are called the seeds. Next, the k-means algorithm proceeds iteratively with the following two steps (1) using the given mean vectors, the algorithm assigns each gene (or experiment) to the cluster represented by the closest mean vector, (2) the algorithm recalculates the mean vectors (which

are the sample means) for all the clusters. The iterative procedure converges when all the mean vectors of the clusters remain stationary.

From the above description, *k*-means method can be understood as a distance-based approach. Both the Pearson correlation and the Euclidean distance can be used as the distance measure. However, the squared Pearson correlation metric should not be used in combination with the *k*-means algorithm, since this distance measure takes the inverse relationships between the clustering subjects into account, which can lead to problems when calculating the means. On the other hand, *k*-means algorithm is closely related to model-based methods. When the Euclidean distance metric is used, the step of recalculating the means of the clusters actually corresponds to minimizing the within-cluster sum of squared distances from the cluster mean [102]. We will show in Section 3.5.1 that in this case the *k*-means algorithm is an approximation to the expectation-maximization (EM) method for Gaussian mixture model parameterization [8].

A significant problem associated with *k*-means algorithm is the arbitrariness of predefining the number of clusters, since it is difficult to predict the number of clusters in advance. In practice, this implies the use of a trial-and-error approach where a comparison and biological validations of several runs of the algorithm with different parameter settings are necessary [73]. Another parameter that will influence the result of *k*-means clustering is the choice of the seeds. The algorithm suffers from the problem of converging to local minima (of the likelihood function, when explaining the method as a model-based approach, see Section 3.5.1). This means that with different seeds, the algorithm can yield very different result. Preferably, the seeds should be chosen close to the center of the natural clusters. Of course, this is hard to achieve if no prior knowledge about the clusters is available, which is often the case. Using principal component analysis (PCA) to provide prior knowledge on the number and the means of the clusters was proposed in [77].

3.3.4 Self-organizing maps

SOM [62] is a technique to visualize the high-dimensional input data (in our case, the gene expression data) on an output map of neurons, which are sometimes also called nodes. The map is often presented in a two-dimensional grid (usually of hexagonal or rectangular geometry) of neurons. In the high-dimensional input space, the structure of the data is represented by prototype vectors (serving similar functions as the mean vectors in the *k*-means algorithm), each of which is related to a neuron in the output space.

As an input for the algorithm, the dimension of the output map (e.g., a map of 6×5 neurons) needs to be specified. After initializing the prototype vectors, the algorithm iteratively performs the following steps. (1) Every input vector (e.g., representing a gene expression profile) is associated with the closest prototype vector, and thus is also associated with the corresponding neuron

on the output space. (2) Update the coordinates of a prototype vector based on a weighted sum of all the input vectors that are assigned to it. The weight is given by the neighborhood function (a kernel function in nature), which can be a Gaussian distribution function, applied in the output space. That is, in the updating step, a prototype vector is pulled more toward input vectors that are closer to the prototype vector itself and is less influenced by the input vectors located farther away. In the meantime, this adaption procedure of the prototype vectors is reflected on the output nodes – nodes associated with similar prototype vectors are pulled closer together on the output map. (3) To put a simulated annealing kind of flavor, the initial variance of the Gaussian neighborhood function is chosen so that the neighborhood covers all the neurons, but then the variance is decreased in every iteration so as to achieve a smoother mapping. The algorithm terminates when convergence of the prototype vectors is achieved.

Instead of the batch procedure described above, there is also an online procedure for SOM training. The only difference is, in the online procedure, a random input vector is picked one at a time at the start of the iteration and all the prototype vectors (together with the neurons) are then adjusted accordingly.

From the cluster analysis point of view, SOM methods looks similar to *k*-means methods. SOM clustering differs from *k*-means clustering in that a cluster has two "faces" in a SOM – it is represented by the prototype vector in the input space and the neuron on the output space. In this way, a SOM provides a direct means to visualize relations among different clusters. Moreover, a prototype vector is adjusted according to not only the data points that are associated with it but also data points that are assigned to other prototype vectors. SOM clustering is reported to have satisfactory results on gene expression data [95, 101]. In these experiments, every neuron represents a cluster. As a result, clusters that represents similar gene expression profiles are located closer on the output map, while clusters with anti-correlated expression profiles are put into opposite corners of the grid [90].

Because of the advantage in visualization, choosing the geometry of nodes for a SOM is not as crucial a problem as the choice of the number of clusters for a *k*-means method. Of course, initializing an SOM with too few nodes will result in non-representative and non-distinctive clusters. However, on the contrary, if too many nodes are added to a SOM, clusters with great similarity (located in the same neighborhood on the output map) can be merged to get a more extensive cluster. Based on this idea, a tree-structured SOM clustering method is implemented for gene expression data in [101]. Like the *k*-means method, the initial choice of prototype vectors remains a problem that influence the final clustering result of SOM clustering. A good way to seed the prototype vectors can be the result from a PCA analysis [62].

3.4 A wish list for clustering algorithms

The limitations of the classical clustering algorithms together with the specific characteristics of gene expression data call out for clustering methods tailored for microarray data analysis. Collecting the lessons from the classical clustering algorithms and the demands defined by the specific characteristics of microarray data, we compose here a subjective wish list of the features of an ideal clustering method for gene expression data.

A problem shared by the classical clustering algorithms is the decision of the number of clusters in the data. In *k*-means clustering and SOM clustering, this decision has to be made before the algorithms are executed, while in hierarchical clustering it is postponed till the full dendrogram is formed, where the problem then is to determine where to cut the tree.

Another problem of the classical clustering algorithms is that they all assign every gene in the data set (even outliers) to a particular cluster. Because microarrays perform expression profiling for the whole genome, it is possible that some measured genes do not contribute to the biological process under study. Consequently, the measured expression levels of these genes mostly represent noise. Therefore, it is not sensible to include these in any of the clusters. A proper filtering step in the preprocessing, which use a threshold on the variation of the gene expression profiles, (i.e., variation filter) helps to reduce the number of these genes in the data set. However, it is insufficient. Therefore, a clustering algorithm should be able to identify genes that are not relevant for any clusters and leave them as they are.

A third problem is robustness. For all the three clustering techniques addressed above, difference in the choice of distance metrics (either for between the expression profiles or between the clusters) will result in different final clusters. In *k*-means clustering and SOM clustering, the choices of seeds for the mean vectors or the prototype vectors also greatly influences the result. Taking into account the noisy nature of microarray data, improving the robustness should be one of the goals when designing novel clustering algorithms for gene expression data.

Finally, the biological process under study in a microarray experiment is a complicated process where genes interact with each other in different pathways. Consequently, a gene under study might be directly or indirectly involved in several pathways. With this idea in mind, clustering algorithms that allow a gene to belong to multiple clusters would be favorable.

The desirable properties here are not exhaustive, but they give a number of clear directions for the development of clustering algorithms tailored to microarray data.

3.5 Model-based approaches for gene expression data

Model-based clustering [46] is an approach that is not really new and has already been used in the past for other applications outside bioinformatics. However, its potential use for cluster analysis of gene expression profiles has been proposed only recently [116, 44, 71],compared with those are introduced in Section 3.3. Most importantly, we talk about these methods separately because they provide a base for our discussion on applying Bayesian models to microarray data.

Model-based clustering assumes that the data are generated by a finite mixture of underlying probability distributions, where each distribution represents one cluster. The problem, then, is to associate every gene (or experiment) with the best underlying distribution in the mixture, (the assignment of the objects is often considered as the missing data of the problem), and at the mean time, to find out the parameters for each of these distributions. In a classical view of probabilistic models, the problem is solved by finding the parameters of the distributions that optimizes the likelihood computed on the complete data (i.e., the observed data plus the missing data).

Regardless of the choice of underlying distributions, a mixture model is usually learned by an expectation-maximization (EM) algorithm. Given the microarray data and the current set of model parameters, the probability to associate a gene (or experiment) to every cluster is evaluated in the E step. Then, the M step finds the parameter setting that maximizes the likelihood based on the complete data. The complete data refers to both the (observed) microarray data and the assignment of the genes (or experiments) to the clusters. The likelihood of the model increases as the two steps iterates, and convergence to a stable solution is guaranteed under general conditions [111].

In what follows, we discuss respectively the work of Yeung *et al*. [116] where a mixture model of normal distributions is used for the cluster analysis of microarray data, and the work of McLachlan *et al*. [71] who use a mixture of factor models for clustering experiments.

3.5.1 Mixture model of normal distributions

When multivariate normal distributions are used, each cluster is represented by a hypersphere or a hyperellipsoid in the data space. The mean of the normal distribution gives the center of the hyperellipsoid, and the covariance of the distribution specifies its orientation, shape, and volume. The covariance matrix for each cluster can be represented by its eigenvalue decomposition, with the eigenvectors determining the orientation of the cluster, (i.e., the principal axis of the hyperellipsoid) and the eigenvalues specifying the shape and the volume of the cluster. (Note that the similarity of the objects in a cluster is mea-

sured by the volume of the hyperellipsoid—the smaller the volume, the more similar the objects are with each other.) By using different levels of restrictions on the form of the covariance matrix (i.e., its eigenvectors and eigenvalues), one can control the trade-off between model complexity (the number of parameters to be estimated) and flexibility (the extent to which the model fits the data).

The choice of the normal distribution is partly based on its desirable analytic convenience. Moreover, the assumption for fitting normal distribution to gene expression profiles is considered to be reasonable especially when the proper preprocessing procedures (see Chapter 2) have been applied [116, 7]. Of course, other underlying distributions, such as gamma distributions or mixtures of Gaussian and gamma distributions, can also be used to describe expression profiles [109]. So far, no precise conclusions have been made on what is the most suitable distribution for gene expression data [7].

The EM procedure is repeated for different numbers of clusters and different covariance structures. The result of the first step is thus a collection of different models fitted to the data and all having a specific number of clusters and specific covariance structure. Then, the best model with the most appropriate number of clusters and covariance structure in this group of models is selected. This model selection step involves the calculation of the Bayesian information criterion (BIC) [84] for each model.

Yeung *et al*. [116] reported good results of such analysis as described above using their MCLUST software [33] on several synthetic data sets and real expression data sets.

Relation of normal mixture learning with *k*-means approach

The *k*-means algorithm described in Section 3.3.3 can be viewed as a special case of applying EM algorithm to learn a mixture model of normal distributions where all the normal distributions in the mixture are characterized by spheres of the same volume but different means (the covariance matrices have the same form of an identity matrix multiplied by a constant). The E step of an EM algorithm is to evaluate the conditional probability of the latent variable of a data point (the latent variable indicates to which cluster the data point belongs) based on the observed data (in our case, the microarray data) and the current parameters. In this context, *k*-means method uses an index function (i.e., a distribution function of zero variance) to replace the conditional distribution and thus performs a hard assignment to put a data point in the cluster whose mean (i.e., center) is the closest. The next iterative step of *k*-means clustering is to minimize the cost function of sum-of-squared-distance within the cluster. Using this index function as the conditional distribution, the minimum of this cost function is exactly the maximum of the log-likelihood function [46]. Thus, this second step of *k*-means corresponds to the M step in the EM algorithm in this case. The parameters from the point of view of the optimization are the

sample means of the data points of the clusters.

3.5.2 Mixture model of t distributions and mixture of factor models

For the clustering experiments (e.g., tissue samples), however, problem rises for fitting a normal mixture to the data because the number of genes is much larger than the number of experiments. To solve this problem, [71] applied mixture of factor analysis to the clustering of experiments (see Figure 3.2). The idea can be interpreted as follows. A single factor analysis performs a dimensional reduction in the gene space of a cluster. That is to say, in factor analysis, vectors of experiments located in the original n-dimensional hyperellipsoid (where n represents the number of genes) are projected onto their corresponding vectors of factors located in an m-dimensional unit sphere (usually $m << n$). By using a mixture of factor analysis, clustering of the experiments is done on a reduced feature space (i.e., the m-dimensional factor space) instead of on the original huge dimensional gene space. The EM algorithm is also used to learn the mixture of factor analysis model.

However, the choice for the number of factors in such a model remains a dilemma. If the number is too small, the full correlation structure of the genes cannot be captured; while if it is too large, the EM algorithm for the parameterization of the model can encounter computational difficulties. To alleviate the problem, [71] added another stage to reduce the dimension of the gene space before applying the mixture of factor analysis to the clustering of the experiments. In this stage, both a two-component mixture model of univariate t distributions (where the association of the experiments to the two components is unknown) and a single t distribution are fit to the data for each gene. A threshold on the likelihood ratio between the two models is then applied to determine whether the gene is responsible for the clustering of experiments.

A t mixture model is more suitable for describing a gene expression profile than a normal mixture model because the former is more robust to outliers. A t distribution has an additional parameter called the degree of freedom compared to a normal distribution. The degree of freedom can be seen as a parameter for adjusting the thickness of the tail of the distribution. A t distribution with a relative small degree of freedom will have a thicker tail than a normal distribution with the same mean and variance. However, as the degree of freedom goes to infinity, the t distribution approaches the normal distribution. Because of the thicker tail of a t distribution, the model learned for the t mixture is more robust to the outliers in gene profiles. Therefore, the degree of freedom can be viewed as a robustness tuning parameter.

Their software EMMIX-GENE (where these methods are implemented) yields promising results on several biological data sets.

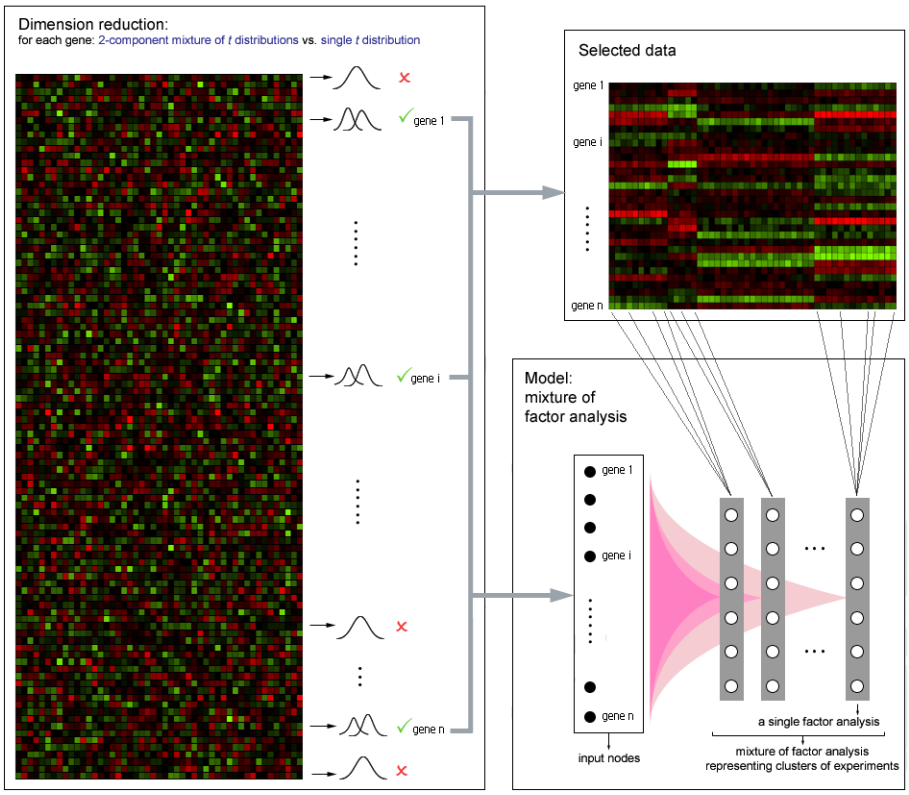


Figure 3.2: McLachlan *et al*. (2002) uses a two-component mixture model of t distributions to examine every gene expression profile against a single t distribution. Expression profiles to which the mixture models fit better (in terms of, for example, likelihood) are selected for further analysis. A mixture of factor analysis is applied on the selected data to cluster the experimental conditions.

3.6 Biclustering algorithms

In addition to discovering the relationship between the genes or the relationships between the experiments as in conventional clustering algorithms, biclustering methods also explore the relationships between the genes and the experiments. The prefix in the word biclustering indicates that this is a technique to cluster both the genes and the experiments at the same time.

Although the existing biclustering algorithms originate from the same idea, they differ from each other by their particular emphasis on the problems they try to solve. Among early papers on biclustering methods, clustering algorithms are applied (iteratively) to both rows and columns of a microarray data set. As a result, genes and experiments are reorganized so as to improve the manifestation of the patterns inherited in both the genes and the experiments. In other words, these algorithms divide the data into checkerboard units of patterns. Examples of these algorithms are in the works by Alon *et al*. [3] and Getz *et al*. [42], in which existing clustering algorithms are used for the task. In addition, there are also algorithms particularly designed for this purpose. In the paper of Lazzeroni and Owen (2002) [65], a mixture model of normal distributions, which is called the plaid model by the authors, is used to describe the microarray data and EM is applied for parameter estimation. For another example, the spectral biclustering method [61] applies singular value decomposition for solving the problem.

However, this type of biclustering algorithms have its limitation when the expression profiles of some genes under study divides the samples in correspondence with one biological explanation (say, tumor type) while profiles of another subset of the genes divides the samples according to another biological process (e.g., drug response) [47]. The second type of biclustering algorithm aims to find genes that are responsible for the classification of the samples. Examples are the gene shaving method [47], which searches for clusters of genes that vary as much as possible across the samples with the help of PCA, and a minimum description length method [56] that identifies gene clusters responsible for classification of experimental conditions.

The third type of biclustering algorithms question conventional clustering algorithms by the idea that genes that share functional similarities do not have to be coexpressed over all the experimental conditions under study. Instead of clustering genes based on their overall expressional behavior, these algorithms look for patterns where genes share similar expressional behavior over only a subset of experimental conditions. The same idea can be used for clustering the experimental conditions. Suppose a microarray study is carried out on tumor samples of different histopathological diagnosis. The problem then is to find tumor samples that have similar gene expression level for a subset of genes (so as to obtain an expressional fingerprint for the tumor). To distinguish the two orientations for this type of biclustering problem, we will refer to the former

case as biclustering genes, and the latter case as biclustering experiments, (see Figure 1.6 for an illustration of the two problems).

This type of biclustering algorithms is pioneered by Cheng and Church (2000) [21], where a heuristic approach is proposed to find patterns as large as possible that minimizes the mean squared residues of their objective function while allowing variance to be present across the experiments when biclustering genes (or across the genes when biclustering experiments). Since then, there has been active research on this type of biclustering problems. Our method was first proposed in Sheng *et al*. (2003) [89], which was among the early papers on this subject. Other early papers include the following. Tanay *et al*. (2002) [96] (known as SAMBA) discretizes the gene expression values into three levels—upregulated, inactive, and downregulated—and represents the discrete matrix as a bipartite graph, whose nodes represent the rows on one side and columns on the other. The edges in the graph have two values—'+1' for upregulation and '-1' for down regulation. At matrix entries whose value is 0 (i.e., inactive), there is no edge between the corresponding rows and columns in the bipartite graph. The method then uses a heuristic approach to find bicliques in the graph, which correspond to biclusters in the matrix. Bergmann *et al*. (2003) [14] describes a "iterative signature algorithm" which starts from a sufficiently large set of random genes, and selects the experimental conditions where the average expression value of these genes is above a threshold. Then the algorithm computes the correlation between a gene and the average profile of the genes in the bicluster under the selected conditions and iterates. Another model-based approach for this type of problem is provided by Barash and Friedman (2002) [11], where the EM algorithm is used for estimating the parameters of the model.

Most of algorithms of this type are not able to incorporate prior knowledge, one exception is the signature algorithm (see Ihmels *et al*. (2002) [53] for applications of the signature algorithm from this aspect). In contrast, Bayesian models provide our method an systematic base for a for integration of prior knowledge and information from other data sources.

The work of Segal *et al*. [87] brings the third type of biclustering problem to a higher level in the following sense. In the paper, they are interested in identifying not only the relevant experimental conditions for which the relation between genes of a potential group exists but also the significant attributes associated with the genes and the conditions that are responsible for the generation of such patterns. The method incorporates additional information of the attributes (for example, for a gene, the attributes could be functional role, cellular location or the transcriptional factor (TF) binding sites in gene's promoter region, and for a experimental condition, they could be tumor type, or gene knock out information) together with the gene expression data in a probabilistic framework – the probabilistic relational models (PRMs) [36] in particular, which is an extension of Bayesian networks – and uses (structural) EM [34] for

the inference for the parameters of the probabilistic models. In [86], the capability of this framework is illustrated in unveiling the regulation programs of the genes from gene expression data.

Note that the target types of biclusters of various approaches differ not only in their structures, but also in the numeric patterns. Madeira and Oliveira (2004) [70] gives a comprehensive survey on the existing biclustering algorithms. According to their criterion, the numerical patterns of biclusters can be divided into following categories: constant biclusters, biclusters of constant rows or constant columns, biclusters of additive coherent values, biclusters of multiplicative values, biclusters of coherent evolutions, biclusters of coherent sign changes. For example, our method aims to find biclusters of constant rows or columns, the method of Cheng and Church (2000) [21] identifies biclusters of coherent additive values, and SAMBA searches for biclusters of sign changes.

In what follows, we discuss the gene shaving algorithm [47] (a representative algorithm for the second type of biclustering algorithm judging from the structure), the biclustering algorithm of Cheng and Church (2000) [21] (the pioneering algorithm of the third type), and the PRMs for microarray data [87, 86] in more detail.

3.6.1 Gene shaving

As we mentioned previously, gene shaving is an algorithm that tries to find a small subset of genes that exhibit the largest variations across the experimental conditions. The intuition is that these genes may help in explaining the classification of the experiments. To search for a gene cluster, the algorithm performs a PCA on the p-dimensional space, where p stands for the number of experiments in the data set. The largest principal component, called the eigengene, points to the direction where the cloud of data points (representing the genes) expands with largest variation. Then the correlation between every gene with this eigengene is calculated. Genes with the smallest (absolute value of) correlation are "shaved off" (discarded). The proportion of the genes to be shaved off is typically 10%. This procedure is re-performed on the "shaved" data set until there is only one gene left in the data set. The remaining data set at every step is treated as a candidate gene cluster. To determine the output cluster from these candidate clusters, the ratio of the within-variance versus the between-variance of every candidate cluster is calculated. The within-variance of a cluster calculates first the sample variance of the genes in the cluster, and then the sample variance is averaged over all the experiments. It measures the similarity of the genes in the cluster. The between-variance of a cluster is defined as the variance of the mean expression profile of the genes in the cluster. A good cluster should have a large between-variance, meaning that the included genes are expressional active, and a small within-variance, indicating a tight cluster. The ratio of a candidate

cluster is compared with the average ratio of the clusters (which contain the same number of genes) obtained from randomized data sets. The candidate cluster with the largest difference in the ratio is selected as the output cluster. Then, the expression profiles in the whole data set is orthogonalized with the mean of the expression profiles in the output cluster so as to encourage the discovery of an uncorrelated second cluster. The procedure restarts searching for the next cluster.

Hastie *et al*. [47] also shows that the algorithm can also be extended to a supervised form for the discovery of genes related with particular sample classification. Results in the paper illustrate a successful application of the algorithm on predicting patient survival.

3.6.2 Cheng and Church's approach

The biclustering algorithm of Cheng and Church (2000) [21] is a greedy search algorithm to find a set of genes that behave consistently under a subset of conditions.

The consistency of the bicluster is measured by the score of the mean squared residue $H(I, J)$. The residue of element a_{ij} in the bicluster indicated by the subsets of I and J is

$$a_{ij} - aiJ - a_{Ij} + aIJ \tag{3.1}$$

where aiJ is the mean of the i^{th} row in the bicluster, a_{Ij} is the mean of j^{th} column in the bicluster, and a_{IJ} is that of all the elements in the bicluster. In order to encourage the gene expression profiles in a bicluster to have fluctuation across the experimental conditions, (because expression profiles with little fluctuation—called trivial biclusters—are not interesting for biologists), they devised another score of row variance as an accompanying score to reject trivial biclusters

$$V(I, J) = \frac{1}{|J|} \sum_{j \in J} (a_{ij} - aIj)^2. \tag{3.2}$$

The goal of the algorithm is thus to find a bicluster with the largest area (i.e. $|I| \cdot |J|$) whose mean squared residue score satisfies $H(I, J) <= \delta$, where δ is a threshold input by the user. The bicluster is rejected if the row variance score $V(I, J)$ is lower than another threshold. Because the optimization problem is an NP hard one, they opted for a greedy search algorithm that starts from the original microarray data set, and delete a few rows or a few columns at a time in the direction to decrease $H(I, J)$. Once the condition $H(I, J) <= \delta$ is reached, there is a chance that the resulting bicluster is not maximal. Thus, another step to add some rows or columns without increasing the score is performed.

The algorithm above aims to find one bicluster at a time for a microarray data set. In order to discover multiple biclusters, the authors mask the found

bicluster by replacing the matrix entries in the bicluster by random values. However, the masks are not used for the rows/columns addition procedure.

3.6.3 Probabilistic relational models for microarray data

In the preparation to address the problem in the language of PRMs [36], Segal *et al*. [87] first define the relational schema as follows. The objects – genes, experiments, and gene expressions – are associated with their corresponding attributes. Among other attributes such as the TF binding sites for a gene and the knock-out information for an experiment, the class of a gene is one of the attributes of the gene; similarly, the class of an experiment is an attribute for the experiment; and for the expression of a specific gene in a particular array, the expression level is one of its attributes. The values of some of these attributes are given by the data, for instance, the TF binding sites, the knock-out information, and the expression level. However, the values of the other attributes are unknown. The particular examples of interest in this case are the class attributes of the genes and the experiments. The task now is not only to infer the unknown value of the attributes (such as the class attributes so as to obtain the biclusters) but also to find the relationships between all the attributes presented in the problem.

A PRM itself can be viewed as the bigger frame defining the relationships between classes of different objects. In our case, the frame is that gene expressions are influenced by both the genes and the experiments. However, when it comes to a particular case, where the given data includes the microarray data and accompanied information of the genes and the experiments, the PRM has to generate a more detailed probability model, namely a Bayesian network, for the specific problem. First, to put them in a probabilistic language, all the attributes are described as random variables. A Bayesian network is set up to address the dependency structures between the random variables (i.e., the attributes). A Bayesian network [49] is a graphical model where nodes (each representing a random variable) are connected with each other by directed edges. The direction of an edge between two nodes (pointing from a parent node to a child node) indicates the influence of one random variable (the parent node) to the other (the child node). The influence is quantified by the probabilistic distribution of the child node conditioned on the value of the parent node – the conditional probability distribution (CPD). A node is conditionally independent of all the other nodes given the value of its parents. For the microarray data, one can suppose that the attribute of expression level is always placed at the bottom of the Bayesian network structure; i.e., it is always a child node but never a parent node.

To learn the relationships between different attributes now means to decide the structure of the Bayesian network (i.e., to draw the edges) and the CPDs between the nodes. In addition, the learning task also includes the estimation of the unknown data (or called missing data) of some of the attributes. The

learning procedure is carried out by the Structural EM algorithm [34]. Briefly speaking, it iteratively executes a structural learning step and an EM step till convergence. In the structural learning step, a search algorithm is performed based on the current estimate of network parameters (CPDs) and missing data to find the structure that maximizes a Bayesian information score [84]. The EM step uses the learned structure to estimate its CPDs (the M step) and the missing data (the E step).

The efficiency of the method is illustrated in [87] on two yeast data sets as well as synthetic data sets. In [86], it is shown that the method can be tailored for unveiling the regulatory program of the genes.

3.7 Assessing cluster quality

Clustering will produce different results. Even random data often produce clusters depending on the specific choice of preprocessing, algorithm, and distance measure. Therefore, validation of the relevance of the cluster results is of utmost importance. Validation can be either statistical or biological. Statistical cluster validation can be done by assessing cluster coherence, by examining the predictive power of the clusters, or by testing the robustness of a cluster result against the addition of noise.

Alternatively, the relevance of a cluster result can be assessed by a biological validation. Of course it is hard, not to say impossible, to select the best cluster output, since "the biologically best" solution will be known only if the biological system studied is completely characterized. Although some biological systems have been described extensively, no such completely characterized benchmark system is now available. A common method to biologically validate cluster outputs is to search for enrichment of functional categories within a cluster. Detection of regulatory motifs (see [97]) is also an appropriate biological validation of the cluster results. Some of the recent methodologies described in literature to validate cluster results will be highlighted in the following.

1. *Testing cluster coherence:* Based on biological intuition, a cluster result can be considered reliable if the within-cluster distance is small (i.e., all genes retained are tightly coexpressed) and the cluster has an average profile well delineated from the remainder of the data set (maximal inter-cluster distance). Such criteria can be formalized in several ways, such as the sum-of-squares criterion of k-means [102], silhouette coefficients [57], or Dunn's validity index [5]. These can be used as stand alone statistics to mutually compare cluster results. They can also be used as an inherent part of cluster algorithms, if their value is optimized during the clustering process.

2. *Figure of Merit:* FOM [117] is a simple quantitative data-driven methodology that allows comparisons between outputs of different clustering algorithms. The methodology is related to the jackknife and leave-one-out cross-validation. The method goes as follows. The clustering algorithm (for the genes) is applied to all experimental conditions (the data variables) except for one left-out condition. If the algorithm performs well, we expect that if we look at the genes from a given cluster, their values for the left-out condition will be highly coherent. Therefore, we compute the FOM for a clustering result by summing, for the left-out condition, the squares of the deviations of each gene relative to the mean of the genes in its cluster for this condition. The FOM measures the within-cluster similarity of the expression values of the removed experiment and therefore reflects the predictive power of the clustering. It is expected that removing one experiment from the data should not interfere with the cluster output if the output is robust. For cluster validation, each condition is subsequently used as a validation condition, and the aggregate FOM over all conditions is used to compare cluster algorithms.

3. *Sensitivity analysis:* Gene expression levels are the superposition of real biological signals and experimental errors. A way to assign confidence to a cluster membership of a gene consists in creating new in silico replicas of the microarray data by adding to the original data a small amount of artificial noise (similar to the experimental noise in the data) and clustering the data of those replicas. If the biological signal is stronger than the experimental noise in the measurements of a particular gene, adding small artificial variations (in the range of the experimental noise) to the expression profile of this gene will not drastically influence its overall profile and therefore will not affect its cluster membership. In this case, the cluster membership of that particular gene is robust with respect to sensitivity analysis, and a reliable confidence can be assigned to the clustering result of that gene. However, for genes with low signal-to-noise ratios, the outcome of the clustering result will be more sensitive to adding artificial noise. Through some robustness statistic [16], sensitivity analysis lets us detect which clusters are robust within the range of experimental noise and therefore trustworthy for further analysis.

   The main issue in this method is to choose the noise level for sensitivity analysis. Bittner et al. [16] perturb the data by adding random Gaussian noise with zero mean and a standard deviation that is estimated as the median standard deviation for the log-ratios for all genes across the experiments. This implicitly assumes that ratios are unbiased estimators of relative expression, yet reality shows often otherwise.

   The bootstrap analysis methods described by Kerr and Churchill [59] to identify statistically significant expressed genes or to assess the reliability of a clustering result offers a more statistically founded basis for sensitivity analysis and overcomes some of the problems of the method

described by Bittner et al. [16]. Bootstrap analysis uses the residual values of a linear analysis of variance (ANOVA) model as an estimate of the measurement error. By using an ANOVA model, nonconsistent measurement errors can be separated from variations caused by alterations in relative expression or by consistent variations in the data set. These errors are assumed to be independent with mean zero and constant variance σ^2 but no explicit assumption on their distribution is made. The residuals are subsequently used to generate new replicates of the data set by bootstrapping (adding residual noise to estimated values).

4. *Use of different algorithms:* Just as clustering results are sensitive to adding noise, they are sensitive to the choice of clustering algorithm and to the specific parameter settings of a particular algorithm. Many clustering algorithms are available, each of them with different underlying statistics and inherent assumptions about the data. The best way to infer biological knowledge from a clustering experiment is to use different algorithms with different parameter settings. Clusters detected by most algorithms will reflect the pronounced signals in the data set. Again statistics similar to that of Bittner et al. [16] are used to perform these comparisons.

   Biologists tend to prefer algorithms with a deterministic output, since this gives the illusion that what they find is "right". However, nondeterministic algorithms offer an advantage for cluster validation, since their use implicitly includes a form of sensitivity analysis.

5. *Enrichment of functional categories:* One way to biologically validate results from clustering algorithms is to compare the gene clusters with existing functional classification schemes. In such schemes, genes are allocated to one or more functional categories [44, 97] representing their biochemical properties, biological roles, and so on. Finding clusters that have been significantly enriched for genes with similar function is proof that a specific clustering technique produces biologically relevant results.

   Using the cumulative hypergeometric probability distribution, we can measure the degree of enrichment by calculating the probability or P-value of finding by chance at least k genes in this specific cluster of n genes from this specific functional category that contains f genes out of the whole g annotated genes

$$P = 1 - \sum_{i=0}^{k-1} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}} = \sum_{i=k}^{\min(n,f)} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}}. \tag{3.3}$$

   These P-values can be calculated for each functional category in each cluster. Note that these P-values must be corrected for multiple testing according to the number of functional categories.

3.8 Conclusion

In this chapter, we provide an overview of clustering algorithms on microarray data from three points of view. By looking at the advantages and disadvantages of classical clustering algorithms, we conclude that tailored clustering algorithms are necessary for better analysis of microarray data. We reviewed a couple of model-based methods where traditional probabilistic models are applied. This review provides a basis for understanding our Bayesian models for microarray data. After that, we briefly discussed the current status of research in biclustering, which situates our research properly. Finally, we give a guideline for evaluating the quality of a cluster.

Chapter 4

Gibbs sampling on Bayesian hierarchical models the biclustering

This chapter explains in detail the main framework of our biclustering strategy. We start with an introduction of Gibbs sampling. Next, we introduce the concept of Bayesian hierarchical models, followed by a detailed description of how it is applied to the modeling for the biclustering problem. Then, we combine the two concepts, and elaborate the Gibbs sampling procedure on the Bayesian hierarchical model for biclustering—the framework of our algorithm.

4.1 Introduction

Most clustering techniques provide a global view of the structure of microarray data, revealing genes that share similar expression profiles, 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 (when clustering patients, for example) about what the classifications of the patients are. However, this type of analysis does not provide insights to those specific questions that biologists ask.

These specific questions, nowadays, with the developments in systems biology, are often derived from information from other sources. The increasingly overwhelming amount of data from heterogeneous sources, besides mRNA level data (e.g. DNA sequence information, protein structural information) calls

out for analysis strategies that can integrate these biological insights from different aspects to study the concerted gene interactions. To this end, clustering algorithms that are compatible for such integration are favorable. For example, suppose that a few genes are identified to share the same *cis*-regulatory elements in their promoter regions by motif finding algorithms. However, because of the large noise in DNA sequence data, this discovery does not guarantee the coregulation of the genes. Evidence from gene expression data can not only support (or reject) such discoveries in the DNA sequence data, but also provide new clues of other genes that could be in the same regulatory transcriptional module (i.e., genes that are coregulated under a certain circumstance). To be more concrete, a desirable biclustering algorithm should allow input of information from a set of genes that share same motif combination in their promoter region, identify a transcriptional module for these genes, and leave out those input genes whose expression profile do not match the found transcriptional module.

Clustering algorithms based on Bayesian probabilistic models are promising as candidates meeting these requirements. Most importantly, Bayesian inference allows introducing prior knowledge, which captures the specific questions of biologists. The incorporation of prior also provides a reliable and interpretable platform for the integration of the analysis of gene expression data with other sources of biological information [88, 35]. In addition, Bayesian probabilistic models retain all the property of general probabilistic models to reveal the fundamental structure that biologists seek in microarray data [87, 86] (see Section 3.6).

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 biclustering is to group experiments (e.g., patients) under each of which a subset of genes have almost the same expression values, see Figure 1.6. 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 hierarchical models to describe both of the problems. We choose to use Gibbs sampling to learn such Bayesian models.

The Gibbs sampling strategy was first introduced to the field of bioinformatics for its applications to the motif finding problem [64] in DNA sequence analysis, and has become the method-of-choice for this problem [99]. Our idea to apply Bayesian strategy to the biclustering problem of microarray data was inspired by its success in motif finding.

Gibbs sampling is a Markov chain Monte Carlo technique to draw samples from a joint distribution, when the conditional distributions of all the target random variables are available. It has become a popular alternative to the expectation-maximization (EM) algorithm for solving incomplete-data prob-

lems. In a typical incomplete data problem, the observed data is described by a set of random variables (i.e., the observed variables). For each data point (a vector whose length equals the number of random variables in the set), the particular distribution applied to the observed random variables depends on the corresponding values of a set of hidden variables. However the values of the hidden variables are not observed, and they form the missing data of the problem*. The task is to estimate the missing data for the hidden variables as well as the parameters of the involved distributions in the model.

The EM algorithm iterates between an expectation step for estimating (the sufficient statistics of) the hidden variables, and a maximization step that selects the model parameters that maximize the likelihood based on the complete data (which includes both the observed data and the missing data). Gibbs sampling, on the other hand, treats both the hidden variables and the model parameters as random variables, and aims at estimating their joint distribution. Once the joint distribution is obtained, posterior mean estimate (PME) estimates are often used for the hidden variables and the model parameters.

So far, we have only addressed the case when the relation between the hidden variables and the observed variables is known (i.e., known structure). However, when such structure is unknown, Gibbs sampling becomes an alternative to structural EM. Structural EM resembles EM by adding a step for structure optimization. Gibbs sampling, in this case, describes the structure also by random variables and applies the same strategy.

As explained later in this chapter, the biclustering problem belongs to the latter type of problem (with incomplete data and unknown model structure). Gibbs sampling is often found to be time consuming and computationally intensive for structural learning. However, for the particular class of structure that addresses the biclustering algorithm (the same general structure is applied to both the biclustering of genes and the biclustering of experiments), we found that Gibbs sampling to be efficient, and we favor the Gibbs sampling strategy over structural EM.

This chapter is organized as follows. First, in Section 4.2, we review some essential concepts for Gibbs sampling, especially on how and why it works. Then in Section 4.3, we discuss Bayesian model that we use for the biclusering problem, as well as why we prefer Gibbs sampling to structural EM for solving the problem. Finally, we detail the Gibbs sampling procedure in Section 4.4.

*Note that we distinguish between the phrases "missing data" and "missing values". Missing data refers to the data of the hidden variables, while missing values refer to data points in a microarray data matrix whose values are marked as unavailable (see Section 2.5). Bayesian inference can be applied to deal with both missing values and missing data, however, the detailed procedure can be quite different between the cases.

4.2 Gibbs sampling

Gibbs sampling is a technique to draw samples from a joint distribution based on the full conditional distributions of all the associated random variables. Though the idea roots back to the work of Hasting (1970) [48], whose focus was on its Markov chain Monte Carlo (MCMC) nature, the Gibbs sampler was first formally introduced by Geman and Geman (1984) [40] 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) [38]). 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 [19]. We will talk about the difference between Bayesian statistics and classical statistics in Section 4.3. In this section, we discuss the working mechanism of Gibbs sampling.

Gibbs sampling allows statisticians to avoid the tedious and sometimes non-trivial mathematical calculations of integrals for obtaining the joint distribution, by sampling directly from the full conditional distributions[†]. Suppose that we want to draw samples for the set of random variables $X_1, X_2, \ldots, X_m$, but that the marginal distributions (and thus their joint distribution) are (is) too complex to directly sample from. Suppose also that the full conditional distributions $p(X_i \mid X_j;\ j \neq i)$ (for $i = 1, \ldots, n$) can easily be sampled from. Starting from initial values $x_1^{(0)}, x_2^{(0)}, \ldots, x_m^{(0)}$, the Gibbs sampler draws samples for the random variables in the following manner,

$$
\begin{aligned}
x_1^{(t+1)} &\sim p(X_1 \mid X_2 = x_2^{(t)}, \ldots, X_m = x_m^{(t)}) \\
x_2^{(t+1)} &\sim p(X_2 \mid X_1 = x_1^{(t+1)}, X_3 = x_3^{(t)}, \ldots, X_m = x_m^{(t)}) \\
&\vdots \\
x_i^{(t+1)} &\sim p(X_i \mid X_1 = x_1^{(t+1)}, \ldots, X_{i-1} = x_{i-1}^{(t+1)}, X_{i+1} = x_{i+1}^{(t)}, \ldots, X_m = x_m^{(t)}) \\
&\vdots \\
x_k^{(t+1)} &\sim p(X_m \mid X_1 = x_1^{(t+1)}, \ldots, X_{m-1} = x_{m-1}^{(t+1)}),
\end{aligned}
\tag{4.1}
$$

where t indexes the iterations.

Geman and Geman (1984) [40] shows that as $t \to \infty$, the distribution $p(X_1^{(t)}, \ldots, X_m^{(t)})$ converges to $p(X_1, \ldots, X_m)$. Equivalently, as $t \to \infty$, the distribution $p(X_i^{(t)})$ converges to $p(X_i)$ (for $i = 1, \ldots, m$).

[†]Because the same mechanism applies to both discrete models and continuous models, we use the terms "distribution" and "density" interchangeably, and we use $p(\cdot)$ to denote both in this section.

4.2.1 The Markov chain property

The convergence of samples drawn by the Gibbs sampler relies on the fact that these samples form aMarkov chain. To be more explicit,

$$\left((X_1^{(1)}, \ldots, X_m^{(1)}), \ldots, (X_1^{(t)}, \ldots, X_m^{(t)})\right)$$

as well as

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

are Markov chains, where $(X_1^{(t)}, \ldots, X_m^{(t)})$ and $X_i^{(t)}$ are called the states of $(X_1, \ldots, X_m)$ and X_i respectively. The basic property of a Markov chain, take that of X_i for example, is

$$p(X_i^{(t+1)} \mid X_i^{(t)}, \ldots, X_i^{(0)}) = p(X_i^{(t+1)} \mid X_i^{(t)}), \tag{4.2}$$

which means that the future state of the random variable depends only on its current state but not on its past states. In other words, the current state summarizes the past.

Now suppose that we only consider the case of X_i, we use $\pi_u(v)$ to denote the probability that X_i is in state b at time point $t+1$. Writing

$$\pi_b(t+1) = p(X_i^{(t+1)} = b) \tag{4.3}$$

$$\pi_a(t) = p(X_i^{(t)} = a) \tag{4.4}$$

$$\text{and} \quad p(a \to b) = p(X_i^{(t+1)} = b \mid X_i^{(t)} = a), \tag{4.5}$$

we have

$$\pi_b(t+1) = \sum_a p(a \to b)\pi_a(t). \tag{4.6}$$

$p(a \to b)$ is called the transition probability of going from state a to b (for random variable X_i). When X_i is a discrete random variable, the probability transition matrix $\boldsymbol{P}$ is obtained by listing all the possible states for X_i along the rows and the columns, and filling the stochastic matrix with all the transition probabilities. Note that this implies that each row of $\boldsymbol{P}$ sums to 1. When X_i is a continuous variable, the transition matrix can be seen to have infinite dimensionality, and is represented by a density function.

Thus to generalize Equation 4.6, we have

$$\pi(t+1) = \boldsymbol{P}\,\pi(t). \tag{4.7}$$

A Markov chain will reach a unique stationary distribution π^*, such that,

$$\pi^* = \boldsymbol{P}\,\pi^*, \tag{4.8}$$

if

$$p(j \to k) \cdot \pi_j^* = p(k \to j) \cdot \pi_k^*. \tag{4.9}$$

This sufficient (but not necessary) condition is called detailed balance. When this condition is met, samples of the concerned variables obtained by Gibbs sampling are guaranteed to converge to the stationary distribution π^*, independent of the initial distribution of the states $\pi(0)$.

Casella and George (1992) [19] gives a simple yet intuitive proof that the stationary distributions of the Markov chains generated by Gibbs sampling are the joint distribution $p(X_1, \ldots, X_m)$ 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. We demonstrate in Figure 4.1 the convergence of the Gibbs sampling procedure on a simple two dimensional Gaussian distribution,

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \mathbf{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}\right). \tag{4.10}$$

As the figure illustrates, the samples collected by the Gibbs sampler converge to the target distribution, (instead of a single point).

4.2.2 The Monte Carlo property

We leave the discussion of convergence diagnosis of Gibbs sampling to the next section, and for the moment we assume that we have decided a time point by which we consider the procedure to have converged. 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 phase performed before the convergence is reached is often referred to as the "burn-in phase", and the phase during which samples are collected will be called the "sampling phase" hereafter. The samples collected in the sampling phase 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

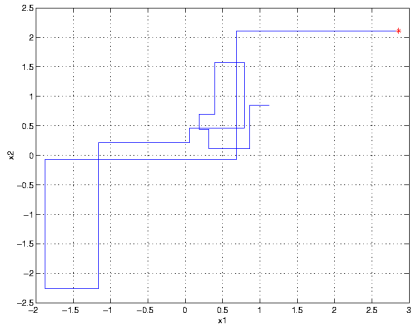
$$\mathrm{E}_{p(X_i)}[f(X_i)] = \int f(X_i) \cdot p(X_i) \mathrm{d}X_i \approx \frac{1}{T} \sum_{t=1}^{T} f(x_i^{(t)}), \tag{4.11}$$

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

$$\mathrm{E}_{p(X_i)}[X_i] = \int X_i \cdot p(X_i) \mathrm{d}X_i \approx \frac{1}{T} \sum_{t=1}^{T} X_i^{(t)}. \tag{4.12}$$

However, as illustrated by Gelfand and Smith (1990) [38] (using the Rao-Blackwell theorem), a more accurate estimate of the expected value of X_i is provided by

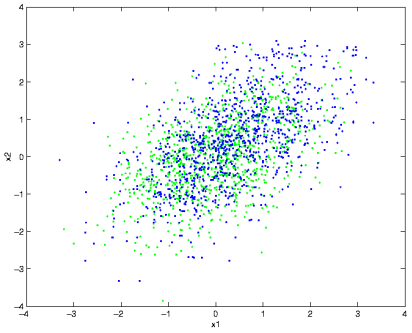
$$\mathrm{E}_{p(X_i)}[X_i] = \frac{1}{T} \sum_{t=1}^{T} \mathrm{E}_{p(X_i \mid X_j;\, j \neq i)}[X_i]. \tag{4.13}$$



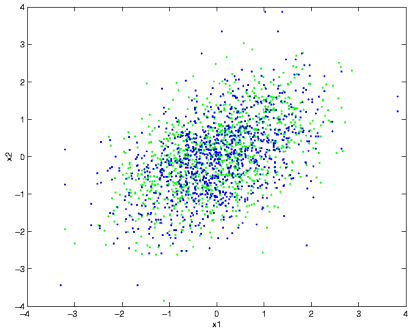
(a) Trace plot of the first 20 samples drawn by the Gibbs sampler. The red dot is the starting point.

	μ	Σ
true	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$
burn-in	$\begin{pmatrix} 0.3634 \\ 0.4190 \end{pmatrix}$	$\begin{pmatrix} 1.12433 & 0.7443 \\ 0.7443 & 1.3724 \end{pmatrix}$
converged	$\begin{pmatrix} 0.0187 \\ -0.0443 \end{pmatrix}$	$\begin{pmatrix} 1.0282 & 0.5052 \\ 0.5052 & 1.0621 \end{pmatrix}$

(b) Parameters of the true distribution, and sample statistics obtained during the burn-in procedure and on the converged samples.



(c) The green dots are samples obtained for the true distribution, using generic functions in MATLAB. The blue dots are samples obtained during the burn-in procedure of Gibbs sampling.



(d) The green dots are samples obtained for the true distribution, using generic functions in MATLAB. The blue dots are samples obtained after convergence of the Gibbs sampling procedure is reached.

Figure 4.1: Example of Gibbs sampling on a two-dimensional Gaussian distribution. During the burn-in phase, the samples are coming into form so that they begin to overlap with the true distribution. The samples collected after convergence is reached represent the true distribution well.

Similarly, the posterior distribution itself can be approximated by

$$\mathrm{E}[p(X_i)] = \frac{1}{T}\sum_{t=1}^{T} p(X_i \mid X_j;\ j \neq i). \tag{4.14}$$

With more generality, a better alternative for Equation 4.11 is

$$\mathrm{E}_{p(X_i)}[f(X_i)] = \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}_{p(X_i \mid X_j;\, j\neq i)}[f(X_i)]. \tag{4.15}$$

In the above three equations, $p(X_i \mid X_j;\ j \neq i)$ denotes the full conditional distribution of X_i. The estimators obtained by Monte Carlo integration are unbiased *maximum a priori* (PME) estimators.

4.2.3 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 case to case. For a well-mixed Markov chain—whose samples cover most of the region of the random variable space—the convergence can be reached within 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 [39] 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 the likelihood. Various formal procedures has also been proposed. The method of Raftery and Lewis (1992) [79] aims to bound the summary statistics of the target variables within a certain precision. Their approach calculates the number of burn-in iterations as well as the total number of iterations that are needed to reach a specific quantile with a desired accuracy with a pre-specified probability. Geweke's diagnostic [43] checks if a Markov chain approaches a stationary distribution by examining if the standardized difference between the mean of the samples taken in the beginning of the chain and the mean of the samples taken from the end of the chain reaches a normal distribution. Both of the methods mentioned above perform diagnostics on a single Markov chain. Gelman and Rubin (1992) [39] proposed an approach based on parallel Markov chains. The method is based

on comparison of the within chain variance with the between chain variance for each target variable. A good review on various convergence diagnostics is provided by Cowles and Carlin (1996) [25].

In the rest of this section, we focus on one of the important issues for the convergence of a Markov chain—the autocorrelation—which is used as our formal criterion to determine if the Gibbs sampling procedure for biclustering has converged.

One of the reasons that a Markov chain generated by the Gibbs sampler has a slow convergence is that the samples at successive iterations are not independent. This dependency implies that the variance of the model obtained by averaging the parameters may be much higher (i.e., the accuracy of the model is lower) 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 vector of parameters obtained at each iteration and by

$$\bar{\boldsymbol{\omega}} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\omega}^{(t)} \tag{4.16}$$

the average set of parameters, the autocorrelation function ρ for a lag of h can be estimated as

$$\hat{\rho}_h = \frac{\mathrm{Cov}(\boldsymbol{\omega}^{(t)}, \boldsymbol{\omega}^{(t+h)})}{\mathrm{Var}(\boldsymbol{\omega}^{(t)})} = \frac{\sum_{t=1}^{T-h}(\boldsymbol{\omega}^{(t)} - \bar{\boldsymbol{\omega}})(\boldsymbol{\omega}^{(t+h)} - \bar{\boldsymbol{\omega}})}{\sum_{t=1}^{T-h}(\boldsymbol{\omega}^{(t)} - \bar{\boldsymbol{\omega}})^2}. \tag{4.17}$$

In the frequent case where the autocorrelation function can be described as an autoregressive process, the autocorrelation time ι

$$\iota = \sum_{h=1}^{\infty}\hat{\rho}_h \tag{4.18}$$

can be simplified to

$$\iota = (1 + \hat{\rho}_1)/(1 - \hat{\rho}_1). \tag{4.19}$$

A large autocorrelation time indicates that the chain is poorly mixing, and the convergence takes a long period.

One way to reduce the autocorrelation is to use the thinning of the Markov chain. Thinning with a factor l means that each l^{th} element in the chain will be used for the posterior summary statistics (see Equation 4.15). Another computational advantage of using the thinning procedure is that it saves the memory complexity of the Gibbs sampling procedure (although it does not reduce the computational complexity in any way).

We will show an example of diagnosing the autocorrelations of samples produced for our biclustering algorithm in Section 5.6.2.

4.3 Bayesian hierarchical model for biclustering

4.3.1 Bayesian hierarchical models

Suppose that the data for all the concerned random events are collected, and that the probabilistic model to describe the data is known (or, more realistically, when the model is determined), and that we would like to estimate the parameters of the model. From a frequentist point of view, the true parameters are reflected in the data. However, for a Bayesian, the problem is "starting from my current knowledge, what do I learn from the collected information (i.e., data) about the probability that these random events could happen?" or "how do these data change my point of view?". This updating procedure of belief is the essential question that Bayesian inference addresses. The mathematical form of Bayesian inference provides a natural form for translating the problem to mathematical language,

$$p(\boldsymbol{\Theta} \mid \mathcal{D}, \xi) = \frac{p(\boldsymbol{\Theta} \mid \xi) \cdot p(\mathcal{D} \mid \boldsymbol{\Theta}, \xi)}{p(\mathcal{D} \mid \xi)}. \tag{4.20}$$

In Equation 4.20, $\mathcal{D}$ represents the data for the random events (i.e., random variables of interest) $\mathbf{X}_m = \{X_1, X_2, \ldots, X_m\}$ of interest (where m is the number of random variables), i.e., $\mathcal{D} = \left\{\mathbf{X}_m = \mathbf{x}_m[1], \ldots, X_n = \mathbf{x}_m[n]\right\}$ (where n is the total number of instances in the data); and $\boldsymbol{\Theta}$ stands for the parameters of the distribution of $\mathbf{X}_m$ (i.e., model parameters, which quantify our belief). Equation 4.20 tells us that

$$\text{Posterior probability} = \frac{\text{Prior probability} \times \text{Likelihood}}{\text{Evidence}},$$

see Figure 4.2 for a further illustration. The first term in the numerator of Equation 4.20, $p(\boldsymbol{\Theta} \mid \xi)$, is called the prior distribution of $\boldsymbol{\Theta}$, where ξ parameterizes this prior distribution (ξ is therefore called the hyperparameter of a Bayesian model). The second term in the numerator of Equation 4.20, $p(\mathcal{D} \mid \boldsymbol{\Theta}, \xi)$, is called the likelihood of $\boldsymbol{\Theta}$, which is computed based on the probability distribution of $\mathbf{X}_m$, which is parameterized by $\boldsymbol{\Theta}$, which is referred to as the parameters of the Bayesian model. The likelihood funciton is the probability that we observe the data given the model parameters. Finally, the denominator, $p(\mathcal{D} \mid \xi)$, is called the evidence. It is a normalization term to ensure that the result of the calculation is still in the form of a probability. The evidence can be obtained by

$$p(\mathcal{D} \mid \xi) = \int_{\boldsymbol{\Theta}} p(\mathcal{D} \mid \boldsymbol{\Theta}, \xi) \cdot p(\boldsymbol{\Theta} \mid \xi) \mathrm{d}\boldsymbol{\Theta}. \tag{4.21}$$

The left-hand part of the Equation 4.20, $p(\boldsymbol{\Theta} \mid \mathcal{D}, \xi)$, is called the posterior distribution of $\boldsymbol{\Theta}$. It represents of our belief after our knowledge is updated by the presence of the data. In this way, Bayesian inference resembles the human learning process.

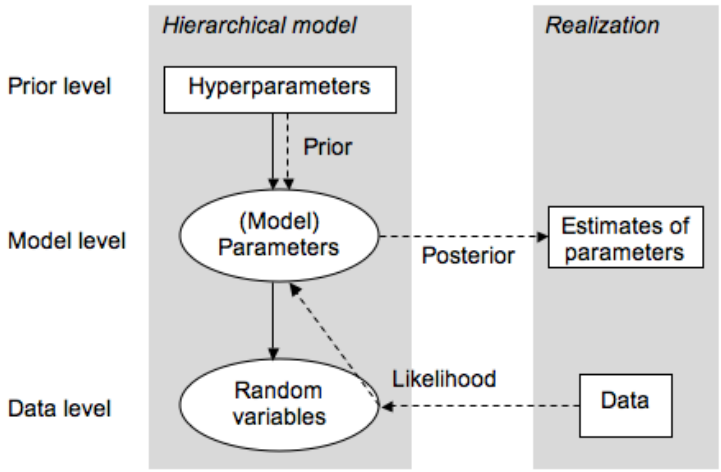


Figure 4.2: Bayesian hierarchical model and Bayesian inference. The solid arrows show the direction of the modeling in a Bayesian hierarchical model (i.e., the distribution of the model parameters is parameterized by the hyperparameters, and the distribution of the random variables of interest is parameterized by the model parameters). Bayesian inference concerns the problem of how our belief of the model parameters changes from our prior belief, given the realization of the random variables (i.e., the data). The dashed arrows show the information flow in Bayesian inference. Note that the hyperparameters are seen as fixed values, and are not inferred.

We have explained above the Bayesian model from the inference point of view. From the modeling point of view, Bayesian hierarchical model comprises of three levels, see Figure 4.2. At the bottom level of the hierarchy are the random variables $\mathbf{X}_m$ to which the data $\mathcal{D}$ is presented. This level is thus referred to as data level. In the middle of the hierarchy are the model parameters for $\mathbf{X}_m$—$\mathbf{\Theta}$. Therefore, this level is called the model level. Finally on the top—the prior level, we have the prior of the model ξ.

When both the data and the prior are given, the inference of the model parameters $\mathbf{\Theta}$ is straightforward by applying Equation 4.20. However, for our biclustering problem, the microarray data that we obtained only composes the "observed data" of the problem, which are represented by the observed random variables in the model. But the partition of the observed data into the bicluster and the background requires some hidden variables, whose values are unknown. Therefore, the biclustering problem is an incomplete-data problem. Gibbs sampling is one of the techniques for solving the inference of models for this type of problems.

4.3.2 Biclustering: an incomplete-data problem

Graphical models, especially directed acyclic graphical (DAG) models (i.e., for any vertex v in the graph, there is no path that starts and ends at v) provide a good representation tool to visualize the relations between the missing data and the observed data in the biclustering problem. The theory of graphical models combines probability theory and graph theory, which have become important tools for machine learning. It is an active research area. A good tutorial for this field is provided by the book edited by Jordan (1999) [55]. In what follows, we briefly introduce the terminology used for graphical models. We then go directly into the discussion of applying graphical models to depict the data level of Bayesian hierarchical model of biclustering.

Graphical models

A probabilistic graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ contains two components, one for the set of vertices $\mathcal{V}$ in the graph, and the other for the set of edges $\mathcal{E}$ in the graph. The vertices of a graph are also called nodes. They represent the random variables under consideration. An edge in the graph refers to the relation between two random variables by means of a conditional distribution. An edge points from a parent variable to its child variable.

In the example of Figure 4.3(A), Y is the parent of X, the edge in between refers to $p(X \mid Y)$. In this way, graphical models also depict independence between the variables. When there is no path of edges that connects two nodes, it implies that the two nodes are totally independent. In the example of Figure 4.3(B), X is independent of Y, mathematically denoted as $X \perp\!\!\!\perp Y$, which means

$$p(X, Y) = p(X) \cdot p(Y). \tag{4.22}$$

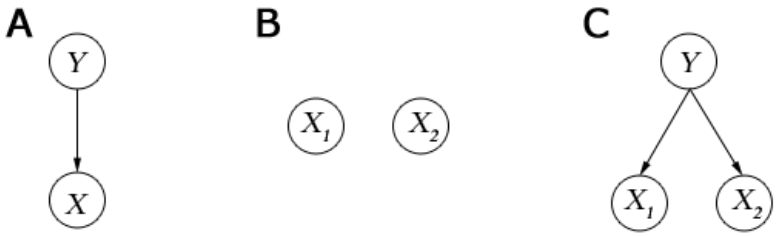


Figure 4.3: Examples of simple graphical model structures.

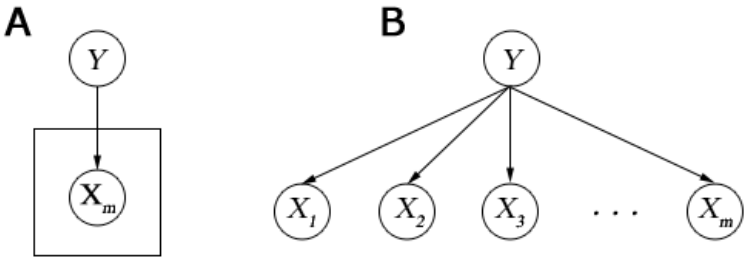


Figure 4.4: Example of plate: A uses a plate to simplify the structure presented in B. Both of the graphs imply that the m variables in $\mathbf{X}_m$ are conditionally independent and identically distributed given Y. This structure is known as a naive Bayes structure.

A more useful and commonly seen independence relation is that of the conditional independence. This is when two nodes share the same parent or ancestors. In Figure 4.3(C), X_1 and X_2 are conditionally independent of each other given Y, written $X_1 \perp\!\!\!\perp X_2 \mid Y$, i.e.

$$p(X_1, X_2 \mid Y) = p(X_1 \mid Y) \cdot p(X_2 \mid Y). \tag{4.23}$$

This means that all the dependency between X_1 and X_2 is addressed by random variable Y.

When a graphical model has a replicated structure, a plate can be used to simplify the graph, See Figure 4.4.

Graphical models for clustering using mixture models

We start our discussion of the application of graphical models by considering a simpler case—i.e., clustering gene expression profiles by fitting the data to a mixture model. As we explained in Section 3.5, the task of clustering here is to associate each gene expression profile to the best fitting component in the mixture. Thus the data instances refer to the gene expression profiles.

We illustrate the data level of a Bayesian model for the clustering problem in

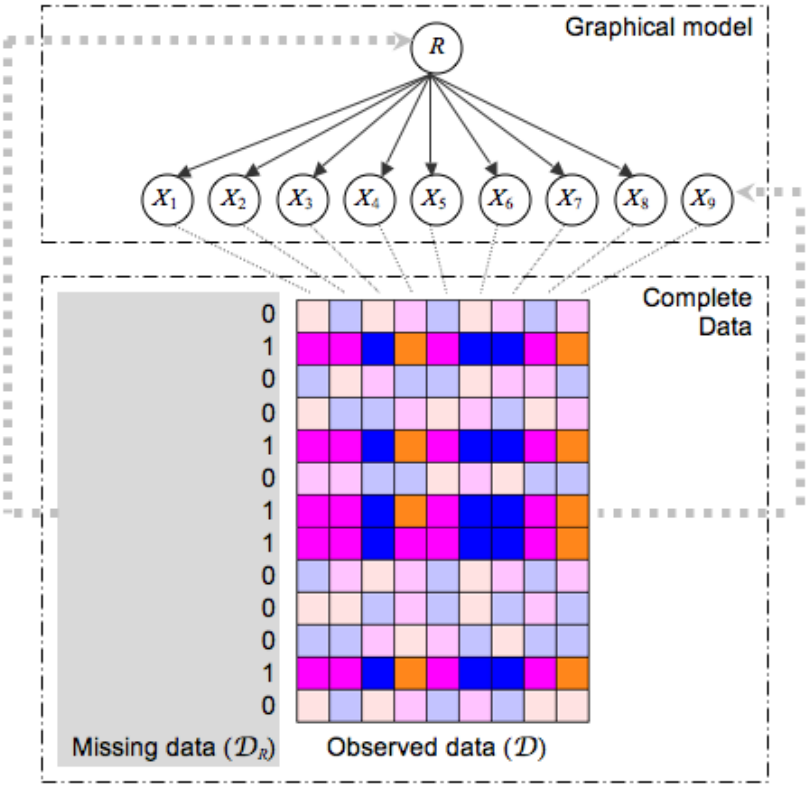


Figure 4.5: Data level of a Bayesian model for the clustering problem. The colored grids represent a microarray data matrix, where a cluster is highlighted. Each column in the data matrix is represented by a random variable X_i, ($i = 1, \ldots, 9$), and the rows are treated as data instances. A hidden variable R describes whether a row belongs to the cluster. Both $\mathbf{X}$ and R are random variables at the data level of the Bayesian hierarchical model. The upper part of the figure shows the graph model that depicts the relation between the random variables, which shows that $\mathbf{X}$ is dependent on R. The lower part of the figure shows the realization of the model. While the data for $\mathbf{X}$ is observed (i.e., microarray data), the task of clustering is to find out the value of R for each row, which is unknown.

Figure 4.5. Each row in the data matrix, is considered as a data instance, and is described by m random variables ($\mathbf{X}_m$), corresponding to the m experimental conditions in the data set. The distribution of $\mathbf{X}_m$ depends on the value of a hidden variable R. In Figure 4.5, R has only two possible values—"1" to indicate that the row belongs to the cluster, and "1" for otherwise. However, R can also take value $1, \ldots, k$ indicating to which of the k components (of the mixture) the row belongs. The values for $\mathbf{X}_m$ are observed (i.e., the microarray data $\mathcal{D}$), however, the values for R is not. Therefore, the data for R is called missing data. The nodes in the graph only represent the data level of the problem. The edges in Figure 4.4 (i.e., the conditional distribution between $\mathbf{X}_m$ and R) is where the hierarchical model discussed in Section 4.3.1 comes in for $\mathbf{X}_m$,

$$p(\mathbf{X}_m \mid R = r) = f(\boldsymbol{\Theta}_{\mathbf{X}_m \mid r}), \tag{4.24}$$

where $f(\boldsymbol{\Theta}_{\mathbf{X}_m|y})$ is a probability density function for $\mathbf{X}_m$, and $\boldsymbol{\Theta}_{\mathbf{X}_m|r}$ stands for the set of parameters for the density function when $R = r$. For example, when a normal mixture model is used (see Section 3.5.1),

$$f(\boldsymbol{\Theta}_{\mathbf{X}_m|r}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{X}_m|r}, \boldsymbol{\Sigma}_{\mathbf{X}_m|r}). \tag{4.25}$$

$\boldsymbol{\Theta}_{\mathbf{X}_m|r}$ is further modeled by a prior distribution $p(\boldsymbol{\Theta}_{\mathbf{X}_m|r} \mid \xi_{\mathbf{X}_m|r})$.

Graphical models for the biclustering problem

Let us first restate the biclustering problem. We mentioned in the beginning of this chapter that biclustering can be applied to both orientations of a microarray data matrix—i.e., biclustering the genes and biclustering the experiments, see Figure 1.6 for an illustration. In the rest of this chapter, we generalize the problem by treating both of the problems as biclustering the rows of a matrix. That is to say, in the case for biclustering experiments, we transpose the matrix. 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 the matrix illustrated in Figure 4.6). Note that we search for one bicluster at a time. (We will talk about how to find multiple biclusters later in this chapter.)

As shown in Figure 4.6, the n rows of a microarray data matrix are the data instances in the problem, and the m columns corresponds to the random variables, represented by $\mathbf{X}_m$. A hidden variable R describes whether a row belongs to the bicluster. Thus, R only takes the value of 0 (which indicates that the data point belongs to the background; i.e., not in the bicluster) and 1 (which indicates the data point belongs to the bicluster). Unlike in the clustering case, only part of the observed variables $\mathbf{X}^{\text{bcl}} \in \mathbf{X}_m$ is conditioned on the hidden variable R, while the rest of the variables $\mathbf{X}^{\text{bgd}} \in \mathbf{X}_m$, ($\mathbf{X}^{\text{bcl}} \cap \mathbf{X}^{\text{bgd}} = \emptyset$ and $\mathbf{X}^{\text{bcl}} \cup \mathbf{X}^{\text{bgd}} = \mathbf{X}_m$) are always modeled by the background distribution. This relationship between $\mathbf{X}_m$ and R is depicted by Figure 4.6.

The graph model in Figure 4.6 only illustrates the data level of the Bayesian hierarchical model (see Figure 4.2). The rest of the Bayesian hierarchical model

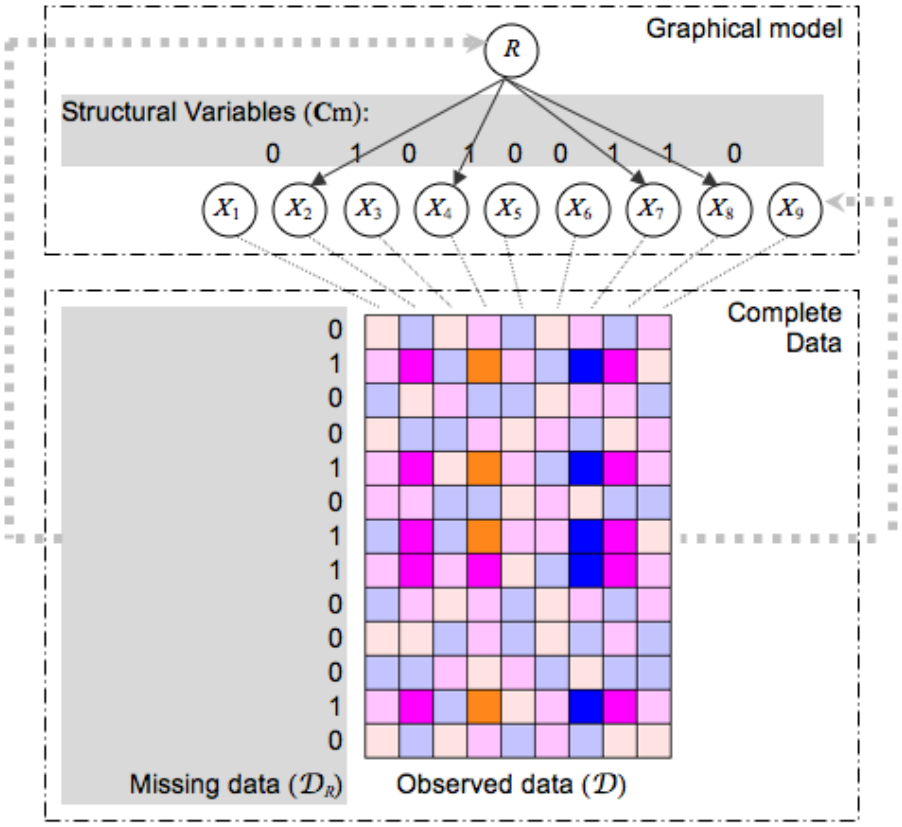


Figure 4.6: Data level of a Bayesian model for biclustering. The colored matrix represents microarray data, where an embedded bicluster is highlighted. Each column in the data matrix is represented by a random variable X_i, $(i = 1, \ldots, 9)$. The rows in the matrix are treated as data instances. A hidden variable R describes whether a row belongs to the cluster. Both $\mathbf{X}$ and R are random variables at the data level of the Bayesian hierarchical model. The upper part of the figure shows the graph model that depicts the relation between the random variables. Unlike the case for clustering problems, now only those X_i's whose represented column belong to the bicluster are dependent on R. This dependence is reflected by the edges in the graph model. In addition, a set of structure variables $\mathbf{C}_m$ is introduced. Each value C_j, $(j = 1, \ldots, 9)$, indicates whether the corresponding edge is represented in the graph. The lower part of the figure shows the realization of data level in the hierarchical model. While the data for $\mathbf{X}$ is observed (i.e., microarray data), the task of clustering is to find out the value of R for each row, which is unknown. Moreover, an additional task for biclustering is to infer the value of the structural variables $\mathbf{C}_m$ (i.e., to learn the structure of the graph model).

for $\mathbf{X}_m$ is explained as follows. The conditional distribution for $\mathbf{X}^{\text{bcl}}$ is

$$p(\mathbf{X}^{\text{bcl}} \mid R = 1) = f(\mathbf{\Theta}^{\text{bcl}}), \tag{4.26}$$
$$p(\mathbf{X}^{\text{bcl}} \mid R = 0) = f(\mathbf{\Theta}^{\text{bgd}}). \tag{4.27}$$

The distribution for $\mathbf{X}^{\text{bgd}}$ is

$$p(\mathbf{X}^{\text{bgd}}) = f(\mathbf{\Theta}^{\text{bgd}}). \tag{4.28}$$

Again, $f(\cdot)$ denotes a probabilistic distribution in Equations 4.26 to 4.28. The parameters $\mathbf{\Theta}^{\text{bcl}}$ and $\mathbf{\Theta}^{\text{bgd}}$ are modeled by their prior distributions—$p(\mathbf{\Theta}^{\text{bcl}} \mid \xi^{\text{bcl}})$ and $p(\mathbf{\Theta}^{\text{bgd}} \mid \xi^{\text{bgd}})$ respectively.

Learning the model: Gibbs sampling vs. structural EM

Gibbs sampling has become a popular alternative to the EM for solving the incomplete-data problem when the data structure is known, such as in the case of using mixture models for clustering (see Figure 4.5). (See Section 3.5 for a brief explanation of EM.) The problem is to estimate the missing data (i.e., values of the hidden variables) as well as the model parameters.

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 [28], (1) assuming that the model parameters are known, it calculates the expected value of the hidden variables (i.e., the sufficient statistics[‡] of the missing data), (2) with the expected values of the hidden variables estimated, it finds the model parameter that maximizes the likelihood computed on the complete data. Instead of the likelihood function, the posterior distribution of the complete data can also be used as the target function for the maximization step to accommodate the introduction of prior knowledge and thus to put the method in a Bayesian context. The procedure is guaranteed to converge to a stable solution under general conditions [111]. However, the obtained solution of the EM often gets stuck at local maxima modes of the likelihood function (or the posterior distribution). To alleviate the problem, multiple runs of EM procedure with independent initializations are often performed, and the solution with the highest likelihood is selected.

Gibbs sampling, on the other hand, treats the model parameters as random variables as well—i.e., in the same way as the hidden variables. The task for the Gibbs sampling procedure is therefore to estimate the joint posterior distribution $p(\mathcal{D}_R, \mathbf{\Theta} \mid \mathcal{D})$, where $\mathcal{D}_R$ stands for the data for the hidden variable R (i.e., the value of R for each row of the data matrix). As explained in Section 4.2,

[‡]With D representing the data and θ representing the parameter of the underlying probability distribution describing D, a statistic F(D) is sufficient for θ if the conditional probability distribution $p(D \mid F(D))$ does not depend on θ.

the Gibbs sampling strategy estimates the joint distribution by sampling from the full conditional distributions of the random variables involved (i.e., nodes), and the PME estimates of the random variables are obtained by performing Monte Carlo integrations. In other words, PME estimates are made after an estimate of the whole posterior distribution is obtained (by the samples collected during the Gibbs sampling procedure). This strategy increases the probability of finding the global maximum solution.

In the above case, the known structure of the graph implies the assumption that the association of columns to the bicluster is known. The estimation of the value of the hidden variable R answers whether a row of the matrix belongs to the bicluster. However, for our biclustering problem, neither the association of the rows, nor the association of the columns is known. In addition to the value of the hidden variables and the model parameters, the task includes finding the edges between R and $\mathbf{X}_m$; i.e., the structure of the graph, see Figure 4.6.

The task of Bayesian inference is therefore extended. Using $\mathcal{M}$ to denote the structure of the graphical model, Equation 4.20 can be rewritten as

$$p(\boldsymbol{\Theta} \mid \mathcal{D}, \mathcal{M}, \xi) = \frac{p(\boldsymbol{\Theta} \mid \mathcal{M}, \xi) \cdot p(\mathcal{D} \mid \boldsymbol{\Theta}, \mathcal{M}, \xi)}{p(\mathcal{D} \mid \mathcal{M}, \xi)}. \tag{4.29}$$

With the model structure unknown, another term specifying the prior for the model structure, $p(\mathcal{M} \mid \kappa)$, needs to be added to the denominator on the right-hand-side of Equation 4.29, where κ is the hyperparameter for the distribution of $\mathcal{M}$

$$p(\boldsymbol{\Theta} \mid \mathcal{D}, \mathcal{M}, \kappa, \xi) = \frac{p(\mathcal{M} \mid \kappa) \cdot p(\boldsymbol{\Theta} \mid \mathcal{M}, \xi) \cdot p(\mathcal{D} \mid \boldsymbol{\Theta}, \mathcal{M}, \xi)}{p(\mathcal{D} \mid \mathcal{M}, \kappa, \xi)}. \tag{4.30}$$

In this case, both structural EM [34] (which can be seen as an extension of the EM algorithm) and Gibbs sampling can be used for solving the problem.

Structural EM extends EM by a step for structural search. Starting with an initial structure $\mathcal{M}^{(0)}$, the algorithm performs an EM procedure to find the estimations for the hidden variables and the model parameters. To put the procedure in a Bayesian context, PME is used for the maximization step—that is,

$$\hat{\Theta} = \text{argmax}_{\Theta}\{E[\mathcal{D}^c \mid \Theta, \mathcal{M}, \xi]\}, \tag{4.31}$$

where $\mathcal{D}^c$ denotes the complete data. Then, a Bayesian information criterion (BIC) [84] score is calculated for the model structure based on the complete data and the optimized model parameter, $s(\mathcal{D}^c, \hat{\Theta}, \mathcal{M}, \xi)$. Next, a structural search is performed, and the BIC score is calculated in the same fashion for each of the encountered model structures. The model structure with the highest score is selected as the starting point for the next iteration of structure search. The whole procedure iterates until the BIC score converges.

As for the Gibbs sampling strategy, the structural estimation is performed in the same manner as the estimation for the hidden variables and the model parameters. Whether there is an edge between R and X_j (for $j = 1, \ldots, m$) is seen as a random event C_j, see Figure 4.6. Note that C_m is thus equivalent to $\mathcal{M}$. The target posterior joint distribution becomes $p(\mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta} \mid \mathcal{D})$. Gibbs sampling is often thought to be a computational intensive procedure for structural learning. However, because candidate structures for the biclustering problem conform to the same general structure, (namely only node R is allowed to be a parent node, see Figure 4.6), we found Gibbs sampling to be an efficient strategy to find the global maximum mode in the posterior mixture model. Because in this case, the number of structural variables ($\mathbf{C}_m$) added to the Gibbs sampling procedure is linear (instead of exponential) with respect to the number of random variables ($\mathbf{X}_m$).

In addition, as we have mentioned several times, Gibbs sampling paints out the entire posterior distribution of interest (by the samples that it collects) before a PME estimate is made. This property makes Gibbs sampling a suitable candidate for solving model-based problems in bioinformatics, where the likelihood function or the posterior function usually consists of a large number of modes because of the high complexity of the data. Though it takes Gibbs sampling longer to converge and to collect the samples (than performing one run of EM or structural EM), considering that there is little knowledge in advance about how many multiple runs are needed for EM or structural EM to find the global maximum, we consider Gibbs sampling an efficient technique for solving this type of problem in bioinformatics.

4.4 Gibbs sampling for biclustering

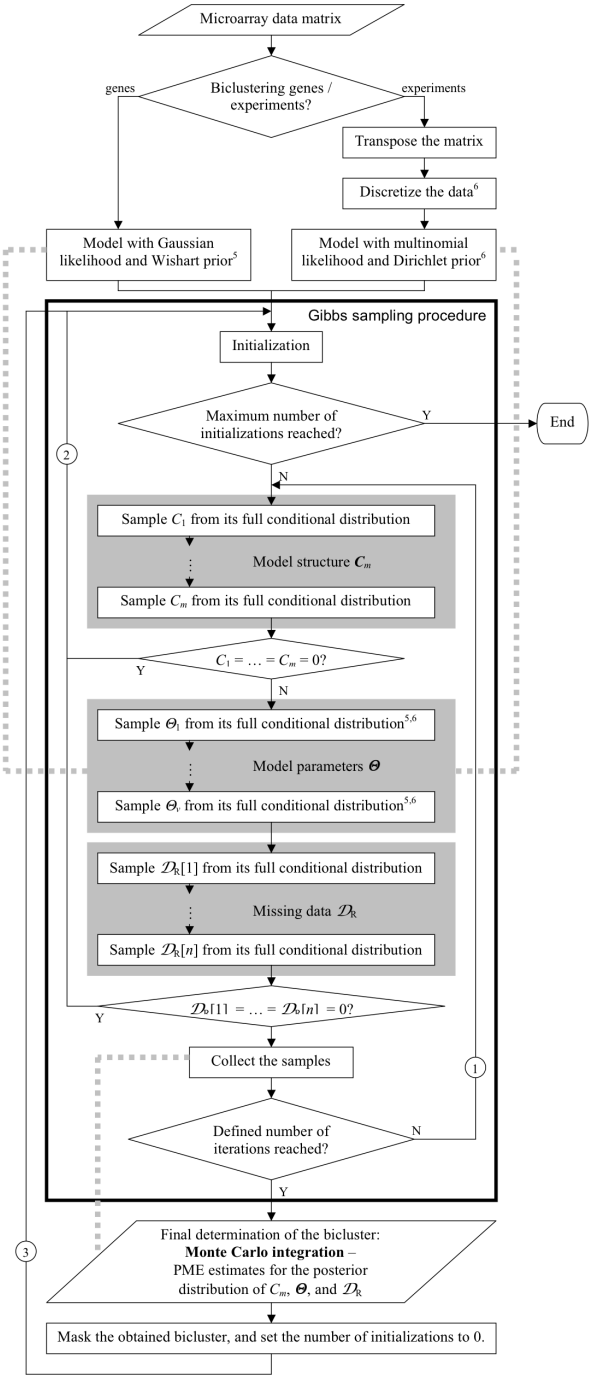
The entire Gibbs sampling scheme for biclustering is summarized in Figure 4.7. In what follows, we explain the scheme in detail.

We have mentioned in Section 4.3 that the Bayesian hierarchical models are applied to the conditional distributions that are represented by the edges in the graph. In addition, to put the whole model in a Bayesian context means that Bayesian hierarchical models are also used for R and $\mathbf{C}_m$ in Figure 4.6. All of them are Bernoulli variables by nature (as explained in Section 4.3),

$$P(R) \sim \text{Bernoulli}(\Lambda^{\text{r}}), \tag{4.32}$$

$$P(C_j) \sim \text{Bernoulli}(\Lambda^{\text{c}}) \quad j = 1, \ldots, m, \tag{4.33}$$

where Λ^{r} and Λ^{c} are the parameters of the Bernoulli distribution (see Appendix for a description of the distribution). Equation 4.33 implies the assumption that whether there is an edge between R and X_i is modeled by the same distribution for $i = 1, \ldots, m$. The Bernoulli parameters Λ^{r} and Λ^{c} are further modeled by their respective priors—conjugate priors are used for this purpose—which fol-



5 The procedure will be explained in detail in Chapter 5.
6 The procedure will be explained in detail in Chapter 6.

Figure 4.7: The Gibbs sampling scheme for biclustering.

lows Beta distributions (see the Appendix for a discussion on Beta distribution, and see Section 5.5 for a definition and discussion of conjugate priors.)

$$p(\Lambda^{\mathrm{r}}) \sim \mathrm{Beta}(\zeta^{\mathrm{r}}), \tag{4.34}$$
$$p(\Lambda^{\mathrm{c}}) \sim \mathrm{Beta}(\zeta^{\mathrm{c}}). \tag{4.35}$$

The model level (see Figure 4.2) of the whole Bayesian model for the biclustering problem (which is composed of model parameters $\mathbf{\Theta}^{\mathrm{bcl}}$, $\mathbf{\Theta}^{\mathrm{bgd}}$—see Equation 4.26 to Equation 4.28, Λ^{r}, and Λ^{c}), the data of the hidden variable (i.e., $\mathcal{D}_R$), and the structure of the data level of the Bayesian hierarchical model are the targets of Gibbs sampling. However, the prior level of the hierarchical model consists of hyperparameters ξ^{bcl}, ξ^{bgd}, and the ζ's, and is treated as input of the algorithm, and is not updated during the Gibbs sampling procedure, see Figure 4.2.

We discuss in more details about $\mathbf{\Theta}^{\mathrm{bcl}}$, $\mathbf{\Theta}^{\mathrm{bgd}}$ and their priors in the Bayesian hierarchical models in Chapter 5 and Chapter 6 where different models are established for biclustering experiments and biclustering genes respectively. In the following, we discuss some common steps for the Gibbs sampling procedure that are carried out in both cases.

Including the hyperparameters of the prior distributions in the target joint distribution, the task of the Gibbs sampling procedure is to infer the posterior joint distribution

$$p(\mathcal{D}_R, \mathbf{C}_m, \mathbf{\Theta}^{\mathrm{bcl}}, \mathbf{\Theta}^{\mathrm{bgd}}, \Lambda^{\mathrm{r}}, \Lambda^{\mathrm{c}}, | \mathcal{D}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}), \tag{4.36}$$

which means that the full conditional distribution of each of the random variables is needed to carry out Gibbs sampling. In this chapter, we consider the derivation of the full conditional distributions of the hidden data $\mathcal{D}_R$ and the variables for structural specification $\mathbf{C}_m$, as well as the manipulation of their prior Bernoulli parameters Λ^{r} and Λ^{c}. These procedures turn out to have some common interpretations for both biclustering genes and biclustering experiments.

4.4.1 The target posterior joint distribution

As we will find out later, a decomposed target posterior joint distribution makes the derivation of the full conditional distribution easier. Therefore, we start with some analysis of the target posterior joint distribution using Bayes' rule so that we can decompose the target posterior joint distribution. First, separately out the evidence component, we have

$$\begin{aligned} & p(\mathcal{D}_R, \mathbf{C}_m, \mathbf{\Theta}^{\mathrm{bcl}}, \mathbf{\Theta}^{\mathrm{bgd}}, \Lambda^{\mathrm{r}}, \Lambda^{\mathrm{c}} | \mathcal{D}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}) \\ = & \frac{p(\mathcal{D}, \mathcal{D}_R, \mathbf{C}_m, \mathbf{\Theta}^{\mathrm{bcl}}, \mathbf{\Theta}^{\mathrm{bgd}}, \Lambda^{\mathrm{r}}, \Lambda^{\mathrm{c}} | \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}})}{p(\mathcal{D} | \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}})}. \end{aligned} \tag{4.37}$$

The denominator in the above equation (i.e., the evidence) is a normalization term, which is independent of the model. Thus, we write,

$$\begin{aligned} & p(\mathcal{D}_R, \mathbf{C}_m, \mathbf{\Theta}^{\text{bcl}}, \mathbf{\Theta}^{\text{bgd}}, \Lambda^{\text{r}}, \Lambda^{\text{c}} \,|\, \mathcal{D}, \xi^{\text{bcl}}, \xi^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}) \\ & \quad \propto \quad p(\mathcal{D}, \mathcal{D}_R, \mathbf{C}_m, \mathbf{\Theta}^{\text{bcl}}, \mathbf{\Theta}^{\text{bgd}}, \Lambda^{\text{r}}, \Lambda^{\text{c}} \,|\, \xi^{\text{bcl}}, \xi^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}), \end{aligned} \tag{4.38}$$

where $\propto$ means proportional to. Then, according to the Bayes' rule (see Equation 4.20), the right-hand side in Equation 4.38 can be decomposed into components of likelihood and priors.

$$\begin{aligned} & p(\mathcal{D}, \mathcal{D}_R, \mathbf{C}_m, \mathbf{\Theta}^{\text{bcl}}, \mathbf{\Theta}^{\text{bgd}}, \Lambda^{\text{r}}, \Lambda^{\text{c}} \,|\, \xi^{\text{bcl}}, \xi^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}) \\ & \quad = \quad p(\mathcal{D}, \mathcal{D}_R \,|\, \mathbf{C}_m, \mathbf{\Theta}^{\text{bcl}}, \mathbf{\Theta}^{\text{bgd}}, \Lambda^{\text{r}}).p(\mathbf{\Theta}^{\text{bcl}}, \mathbf{\Theta}^{\text{bgd}} \,|\, \mathbf{C}_m, \xi^{\text{bcl}}, \xi^{\text{bgd}}) \\ & \qquad .p(\Lambda^{\text{r}} \,|\, \zeta^{\text{r}}) \cdot P(\mathbf{C}_m \,|\, \Lambda^{\text{c}}) \cdot p(\Lambda^{\text{c}} \,|\, \zeta^{\text{c}}). \end{aligned} \tag{4.39}$$

Adding the conditional independence of the observed data given the hidden data as depicted in Figure 4.6 to the above equation, we have

$$\begin{aligned} & p(\mathcal{D}\, \mathcal{D}_R, \mathbf{C}_m, \mathbf{\Theta}^{\text{bcl}}, \mathbf{\Theta}^{\text{bgd}}, \Lambda^{\text{r}}, \Lambda^{\text{c}} \,|\, \xi^{\text{bcl}}, \xi^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}) \\ & \quad = \quad p(\mathcal{D} \,|\, \mathcal{D}_R, \mathbf{C}_m, \mathbf{\Theta}^{\text{bcl}}, \mathbf{\Theta}^{\text{bgd}}).P(\mathcal{D}_R \,|\, \Lambda^{\text{r}}) \cdot P(\mathbf{C}_m \,|\, \Lambda^{\text{c}}) \\ & \qquad .p(\mathbf{\Theta}^{\text{bcl}}, \mathbf{\Theta}^{\text{bgd}} \,|\, \mathbf{C}_m, \xi^{\text{bcl}}, \xi^{\text{bgd}}).p(\Lambda^{\text{r}} \,|\, \zeta^{\text{r}}) \cdot p(\Lambda^{\text{c}} \,|\, \zeta^{\text{c}}) \\ & \quad = \quad \prod_{i=1}^{n} \left\{ p(\mathbf{X}_m = \mathcal{D}[i, \cdot] \,|\, R = \mathcal{D}_R[i], \mathbf{C}_m, \mathbf{\Theta}^{\text{bcl}}, \mathbf{\Theta}^{\text{bgd}}) \cdot P(R = \mathcal{D}_R[i] \,|\, \Lambda^{\text{r}}) \right\} \\ & \qquad \times p(\mathbf{\Theta}^{\text{bcl}}, \mathbf{\Theta}^{\text{bgd}} \,|\, \mathbf{C}_m, \xi^{\text{bcl}}, \xi^{\text{bgd}}).p(\Lambda^{\text{r}} \,|\, \zeta^{\text{r}}).p(\Lambda^{\text{c}} \,|\, \zeta^{\text{c}}).P(\mathbf{C}_m \,|\, \Lambda^{\text{c}}), \end{aligned} \tag{4.40}$$

where $\mathcal{D}[i, \cdot]$ denotes the i^{th} row in the microarray data matrix, and $\mathcal{D}_R[i]$ is the value for R for the corresponding row. The latter equality comes from the assumption that $\mathcal{D}[i, \cdot]$ (for $i = 1, \ldots, n$) are independently and identically distributed (i.i.d.) given the model—$\mathbf{C}_m$, $\mathbf{\Theta}^{\text{bcl}}$, and $\mathbf{\Theta}^{\text{bgd}}$—and the missing data $\mathcal{D}_R[i]$, and that $\mathcal{D}_R[i]$ is i.i.d. given Λ^{r}. From now on, we base the derivation of the full conditional distributions on Equation 4.40.

4.4.2 The manipulation of Λ^{r} and Λ^{c}

To decrease the number of parameters that needs to be estimated for our Bayesian hierarchical model, we show in the following that Λ^{r} and Λ^{c} can be integrated out of the target distribution, and will not be sampled during the Gibbs sampling procedure. Because conjugate priors are used for Λ^{r} and Λ^{c}, their full conditional distributions are in the same form as the prior—Beta

distributions.

$$
\begin{aligned}
& p(\mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}} \,|\, \mathcal{D}, \xi^{\text{bcl}}, \xi^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}) \\
& \quad \propto \int\int p(\mathcal{D}, \mathcal{D}_R, \mathbf{C}_m, \Lambda^{\text{r}}, \Lambda^{\text{c}}, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}} \,|\, \xi^{\text{bcl}}, \xi^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}})\, \mathrm{d}\Lambda^{\text{r}}\, \mathrm{d}\Lambda^{\text{c}} \\
& \quad = p(\mathcal{D} \,|\, \mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}).p(\boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}} \,|\, \mathbf{C}_m, \xi^{\text{bcl}}, \xi^{\text{bgd}}) \\
& \qquad . \int P(\mathcal{D}_R \,|\, \Lambda^{\text{r}}) \cdot p(\Lambda^{\text{r}} \,|\, \zeta^{\text{r}})\, \mathrm{d}\Lambda^{\text{r}}. \int P(\mathbf{C}_m \,|\, \Lambda^{\text{c}}) \cdot p(\Lambda^{\text{c}} \,|\, \zeta^{\text{c}})\, \mathrm{d}\Lambda^{\text{c}} \\
& \quad = p(\mathcal{D} \,|\, \mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}) \cdot P(\mathcal{D}_R \,|\, \zeta^{\text{r}}) \\
& \qquad \times p(\boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}} \,|\, \mathbf{C}_m, \xi^{\text{bcl}}, \xi^{\text{bgd}}).P(\mathbf{C}_m \,|\, \zeta^{\text{c}})
\end{aligned} \tag{4.41}
$$

Consequently, the target joint distribution becomes

$$
\begin{aligned}
& p(\mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}} \,|\, \mathcal{D}, \xi^{\text{bcl}}, \xi^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}) \\
& \quad \propto \; p(\mathcal{D}, \mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}} \,|\, \xi^{\text{bcl}}, \xi^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}).
\end{aligned} \tag{4.42}
$$

For the detailed integration of Λ^{r}, because $\mathcal{D}_R$ are independent Bernoulli trials given Λ^{r}, we have

$$
\begin{aligned}
P(\mathcal{D}_R \,|\, \zeta^{\text{r}}) &= \int P(\mathcal{D}_R \,|\, \Lambda^{\text{r}}) \cdot p(\Lambda^{\text{r}} \,|\, \zeta^{\text{r}})\, \mathrm{d}\Lambda^{\text{r}} \\
&= \int p(\Lambda^{\text{r}} \,|\, \zeta^{\text{r}}) \cdot \prod_{i=1}^{n} P(R = \mathcal{D}_R[i] \,|\, \Lambda^{\text{r}}) \mathrm{d}\Lambda^{\text{r}} \\
&\propto \int (\Lambda^{\text{r}})^{\zeta_1^{\text{r}}-1} \cdot (1-\Lambda^{\text{r}})^{\zeta_0^{\text{r}}-1} \cdot (\Lambda^{\text{r}})^{v} \cdot (1-\Lambda^{\text{r}})^{(n-v)} \mathrm{d}\Lambda^{\text{r}} \\
&\propto \frac{\Gamma(\zeta_0^{\text{r}} + n - v)\Gamma(\zeta_1^{\text{r}} + v)}{\Gamma(\zeta_0^{\text{r}} + \zeta_1^{\text{r}} + n)},
\end{aligned} \tag{4.43}
$$

where ζ_0^{r} and ζ_1^{r} are the two elements of ζ^{r} corresponding respectively to the prior probability that $R = 0$ and $R = 1$, and v denotes the number of $\mathcal{D}_R[i] = 1$ for $i = 1, \ldots, n$. Similarly, for $\mathbf{C}_m$, we have

$$
P(\mathbf{C}_m \,|\, \zeta^{\text{c}}) \propto \int P(\mathbf{C}_m \,|\, \Lambda^{\text{c}}) \cdot p(\Lambda^{\text{c}} \,|\, \zeta^{\text{c}})\, \mathrm{d}\Lambda^{\text{c}} \propto \frac{\Gamma(\zeta_0^{\text{c}} + m - w)\Gamma(\zeta_1^{\text{c}} + w)}{\Gamma(\zeta_0^{\text{c}} + \zeta_1^{\text{c}} + m)}, \tag{4.44}
$$

where ζ_0^{c} and ζ_1^{c} are the two elements of ζ^{c} corresponding respectively to the prior probability that $C_j = 0$ and $C_j = 1$ (for $j = 1, \ldots, m$), and w is the number of $C_j = 1$ for $j = 1, \ldots, m$.

Equation 4.43 and 4.44 show that after the integration, $\mathcal{D}_R[i]$ (for $i = 1, \ldots, n$) are not i.i.d. given ζ^{r}, and C_j (for $j = 1, \ldots, m$) also become dependent of each other given ζ^{c}.

4.4.3 Full conditional distributions of the missing data and the structural variables

We now consider the full conditional distributions of R. For the i^{th} row in the data, the full conditional probability $R = \mathcal{D}_R[i]$ is modeled by a Bernoulli distribution with parameter Λ_i^{r} (because of the use of conjugate priors—which will be explained in more detail in Section 5.5), which is the probability that $\mathcal{D}_R[i] = 1$.

$$\begin{aligned}
\Lambda_i^{\text{r}} &= P(\mathcal{D}_R[i] = 1 \,|\, \mathcal{D}, \mathcal{D}_R[\bar{i}], \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}, \boldsymbol{\xi}^{\text{bcl}}, \boldsymbol{\xi}^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}) \\
&\propto p(\mathcal{D}, \mathcal{D}_R[i] = 1, \mathcal{D}_R[\bar{i}] \,|\, \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}) \\
&= P(\mathcal{D}_R[i] = 1, \mathcal{D}_R[\bar{i}] \,|\, \zeta^{\text{r}}) \cdot p(\mathbf{X}_m = \mathcal{D}[i, \cdot] \,|\, R = 1, \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}) \\
&\quad \cdot \prod_{k=1, k\neq i}^{n} p(\mathbf{X}_m = \mathcal{D}[k, \cdot] \,|\, R = \mathcal{D}_R[k], \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}) \\
&= P(\mathcal{D}_R[i] = 1, \mathcal{D}_R[\bar{i}] \,|\, \zeta^{\text{r}}) \cdot \prod_{\{j \,|\, \mathbf{C}_j = 1\}} p(X_j = \mathcal{D}[i, j] \,|\, \boldsymbol{\Theta}^{\text{bcl}}) \\
&\quad \cdot \prod_{\{j \,|\, \mathbf{C}_j = 0\}} p(X_j = \mathcal{D}[i, j] \,|\, \boldsymbol{\Theta}^{\text{bgd}}) \\
&\quad \cdot \prod_{k=1, k\neq i}^{n} p(\mathbf{X}_m = \mathcal{D}[k, \cdot] \,|\, R = \mathcal{D}_R[k], \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}),
\end{aligned} \tag{4.45}$$

where $\mathcal{D}_R[\bar{i}]$ denotes the data in all the other rows except the i^{th} row in the microarray data matrix. (Though R is the random variable whose parameter we want to estimate, we use $\mathcal{D}_R[i]$ in the following to specify the value of R for the i^{th} row in the data matrix). The last equality in the above equation is justified by the conditional independence of $\mathbf{X}^{\text{bcl}}$ on R. The complement of the Bernoulli parameter, $1 - \Lambda_i^{\text{r}}$, is

$$\begin{aligned}
1 - \Lambda_i^{\text{r}} &= P(\mathcal{D}_R[i] = 0 \,|\, \mathcal{D}, \mathcal{D}_R[\bar{i}], \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}, \boldsymbol{\xi}^{\text{bcl}}, \boldsymbol{\xi}^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}) \\
&\propto P(\mathcal{D}_R[i] = 0, \mathcal{D}_R[\bar{i}] \,|\, \zeta^{\text{r}}) \cdot p(\mathbf{X}_m = \mathcal{D}[i, \cdot] \,|\, R = 0, \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}) \\
&\quad \cdot \prod_{k=1, k\neq i}^{n} p(\mathbf{X}_m = \mathcal{D}[k, \cdot] \,|\, R = \mathcal{D}_R[k], \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}) \\
&= P(\mathcal{D}_R[i] = 1, \mathcal{D}_R[\bar{i}] \,|\, \zeta^{\text{r}}) \cdot \prod_{j=1}^{m} p(X_j = \mathcal{D}[i, j] \,|\, \boldsymbol{\Theta}^{\text{bgd}}) \\
&\quad \cdot \prod_{k=1, k\neq i}^{n} p(\mathbf{X}_m = \mathcal{D}[k, \cdot] \,|\, R = \mathcal{D}_R[k], \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}).
\end{aligned} \tag{4.46}$$

Note that the Bernoulli parameter Λ_i^{r} can be transformed to the odds γ_i^{r} between Λ_i^{r} and $1 - \Lambda_i^{\mathrm{r}}$,

$$\gamma_i^{\mathrm{r}} = \frac{\Lambda_i^{\mathrm{r}}}{1 - \Lambda_i^{\mathrm{r}}} \tag{4.47}$$

$$\Lambda_i^{\mathrm{r}} = \frac{\gamma_i^{\mathrm{r}}}{1 + \gamma_i^{\mathrm{r}}} \tag{4.48}$$

We show in the following equation that γ_i^{r} has an interpretable meaning:

$$\begin{aligned}
\gamma_i^{\mathrm{r}} &= \frac{P(\mathcal{D}_R[i] = 1 \,|\, \mathcal{D}, \mathcal{D}_R[\bar{i}], \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \boldsymbol{\xi}^{\mathrm{bcl}}, \boldsymbol{\xi}^{\mathrm{bgd}}, \boldsymbol{\zeta}^{\mathrm{r}}, \boldsymbol{\zeta}^{\mathrm{c}})}{P(\mathcal{D}_R[i] = 0 \,|\, \mathcal{D}, \mathcal{D}_R[\bar{i}], \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \boldsymbol{\xi}^{\mathrm{bcl}}, \boldsymbol{\xi}^{\mathrm{bgd}}, \boldsymbol{\zeta}^{\mathrm{r}}, \boldsymbol{\zeta}^{\mathrm{c}})} \\
&= \frac{P(\mathcal{D}_R[i] = 1, \mathcal{D}_R[\bar{i}] \,|\, \boldsymbol{\zeta}^{\mathrm{r}})}{P(\mathcal{D}_R[i] = 0, \mathcal{D}_R[\bar{i}] \,|\, \boldsymbol{\zeta}^{\mathrm{r}})} \cdot \frac{p(\mathcal{D} \,|\, \mathcal{D}_R[i] = 1, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}})}{p(\mathcal{D} \,|\, \mathcal{D}_R[i] = 0, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}})} \\
&= \frac{\Gamma(n - v_{\bar{i}} + \zeta_0^{\mathrm{r}})\Gamma(v_{\bar{i}} + 1 + \zeta_1^{\mathrm{r}})}{\Gamma(n + \zeta_0^{\mathrm{r}} + \zeta_1^{\mathrm{r}})} \cdot \frac{\Gamma(n + \zeta_0^{\mathrm{r}} + \zeta_1^{\mathrm{r}})}{\Gamma(n - v_{\bar{i}} + 1 + \zeta_0^{\mathrm{r}})\Gamma(v_{\bar{i}} + \zeta_1^{\mathrm{r}})} \\
&\quad \cdot \frac{p(\mathcal{D} \,|\, \mathcal{D}_R[i] = 1, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}})}{p(\mathcal{D} \,|\, \mathcal{D}_R[i] = 0, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}})} \\
&= \frac{v_{\bar{i}} + \zeta_1^{\mathrm{r}}}{n - v_{\bar{i}} + \zeta_0^{\mathrm{r}}} \cdot \prod_{\{j \,|\, C_j = 1\}} \frac{p(X_j = \mathcal{D}[i, j] \,|\, \boldsymbol{\Theta}^{\mathrm{bcl}})}{p(X_j = \mathcal{D}[i, j] \,|\, \boldsymbol{\Theta}^{\mathrm{bgd}})}.
\end{aligned} \tag{4.49}$$

We use $v_{\bar{i}}$ to denote the number of rows (all but the i^{th} row in the microarray data matrix) that currently belongs to the bicluster. The equation tells us that given the model structure ($\mathbf{C}_m$) and model parameters ($\boldsymbol{\Theta}^{\mathrm{bcl}}$ and $\boldsymbol{\Theta}^{\mathrm{bgd}}$), the full conditional odds of whether the row under consideration belongs to the bicluster is given by a weighted likelihood ratio. The likelihood is calculated between the case where the data of the row under the biclustering columns are generated by the bicluster and the case where these data are generated by the background. The weight is calculated as the ratio between the number of rows that are currently assigned to the bicluster and the number of rows that currently belong to the background. (Note that γ_i^{r} is dependent only on $\mathbf{C}_m$ and $\boldsymbol{\Theta}$.)

The parameter γ_j^{c} of the full conditional posterior distribution of C_j (for $j = 1, \ldots, m$) depends on more parameters. Because of the dependence of $\boldsymbol{\Theta}$ on $\mathbf{C}_m$, the derivation of the distribution requires specification of the type of distributions and especially the correlations between the model parameters $\boldsymbol{\Theta}$ for the Bayesian hierarchical model. A first analysis of the Bernoulli

distribution shows that its parameter is

$$\begin{aligned}\Lambda_j^{\mathrm{c}} &= p(C_j = 1 \,|\, \mathcal{D}, \mathcal{D}_R, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}) \\ &\propto p(\mathcal{D}, \mathcal{D}_R, C_j = 1, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \,|\, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}) \\ &= p(\mathcal{D} \,|\, \mathcal{D}_R, C_j = 1, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}) \cdot P(\mathcal{D}_R \,|\, \zeta^{\mathrm{r}}) \\ &\quad \cdot p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \,|\, C_j = 1, \mathbf{C}_{\bar{j}}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}) \cdot P(C_j = 1, \mathbf{C}_{\bar{j}} \,|\, \zeta^{\mathrm{c}}). \end{aligned} \tag{4.50}$$

The complement of the parameter is,

$$\begin{aligned}1 - \Lambda_j^{\mathrm{c}} &= p(C_j = 1 \,|\, \mathcal{D}, \mathcal{D}_R, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}) \\ &\propto p(\mathcal{D} \,|\, \mathcal{D}_R, C_j = 0, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}) \cdot P(\mathcal{D}_R \,|\, \zeta^{\mathrm{r}}) \\ &\quad \cdot p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \,|\, C_j = 0, \mathbf{C}_{\bar{j}}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}) \cdot P(C_j = 0, \mathbf{C}_{\bar{j}} \,|\, \zeta^{\mathrm{c}}). \end{aligned} \tag{4.51}$$

Therefore the odds between the two, $\gamma_j^{\mathrm{c}} = \frac{\Lambda_j^{\mathrm{c}}}{1-\Lambda_j^{\mathrm{c}}}$, is

$$\begin{aligned}\gamma_j^{\mathrm{c}} &= \frac{p(C_j = 1 \,|\, \mathcal{D}, \mathcal{D}_R, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}})}{p(C_j = 1 \,|\, \mathcal{D}, \mathcal{D}_R, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}})} \\ &= \frac{p(\mathcal{D} \,|\, \mathcal{D}_R, C_j = 1, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}})}{p(\mathcal{D} \,|\, \mathcal{D}_R, C_j = 0, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}})} \\ &\quad \cdot \frac{p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \,|\, C_j = 1, \mathbf{C}_{\bar{j}}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}})}{p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \,|\, C_j = 0, \mathbf{C}_{\bar{j}}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}})} \cdot \frac{w_{\bar{j}} + \zeta_1^{\mathrm{c}}}{m - w_{\bar{j}} + \zeta_0^{\mathrm{c}}}. \end{aligned} \tag{4.52}$$

In Equations 4.50 to 4.52, we use $\mathbf{C}_{\bar{j}}$ to denote the edges between R and $\mathbf{X}_m$ except for X_j, and $w_{\bar{j}}$ for the current number of columns in the microarray data matrix that belong to the bicluster. Equation 4.52 tells us that γ_j^{c} is a product of (1) the likelihood ratio of the data (between the case when the j^{th} column of the microarray data is generated by the bicluster and the case when it is generated by the background), (2) a ratio between the priors of $\boldsymbol{\Theta}$, and (3) a ratio between the number of columns that are already in the bicluster and the number of those that are in the background.

4.4.4 The Gibbs sampling scheme for the biclustering problem

To summarize, the Gibbs sampling scheme for the biclustering problem is as follows, also see the procedure within the black dashed box in Figure 4.7.

1. Initialization: Assign binary random values to $\mathbf{C}_m$ (i.e., initialize the model structure) and $\mathcal{D}_R$ (i.e., initialize the missing data), and initialize the model parameters $\boldsymbol{\Theta}$

2. Update the hidden data: fix the model—both $\mathbf{C}_m$ and $\boldsymbol{\Theta}$—for each row i, $(i = 1, \ldots, n)$, fix $\mathcal{D}_R[\bar{i}]$, and
   (a) Calculate the Bernoulli distribution for $\mathcal{D}_R[i]$ whose parameter $\Lambda_i^{\mathrm{r}} = \frac{\gamma_i^{\mathrm{r}}}{1+\gamma_i^{\mathrm{r}}}$ can be calculated by applying Equation 4.49.
   (b) Draw a sample for $\mathcal{D}_R[i]$ from the Bernoulli distribution.
3. Update the model structure: fix $\mathcal{D}_R$ and $\boldsymbol{\Theta}$, for each C_j, $(j = 1, 2, ..., m)$, fix the value of all $\mathbf{C}_{\bar{j}}$, and
   (a) Calculate the Bernoulli distribution for C_j whose parameter $\Lambda_j^{\mathrm{c}} = \frac{\gamma_j^{\mathrm{c}}}{1+\gamma_j^{\mathrm{c}}}$ can be calculated by applying Equation 4.52.
   (b) Draw a sample for C_j from the Bernoulli distribution.
4. Sample the model parameters Θ according to their conditional distributions (see Chapter 5 and Chapter 6 for detail).
5. Go to Step 2, and iterate for a predefined number of iterations, see Loop 1 in Figure 4.7.

4.4.5 From samples to the final pattern

To evaluate every involved parameter in the target posterior distribution (Equation 4.42) means to collect samples produced at each iteration of the Gibbs sampling (Loop 1 in Figure 4.7) for each of these parameters. The final PME estimates of these parameters are then obtained by performing Monte Carlo integrations the collected samples. For the accuracy of the PME estimates, the number of samples should be as large as possible. This means that the number of sampling iterations of the Gibbs sampler is usually several hundreds. Storing samples of all the parameters (especially those of $\boldsymbol{\Theta}$ which is proportional to the number of columns in microarray data) for each iteration can dramatically increase the memory complexity of the algorithm. However, the main purpose of the algorithm is to find the position of the bicluster, which require the storage of only those samples of $\mathcal{D}_R$ and $\mathbf{C}_m$. The parameters of the model $\boldsymbol{\Theta}$ will be obtained—if necessary—by evaluating their sample statistics according to the position of the bicluster determined by the algorithm. According to Equation 4.13, the PME estimate of $\mathcal{D}_R[i]$ is.

$$\mathrm{E}_{P(\mathcal{D}_R[i]\,|\,\mathcal{D})}\Big[\mathcal{D}_R[i]\Big] = \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}_{P(\mathcal{D}_R[i]\,|\,\mathcal{D}_R[\bar{i}]^{(t)}, \mathbf{C}_m^{(t)}, (\boldsymbol{\Theta}^{\mathrm{bcl}})^{(t)}, (\boldsymbol{\Theta}^{\mathrm{bgd}})^{(t)}, \mathcal{D})}\Big[\mathcal{D}_R[i]\Big] \quad i = 1, \ldots, n. \tag{4.53}$$

Similarly, the PME estimate of C_j is,

$$\mathrm{E}_{P(C_j\,|\,\mathcal{D})}[C_j] = \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}_{P(C_j\,|\,\mathcal{D}_R^{(t)}, \mathbf{C}_{\bar{j}}^{(t)}, (\boldsymbol{\Theta}^{\mathrm{bcl}})^{(t)}, (\boldsymbol{\Theta}^{\mathrm{bgd}})^{(t)}, \mathcal{D})}[C_j] \quad j = 1, \ldots, m. \tag{4.54}$$

The marginal distribution of $\mathcal{D}_R[i]$ and C_j, according to Equation 4.14, are

$$\mathrm{E}[P(\mathcal{D}_R[i] \mid \mathcal{D})] = \frac{1}{T}\sum_{t=1}^{T} P(\mathcal{D}_R[i] \mid \mathcal{D}_R[\bar{i}]^{(t)}, \mathbf{C}_m^{(t)}, (\boldsymbol{\Theta}^{\mathrm{bcl}})^{(t)}, (\boldsymbol{\Theta}^{\mathrm{bgd}})^{(t)}, \mathcal{D}) \quad i = 1, \ldots, n, \tag{4.55}$$

$$\mathrm{E}[P(C_j \mid \mathcal{D})] = \frac{1}{T}\sum_{t=1}^{T} P(C_j \mid \mathcal{D}_R^{(t)}, \mathbf{C}_{\bar{j}}^{(t)}, (\boldsymbol{\Theta}^{\mathrm{bcl}})^{(t)}, (\boldsymbol{\Theta}^{\mathrm{bgd}})^{(t)}, \mathcal{D}) \quad j = 1, \ldots, m. \tag{4.56}$$

Note that to simplify the notation, we omitted the priors ξ^{bcl}, ξ^{bgd}, ζ^{r}, and ζ^{c} out of the notation of the probability distributions. In Equations 4.53 to 4.56, 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 PME estimate of the random variables, and select only the rows and columns for whom the PME 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 random variables.

In our implementation, we use 500 iteration as default for performing the Gibbs sampling procedure (including both the burn-in and the sampling stages). The autocorrelation (see Equation 4.17) between the samples is monitored for determining the convergence, and additional iterations are added if necessary.

4.4.6 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 (see Loop 3 in Figure 4.7). By masking, we mean that the random variable (subset of $\mathbf{C}_m$ in the case of biclustering genes, and subset of $\mathcal{D}_R$ in the case of biclustering experiments) associated with 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 candidates 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 let the genes to be selected in multiple biclusters. 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 4.4.6 for the decision).

Another approach to find multiple biclusters would be to add multiple hidden nodes to the model structure. 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.

Data without a bicluster

To decide that a data set does not or no longer contains a bicluster, we check the number of genes or conditions that belong to the bicluster after Step 2 and Step 3 of the algorithm (see Section 4.4.4). If either of the numbers equals zero, we reinitialize the algorithm and perform Gibbs sampling again, see Loop 2 in Figure 4.7. 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.

4.5 Conclusion

In this chapter, we developed a Bayesian hierarchical model for the biclustering problem, and a Gibbs sampling strategy for refining the structure and the parameterization of the model. We explained in detail the framework of the Gibbs sampling procedure for the biclustering problem. In the following two chapters, we develop dedicated models, which specifies the distribution of and the priors for $\boldsymbol{\Theta}$, respectively for the biclustering of experiments and the biclustering of genes.

Chapter 5

Biclustering experiments in microarray data

In this chapter, we describe a dedicated Bayesian hierarchical model for the problem of biclustering experiments. The model is developed for a discretized microarray data matrix. We first explain why and how to discretize the microarray data. Then, we elaborate on the model and re-explain the Gibbs sampling framework for the model. We show two types of usage of the algorithm by using different prior settings—the first one is to discovery a global bicluster embedded in the data by using a non-informative prior, and the second is to construct a specific prior model to direct the bicluster discovery for a specific pathology.

5.1 Introduction

In this chapter we discuss Gibbs sampling to for biclustering experiments (e.g., tumor samples) on discretized microarray data [89]. The aim is to find experiments whose discrete expression levels are consistent for each gene selected for the bicluster (see Figure 1.6). To keep in accordance with the Gibbs sampling frame that we gave in Chapter 4, we transpose the discretized microarray data matrix (see Figure 4.7) for the biclustering analysis, so that now the rows represent the experiments, and the columns represent the genes.

The choice of using discretized data for the biclustering of experiments was not only inspired by the success of applying Gibbs sampling to motif finding problem in DNA sequence analysis [64, 68, 100], where the data has a discrete nature, so that the mathematical models of the motif finding problem can be conveniently extended to biclustering. This choice is also justified by the con-

sideration that the experiment dimension of microarray data is usually much larger than its gene dimension. Using a normal distribution to model a gene expression profile, for example, is often found to be sensitive to outliers [71]. However, the use of discrete data avoids the problem of outliers by significantly reducing the noise level in the data while reserving the most essential information for biologists.

We discuss the Gibbs sampling strategy for tackling the biclustering problem of experiments in the following four aspects:

- *Discretization of microarray data*: why and how to discretize the microarray data for the biclustering of experiments.
- *Data model*: the hierarchical Bayesian models describing the bicluster and the background.
- *Full conditional distributions*: the distributions from which samples of the missing data $\mathcal{D}_R$ and the variables for model structure specification $\mathbf{C}_m$ are drawn during the Gibbs sampling procedure.
- *More notes on the priors*: Incorporating prior knowledge into the hierarchical model.

By using different constructions of the priors, the biclustering algorithm can be used to assist discoveries of pathology under two circumstances. In the first case, we assume that the biologist (or the doctor) has no idea about the pathology types of the tumors from which samples are collected for the microarray experiments. The task of biclustering is then to discover global patterns that are embedded in the data, which provide expressional fingerprints for different pathology types. However, the expressional patterns of some pathological traits dominate others in their size and amplitude, and consequently dominate the results of biclustering. To enhance the ability of the biclustering algorithm to discover fingerprints for those less dominant pathological traits, priors can be introduced to the algorithm by specifying a small number of positive examples of the tumors that should belong to the pathology of interest. We discuss the usage and the performance of the biclustering algorithm for these two purposes individually. In each case, synthetic data sets are used to illustrate the influence of the input parameters and the performance of the algorithm, and a case study is provided to illustrate the ability of the algorithm to assist the discovery of pathologies.

5.2 The discretization of microarray data

In the biclustering of experiments, the different microarray experiments are the data instances for evaluating the Bayesian hierarchical model, which the genes

Expression profile	1.23	1.55	-0.43	-0.22	0.54	0.11	1.19	0.25	0.6
Rank	8	9	1	2	5	3	7	4	6
Discrete values	3	3	1	1	2	1	3	2	2

Figure 5.1: Discretization with the equal frequency principle: first the expression values in a gene expression profile are ranked from the lowest to the highest. Then, the third of the data points with the lowest rank is assigned with discrete value 1 ("low"), the third of data points ranked in the middle are assigned with discrete value 2 ("medium"), and the top ranked third of data points are assigned with discrete value 3 ("high"). The ranks of equal expression values are decided by the order that they appear in the expression profile—values that appear earlier in the vector (of expression profile) get lower ranks. See Figure 5.2.

are the random variables in the model, see Figure 1.6. Therefore, the sample space has a much smaller dimension than the variable space (i.e., number of experiments is much smaller than the number of genes). An extreme expression value (i.e., outlier) in a gene expression profile can significantly influence the estimate of the probabilistic distribution that describes the expression profile. Discretization groups the continuous values together and thus reduce the number of distinct values. Therefore, discretized expression profiles are more resistant to outliers.

We found that the equal-frequency discretization provides a biologically meaningful and mathematically reasonable method for discretizing microarray data for the purpose of biclustering experiments. By discretizing microarray data into three bins, the discrete gene expression levels correspond respectively to "high" (or "upregulated"), "median" (or "not activated"), and "low" (or "downregulated"). Mathematically speaking, the discrete data obtained by equal-frequency discretization have maximum entropy (i.e., the data represent a uniform distribution). This means that the method does not (in principle) introduce additional information during the discretization procedure (though we discuss an exception in practice at the end of this section, see Figure 5.2). The procedure of equal-frequency discretization is carried out as follows. For each gene profile, we assign the experiments with the lowest third of the expression values to the first bin (corresponding to "low"), similarly, the third experiments with the highest expression values to the third bin (corresponding to "high"), and finally, the rest of the experiments to the second bin (corresponding to "medium"), see Figure 5.1. We choose to use three bins for the discretization because the discrete levels have more intuitive biological interpretations—high/upregulated, medium/not active, and low/downregulated. In addition, we found that discretizations into different number of bins influence little the biclustering results.

Before shuffling the patients:

Patient class	ALL	ALL	ALL	MLL	MLL	MLL	AML	AML	AML
Expression profile	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.2
Rank	1	1	1	1	1	1	1	1	1
Discrete values	1	1	1	2	2	2	3	3	3

After shuffling the patients:

Patient class	AML	ALL	MLL	ALL	MLL	AML	AML	ALL	MLL
Expression profile	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.2
Rank	1	1	1	1	1	1	1	1	1
Discrete values	1	1	1	2	2	2	3	3	3

Figure 5.2: An artifact when applying equal frequency principle discretization on data sets where experiments (e.g., patients or tumor samples) of the same group are put next to each other. An extreme case is shown in the top figure where the resulting discrete data is coupled with the patient class though there is no variation at all in the gene expression profile. Shuffling the order of the patients help to decouple such effect—bottom figure.

To deal with the missing values in the microarray data, we assign them randomly to one of the three bins, so that the maximum-entropy property of the equal-frequency principle is preserved.

Note that the equal-frequency discretization takes care of the standardization of the gene expression profiles automatically, as the discretization is done per gene profile. Therefore, problems arise if the continuous expression profile of a gene remains constant over different experiments. Thus, a filtering procedure based on the variation of the genes is needed before the equal-frequency discretization is performed. For microarray data sets where the experiments are grouped (e.g., tumors of the same type are arranged next to each other in the data matrix), we further an artifact for equal-frequency discretization by permuting the order of the experiments in the data set before performing the discretization, see Figure 5.2.

5.3 The model

The equal frequency discretization procedure justifies the use of a single multinomial distribution to describe the background data, with each entry of the multinomial distribution explaining the frequency of observing the corresponding discrete expression level in background data. We use Ψ to denote the parameter vector of the multinomial distribution,

$$\Psi = [\ \psi_1 \quad \psi_2 \quad \psi_3\ ]^{\mathrm{T}}, \tag{5.1}$$

where $0 \leq \psi_i \leq 1$, for $i = \{1, 2, 3\}$, $\sum_{i=1}^{3} \psi = 1$. Note that the background model is independent of the model structure,

$$\Psi \perp\!\!\!\perp \mathbf{C}_m. \tag{5.2}$$

For the bicluster, to allow the experiments (i.e., rows) for different genes (i.e., columns) to have different expression levels (although the expression levels of the experiments should be the same under the same gene), we use a multinomial distribution to model the data for every gene in a bicluster,

$$\Phi_j = [\ \phi_{1,j} \quad \phi_{2,j} \quad \phi_{3,j}\ ]^{\mathrm{T}} \quad \forall\{j \,|\, C_j = 1,\ j = 1, \ldots, n\}, \tag{5.3}$$

$$\mathbf{\Phi} = \{\Phi_j \,|\, C_j = 1,\ j = 1, \ldots, m\}, \tag{5.4}$$

where $0 \leq \phi_{i,j} \leq 1$, for $i = \{1, 2, 3\}$, $\sum_{i=1}^{3} \phi_{i,j} = 1$; and we assume that the multinomial distributions for different conditions of a bicluster are mutually independent,

$$\Phi_j \perp\!\!\!\perp \Phi_k \quad j \neq k \quad \forall \Phi_j, \Phi_k \in \mathbf{\Phi}. \tag{5.5}$$

Therefore, the bicluster model $\mathbf{\Phi}$ is dependent on $\mathbf{C}_m$.

Both $\mathbf{\Phi}$ and Ψ form the model parameters (denoted as $\mathbf{\Theta}$ in Chapter 4) for the distribution of the observed variables $\mathbf{X}_m$ into which the discrete expression data map. The explicit distribution of $\mathbf{X}_m$ is now: for the background,

$$P(X_j \,|\, R = 0) \sim \text{Multinomial}(\Psi) \quad j = 1, \ldots, m; \tag{5.6}$$

for the model,

$$P(X_j \,|\, R = 1) \sim \begin{cases} \text{Multinomial}(\Phi_j) & \{j \,|\, C_j = 1\} \\ \text{Multinomial}(\Psi) & \{j \,|\, C_j = 0\} \end{cases} \tag{5.7}$$

As we discussed in Chapter 4, the prior level is one of the indispensable ingredients of a Bayesian hierarchical model. The priors in our model play a main role in directing the discovery of biclusters, as will be explained in Section 5.5 and will be further illustrated in Section 5.6 and Section 5.7. To introduce the complete Bayesian hierarchical model that we use for the biclustering of

experiments, we briefly describe here the type of priors that we use in this case.

We assume that each multinomial distribution in Ψ is modeled by a corresponding prior. Further, we use conjugate priors for these multinomial distributions, which are in the form of Dirichlet distributions,

$$\begin{aligned} \Phi_j &\sim \text{Dirichlet}(\boldsymbol{\beta}_j), && (5.8)\\ \boldsymbol{\beta}_j &= [\ \beta_{1,j} \quad \beta_{2,j} \quad \beta_{3,j}\ ]^{\mathrm{T}}, && (5.9)\\ \mathrm{B} &= \{\boldsymbol{\beta}_j \,|\, C_j = 1\}. && (5.10) \end{aligned}$$

The Dirichlet distribution is also used for the prior of Ψ,

$$\begin{aligned} \Psi &\sim \text{Dirichlet}(\boldsymbol{\alpha}), && (5.11)\\ \boldsymbol{\alpha} &= [\ \alpha_1 \quad \alpha_2 \quad \alpha_3\ ]^{\mathrm{T}} && (5.12) \end{aligned}$$

(We motivate our choice for using conjugate priors in Section 5.5.)

5.4 Full conditional distributions

Substituting our model into Equation 4.42, the target joint distribution of the Gibbs sampling procedure for the biclustering problem of experiments is

$$p(\mathcal{D}_R, \mathbf{C}_m, \Psi, \boldsymbol{\Phi} \,|\, \mathcal{D}, \boldsymbol{\alpha}, \mathrm{B}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}) \propto p(\mathcal{D}, \mathcal{D}_R, \mathbf{C}_m, \Psi, \boldsymbol{\Phi} \,|\, \boldsymbol{\alpha}, \mathrm{B}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}). \quad (5.13)$$

The procedure is carried out by sampling iteratively from the full conditional distributions of $\mathcal{D}_R$, $\mathbf{C}_m$, Ψ, and $\boldsymbol{\Phi}$. We have made some general analysis of the Bernoulli conditional distributions of $\mathcal{D}_R$ and $\mathbf{C}_m$ in Section 4.4.3. We will make a more detailed analysis about these two distributions later on in this section. But let us first consider the manipulation of Ψ and $\boldsymbol{\Phi}$.

Using conjugate priors for Ψ and $\boldsymbol{\Phi}$ means that the full conditional distributions of the parameters are also in the form of Dirichlet distributions. Sampling from Dirichlet distributions is not a trivial procedure and consumes a non-negligible amount of computation [68]. In addition, the number of parameters in Φ is proportional to (or three times—when three bins are used for the discretization) the number of genes that are included in the bicluster, which could greatly increase the computational complexity of the algorithm. This procedure can be avoided by integrating Ψ and $\boldsymbol{\Phi}$ out of the target joint distribution (similar to how we deal with Λ^{r} and Λ^{c}, see Section 4.4.2). Consequently, the target joint distribution becomes $P(\mathcal{D}_R, \mathbf{C}_m \,|\, \mathcal{D}, \boldsymbol{\alpha}, \mathrm{B}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}})$.

To facilitate the analysis of the full conditional distributions of $\mathcal{D}_R$ and $\mathbf{C}_m$, we need the help of the following notation. First of all, we use lower-case

bold letters to denote vectors of indices of the rows and the columns of the microarray data matrix. More specifically,

$$\mathbf{r} = \left[i \,|\, \mathcal{D}_R[i] = 1 \right] \tag{5.14}$$

is the vector of the indices of the data instances of R, (i.e., the rows, or in this case the experiments) that are assigned to the bicluster, and

$$\bar{\mathbf{r}} = \left[i \,|\, \mathcal{D}_R[i] = 0 \right] \tag{5.15}$$

is the vector of indices of the data instances of R that are assigned to the background. Similarly,

$$\mathbf{c} = \left[j \,|\, C_j = 1 \right] \tag{5.16}$$

is the vector of the indices of structure variables in $\mathbf{C}_m$ (i.e., in this case, the columns, or the genes) that are assigned to the bicluster, and

$$\bar{\mathbf{c}} = \left[j \,|\, C_j = 0 \right] \tag{5.17}$$

is the vector of the indices of the columns that are assigned to the background. In addition, a subscript $\bar{i}$ (or $\bar{j}$) means that the bicluster and the background is evaluated on the data excluding the i^{th} row (or the j^{th} column) of the matrix. For example, $\bar{\mathbf{r}}_{\bar{i}}$ refers to the rows that are assigned to the background excluding the i^{th} row (regardless of the value of $\mathcal{D}_R[i]$, i.e., whether the row belongs to the bicluster or not).

The data can be either indexed by two integers (e.g., $\mathcal{D}[i, j]$), which refers to the data point at the i^{th} row and the j^{th} column, or by two vectors of indices. Given two vectors of indices $\mathbf{u}$ and $\mathbf{v}$, $\mathcal{D}[\mathbf{u}, \mathbf{v}]$ refers to the part of the data under rows $\mathbf{u}$ and columns $\mathbf{v}$.

We define $h(\cdot)$ as a counting function. Thus $h(\mathcal{D}[\mathbf{u}, \mathbf{v}])$ produces a vector of length three, with each of its entries giving the number of occurrences of the corresponding discrete level (1, 2, or 3) in the specified region of the data matrix. For example, for the following discrete matrix,

$$\mathcal{D} = \begin{bmatrix} 2 & 1 & 3 & 2 \\ 1 & 1 & 2 & 2 \end{bmatrix} \tag{5.18}$$

$$h(\mathcal{D}) = [3, 4, 1]^{\mathrm{T}}, \tag{5.19}$$

because "1" is observed 3 times in the matrix, "2" is observed 4 times in the matrix, and "3" is observed 1 time in the matrix.

For two scalars u and v, we use $\delta_u(v)$ to denote an index vector of length u whose v^{th} entry equals 1, and the rest of its entries equal 0.

$$\delta_3(2) = [0,\ 1,\ 0]^{\mathrm{T}}$$

Given a vector of indices $\mathbf{u}$ and a scalar v, $\mathbf{u} \oplus v$ denotes concatenating v to vector $\mathbf{u}$. For example,

$$\begin{aligned} \mathbf{u} &= [11, 42, 29]^{\mathrm{T}} \\ v &= 63 \\ \mathbf{u} \oplus v &= [11, 42, 29, 63]^{\mathrm{T}}. \end{aligned}$$

Further, for two vectors $\boldsymbol{u} = [u_1, u_2, \ldots, u_k]^{\mathrm{T}}$ and $\boldsymbol{v} = [v_1, v_2, \ldots, v_k]^{\mathrm{T}}$ (of the same length k), we define the power function $\boldsymbol{u}^{\boldsymbol{v}} = u_1^{v_1} \cdot u_2^{v_2} \cdots u_k^{v_k}$, the Gamma function $\Gamma(\boldsymbol{u}) = \Gamma(u_1) \cdot \Gamma(u_2) \cdots \Gamma(u_k)$, and the sum $\sum \boldsymbol{u} = \sum_{i=1}^{k} u_i$.

Coming back to the full conditional distribution of $\mathcal{D}_R[i]$, the final result in Equation 4.49 is not applicable anymore after the integration on the model parameters, because $\mathcal{D}_R[i]$ and $\mathcal{D}_R[j]$ for all $i, j = 1 \ldots n$, $i \neq j$ are conditionally independent given the model parameters ($\boldsymbol{\Theta}^{\mathrm{bcl}} = \boldsymbol{\Phi}$ and $\boldsymbol{\Theta}^{\mathrm{bgd}} = \Psi$) in that equation. Therefore, we have to recalculate the parameter Λ_i^{r} for the posterior conditional Bernoulli distribution for $\mathcal{D}_R[i]$, for which we use the odds $\gamma_i^{\mathrm{r}} = \frac{\Lambda_i^{\mathrm{r}}}{1-\Lambda_i^{\mathrm{r}}}$ instead again (for the use of Step 2 in the Gibbs sampling procedure described in Section 4.4.4).

$$\begin{aligned} \gamma_i^{\mathrm{r}} &= \frac{P(\mathcal{D},\, \mathcal{D}_R[i] = 1,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m \,|\, \boldsymbol{\alpha},\, \mathrm{B},\, \boldsymbol{\zeta}^{\mathrm{r}},\, \boldsymbol{\zeta}^{\mathrm{c}})}{P(\mathcal{D},\, \mathcal{D}_R[i] = 0,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m \,|\, \boldsymbol{\alpha},\, \mathrm{B},\, \boldsymbol{\zeta}^{\mathrm{r}},\, \boldsymbol{\zeta}^{\mathrm{c}})} \\ &= \frac{P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 1,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})}{P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 0,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})} \cdot \frac{P(\mathcal{D}_R[i] = 1,\, \mathcal{D}_R[\bar{i}] \,|\, \boldsymbol{\zeta}^{\mathrm{r}})}{P(\mathcal{D}_R[i] = 0,\, \mathcal{D}_R[\bar{i}] \,|\, \boldsymbol{\zeta}^{\mathrm{r}})} \\ &= \frac{P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 1,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})}{P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 0,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})} \cdot \frac{v_{\bar{i}} + \zeta_1^{\mathrm{r}}}{n - v_{\bar{i}} + \zeta_0^{\mathrm{r}}}. \end{aligned} \tag{5.20}$$

The above equation shows that when $\boldsymbol{\Phi}$ and Ψ are absent, the posterior distribution of $\mathcal{D}_R[i]$ is dependent on the rest of the missing data $\mathcal{D}_R[\bar{i}]$ and the model structure $\mathbf{C}_m$.

To calculate the first term in Equation 5.20, we perform the integration respectively for $P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 1,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})$ and $P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 0,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})$, we have, for each experiment i,

$$\begin{aligned} & P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 1,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B}) \\ &= \int_{\boldsymbol{\Phi}} \int_{\Psi} P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 1,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\Phi},\, \Psi) \cdot p(\boldsymbol{\Phi} \,|\, \mathrm{B},\, \mathbf{C}_m) \cdot p(\Psi \,|\, \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\ &\propto \int_{\boldsymbol{\Phi}} \int_{\Psi} \Psi^{h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}},\, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \bar{\mathbf{c}}]\}} \cdot \prod_{\{j \,|\, C_j = 1\}} \Phi_j^{h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \delta_m(j)]\}} \cdot \Psi^{\boldsymbol{\alpha} - 1} \cdot \prod_{\{j \,|\, C_j = 1\}} \Phi_j^{\boldsymbol{\beta}_j - 1}\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\ &= \frac{\Gamma\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}},\, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}},\, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)\Big\}} \cdot \prod_{\{j \,|\, C_j = 1\}} \frac{\Gamma\Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)\Big\}}, \\ &\quad \text{for } i = 1 \ldots n, \end{aligned} \tag{5.21}$$

and

$$
\begin{aligned}
& P(\mathcal{D} \mid \mathcal{D}_R[i]=0,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m, \boldsymbol{\alpha}, \mathrm{B}) \\
= & \int_{\boldsymbol{\Phi}} \int_{\Psi} P(\mathcal{D} \mid \mathcal{D}_R[i]=0,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\Phi},\, \Psi) \cdot p(\boldsymbol{\Phi} \mid \mathrm{B},\, \mathbf{C}_m) \cdot (\Psi \mid \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
\propto & \int_{\boldsymbol{\Phi}} \int_{\Psi} \Psi^{h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}} \oplus i,\, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \bar{\mathbf{c}}]\}} \cdot \prod_{\{j \mid C_j = 1\}} \Phi_j^{h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \delta_m(j)]\}} \cdot \Psi^{\boldsymbol{\alpha}-1} \cdot \prod_{\{j \mid C_j = 1\}} \Phi_j^{\boldsymbol{\beta}_j - 1}\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
= & \frac{\Gamma\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}} \oplus i,\, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}} \oplus i,\, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)\Big\}} \cdot \prod_{\{j \mid C_j = 1\}} \frac{\Gamma\Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)\Big\}} \\
& \text{for } i = 1 \ldots n. \qquad (5.22)
\end{aligned}
$$

Note that the denominators omitted by the proportional mark "$\propto$" are the same for both Equation 5.21 and Equation 5.22, which means that

$$
\frac{P(\mathcal{D} \mid \mathcal{D}_R[i]=1,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})}{P(\mathcal{D} \mid \mathcal{D}_R[i]=0,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})}
$$

equals the ratio between the end result of Equation 5.21 and Equation 5.22.

To evaluate this ratio, we first need to use the nature of the Gamma function,

$$
\Gamma(u+1) = \Gamma(u) \cdot u \qquad (5.23)
$$

where u is a scalar. Such calculation reveals that ratios (part of the ratio of Equation 5.21 and Equation 5.22) have simpler forms:

$$
\begin{aligned}
\frac{\Gamma\Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)}{\Gamma\Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \delta_m(j)]\} + \boldsymbol{\beta}_{\cdot,j}\Big)} &= \Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)^{\delta_3\{\mathcal{D}[i,j]\}} \qquad (5.24) \\
\{i \mid i = 1 \ldots n\}, \{j \mid C_j = 1\}, &
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)\Big\}}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)\Big\}} &= \frac{\Gamma(v_{\bar{i}} + 1 + \sum \boldsymbol{\beta}_j)}{\Gamma(v_{\bar{i}} + \sum \boldsymbol{\beta}_j)} = v_{\bar{i}} + \sum \boldsymbol{\beta}_j, \qquad (5.25) \\
\text{for } i = 1, \ldots, n,\, j = 1, \ldots, w. &
\end{aligned}
$$

To calculate the ratio of the first factors in Equation 5.21 and 5.22, we need the approximation that when the size of $\mathbf{u}_2$ is relatively small compared to $\mathbf{u}_1$ and the composition of $\mathbf{u}_2$ is relatively diverse [68],

$$
\frac{\Gamma\{h(\mathbf{u}_1) + h(\mathbf{u}_2)\}}{\Gamma\{h(\mathbf{u}_1)\}} \approx h(\mathbf{u}_1)^{h(\mathbf{u}_2)}. \qquad (5.26)
$$

$h(\mathbf{u}_1)$	[16, 238, 46]	[16, 238, 46]
$h(\mathbf{u}_2)$	[3, 3, 4]	[0, 0, 10]
$\log\left(\frac{\Gamma\{h(\mathbf{u}_1)+h(\mathbf{u}_2)\}}{\Gamma\{h(\mathbf{u}_1)\}}\right)$	40.37	39.20
$\log\left(h(\mathbf{u}_1)^{h(\mathbf{u}_2)}\right)$	40.05	38.32

The following table shows an example:

In our case, the values in $h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}$ are much larger those in $h\{\mathcal{D}[\delta_n(i), \mathbf{c}]\}$, and the composition of $h\{\mathcal{D}[\delta_n(i), \mathbf{c}]\}$ is relatively diverse. Therefore,

$$\begin{aligned}
&\frac{\Gamma\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}} \oplus i, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}}, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)} \\
&= \frac{\Gamma\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}}, \bar{\mathbf{c}}]\} + h\{\mathcal{D}[\delta_n(i), \bar{\mathbf{c}}]\} + \boldsymbol{\alpha} + h\{\mathcal{D}[\delta_n(i), \mathbf{c}]\}\Big)}{\Gamma\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}}, \bar{\mathbf{c}}]\} + h\{\mathcal{D}[\delta_n(i), \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)} \\
&\approx \Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)^{h\{\mathcal{D}[\delta_n(i), \mathbf{c}]\}}, \quad i = 1, \ldots, n. \qquad (5.27)
\end{aligned}$$

Similarly,

$$\begin{aligned}
&\frac{\Gamma\Big\{\sum\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}} \oplus i, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}}, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)\Big\}}{\Gamma\Big\{\sum\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)\Big\}} \\
&= \sum\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)^{\sum(h\{\mathcal{D}[\delta_n(i), \mathbf{c}]\})}, \quad i = 1, \ldots, n. \qquad (5.28)
\end{aligned}$$

Equipped with Equations 5.24 to 5.28, we put Equation 5.21, Equation 5.22 together to evaluate Equation 5.20. We finally arrive at

$$\gamma_i^{\mathrm{r}} \approx \prod_{\{j \,|\, C_j = 1\}} \left(\frac{\hat{\Phi}_j}{\hat{\Psi}}\right)^{\delta_3(\mathcal{D}[i,j])} \cdot \frac{v_{\bar{i}} + \zeta_1^{\mathrm{r}}}{n - 1 - v_{\bar{i}} + \zeta_0^{\mathrm{r}}}, \quad i = 1, \ldots, n, \qquad (5.29)$$

where

$$\hat{\Phi}_j = \frac{h\{\mathcal{D}[r_{\bar{i}}, \delta_m(j)]\} + \boldsymbol{\beta}_j}{v_{\bar{i}} + \sum \boldsymbol{\beta}_j} \qquad \{j \,|\, C_j = 1\} \qquad (5.30)$$

$$\hat{\Psi} = \frac{h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}}, \bar{\mathbf{c}}]\} + h\{\mathcal{D}[\delta_n(i), \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}}{n \cdot m - v_{\bar{i}} \cdot w}. \qquad (5.31)$$

The final result of γ_i^{r} in Equation 5.29 is exactly in the same form as Equation 4.49. $\hat{\Phi}$ and $\hat{\Psi}$ are the byproduct of our method, which represent respectively the model of the bicluster and the model of the background. Equation 5.30 and 5.31 reveal that $\hat{\Phi}$ and $\hat{\Psi}$ are essentially the posterior bicluster model evaluated at the currently assigned biclustering positions, and the

posterior background model evaluated at the currently assigned background positions. The "current" bicluster and background refers to the data divided according to $\mathcal{D}_R[\bar{i}]$ and $\mathbf{C}_m$. Intuitively, by fixing all the other random variables $(\mathbf{C}_m, \mathcal{D}_R[\bar{i}])$ to the values sampled in previous Gibbs sampling steps, the possibility that experiment i belongs to the bicluster is associated with the likelihood that the data of the experiment for the genes that are currently assigned to the bicluster is generated by the bicluster model; while the possibility that the experiment belongs to the background is related to the likelihood that those data points are drawn from the background model.

Similar to Equation 5.20, for the odds γ_j^c in evaluating the posterior Bernoulli distribution of the model structural variables $C_j \in \mathbf{C}_m$ (for the use in Setp 3 of the Gibbs sampling procedure described in Section 4.4.4), we have

$$\gamma_j^c = \frac{w_{\bar{j}} + \zeta_1^c}{m - w_{\bar{j}} + \zeta_0^c} \cdot \frac{P(\mathcal{D} \,|\, \mathcal{D}_R,\, C_j = 1,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\alpha}, \mathrm{B})}{P(\mathcal{D} \,|\, \mathcal{D}_R,\, C_j = 0,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\alpha},\, \mathrm{B})}. \tag{5.32}$$

Comparing the graph (see Figure 4.6) where the edge between nodes R and X_j is present (i.e., $C_j = 1$) and the one without the edge (i.e., $C_j = 0$), the model of the bicluster for the former case can be seen as having an extra column copied from the background model, whose multinomial parameter vector is ϕ_j whose distribution is parameterized by $\boldsymbol{\alpha}$. Evaluating $P(\mathcal{D} \,|\, \mathcal{D}_R,\, C_j = 1,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\alpha}, \mathrm{B})$ and $P(\mathcal{D}, \,|\, \mathcal{D}_R,\, C_j = 0,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\alpha},\, \mathrm{B})$ respectively to calculate Equation 5.32, we have

$$\begin{aligned}
&P(\mathcal{D} \,|\, \mathcal{D}_R,\, C_j = 1,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\alpha}, \mathrm{B}) \\
&= \int_{\boldsymbol{\Phi}} \int_{\Psi} P(\mathcal{D} \,|\, \mathcal{D}_R,\, C_j = 1,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\Phi},\, \Psi) \cdot p(\boldsymbol{\Phi} \,|\, C_j = 1, \mathbf{C}_{\bar{j}},\, \mathrm{B}) \cdot p(\Psi \,|\, \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
&\propto \int_{\boldsymbol{\Phi}} \int_{\Psi} \Psi^{h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}}]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}} \oplus j]\}} \cdot \prod_{\{k \,|\, C_k = 1,\, k \neq j\}} \Phi_k^{h\{\mathcal{D}[\mathbf{r},\, \delta_m(k)]\}} \cdot \Phi_j^{h\{\mathcal{D}[\mathbf{r},\, \delta_m(j)]\}} \\
&\quad \cdot \Psi^{\boldsymbol{\alpha}-1} \cdot \prod_{\{k \,|\, C_k = 1,\, k \neq j\}} \Phi_k^{\boldsymbol{\beta}_k - 1} \cdot \Phi_j^{\boldsymbol{\alpha}-1}\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
&= \frac{\Gamma\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}}]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}} \oplus j]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}}]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}} \oplus j]\} + \boldsymbol{\alpha}\Big)\Big\}} \cdot \prod_{\{k \,|\, C_k = 1,\, k \neq j\}} \frac{\Gamma\Big(h\{\mathcal{D}[\mathbf{r},\, \mathbf{c}_k]\} + \boldsymbol{\beta}_k\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r},\, \mathbf{c}_k]\} + \boldsymbol{\beta}_k\Big)\Big\}} \\
&\quad \cdot \frac{\Gamma\Big(h\{\mathcal{D}[\mathbf{r},\, \delta_m(j)]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r},\, \delta_m(j)]\} + \boldsymbol{\alpha}\Big)\Big\}}, \quad j = 1 \ldots m,
\end{aligned} \tag{5.33}$$

and

$$
\begin{aligned}
& P(\mathcal{D} \mid \mathcal{D}_R,\, C_j = 0,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\alpha},\, \mathrm{B}) \\
&= \int_{\boldsymbol{\Phi}} \int_{\Psi} P(\mathcal{D} \mid \mathcal{D}_R,\, C_j = 0,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\Phi},\, \Psi) \cdot p(\boldsymbol{\Phi} \mid C_j = 0,\, \mathbf{C}_{\bar{j}},\, \mathrm{B}) \cdot p(\Psi \mid \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
&\propto \int_{\boldsymbol{\Phi}} \int_{\Psi} \Psi^{h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}} \oplus j]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}}]\}} \cdot \prod_{\{k \mid C_k = 1,\, k \neq j\}} \Phi_k^{h\{\mathcal{D}[\mathbf{r},\, \mathbf{c}_k]\}} \cdot \Psi^{\boldsymbol{\alpha} - 1} \cdot \prod_{\{k \mid C_k = 1,\, k \neq j\}} \Phi_k^{\boldsymbol{\beta}_k}\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
&= \frac{\Gamma\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}} \oplus j]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}}]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}} \oplus j]\} + h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{j}},\, \mathbf{c}]\} + \boldsymbol{\alpha}\Big)\Big\}} \cdot \prod_{\{k \mid C_k = 1,\, k \neq j\}} \frac{\Gamma\Big(h\{\mathcal{D}[\mathbf{r},\, \mathbf{c}_k]\} + \boldsymbol{\beta}_k\}}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r},\, \mathbf{c}_k]\} + \boldsymbol{\beta}_k\Big)\Big\}}, \\
&\quad \text{for} \quad j = 1 \ldots m. \qquad (5.34)
\end{aligned}
$$

Putting the above two equations together to calculate Equation 5.32, we get

$$
\begin{aligned}
\gamma_j^{\mathrm{c}} &= \frac{\Gamma\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}}]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}} \oplus j]\} + \boldsymbol{\alpha}\Big) \cdot \Gamma\Big(h\{\mathcal{D}[\mathbf{r},\, j]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}} \oplus j]\} + h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{j}},\, \mathbf{c}]\} + \boldsymbol{\alpha}\Big)} \\
&\quad \cdot \frac{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}} \oplus j]\} + h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{j}},\, \mathbf{c}]\} + \boldsymbol{\alpha}\Big)\Big\}}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}}]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}} \oplus j]\} + \boldsymbol{\alpha}\Big)\Big\} \cdot \Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r},\, j]\} + \boldsymbol{\alpha}\Big)\Big\}} \\
&\quad \cdot \frac{(w_{\bar{j}} + \zeta_1^{\mathrm{c}})}{(m - w_{\bar{j}} - 1 + \zeta_0^{\mathrm{c}})} \\
&= \frac{\Gamma\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}}]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}} \oplus j]\} + \boldsymbol{\alpha}\Big) \cdot \Gamma\Big(h\{\mathcal{D}[\mathbf{r},\, j]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}} \oplus j]\} + h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{j}},\, \mathbf{c}]\} + \boldsymbol{\alpha}\Big)} \\
&\quad \cdot \frac{\Gamma(n \cdot m - v \cdot w_{\bar{j}} + \sum \boldsymbol{\alpha})}{\Gamma(n \cdot m - v \cdot w_{\bar{j}} - v + \sum \boldsymbol{\alpha}) \cdot \Gamma(v + \sum \boldsymbol{\alpha})} \cdot \frac{(w_{\bar{j}} + \zeta_1^{\mathrm{c}})}{(m - w_{\bar{j}} - 1 + \zeta_0^{\mathrm{c}})} \\
&\quad j = 1 \ldots m. \qquad (5.35)
\end{aligned}
$$

Note that the equation cannot be simplified, because the conditions for the approximation in Equation 5.26 to hold are no longer satisfied. Intuitively, the denominator of Equation 5.35 assumes the current prediction of the background and extends the current bicluster by treating the j^{th} condition as one of the biclustering conditions, while the second term in Equation 5.35 adds the j^{th} condition to the currently assigned background. Again, the current bicluster and background refers to the data divided according to $\mathcal{D}_R$ and $\mathbf{C}_{\bar{j}}$.

5.5 Importance of the priors

It is important to emphasize again (see explanations in Chapter 4) that the power of the Bayesian model lies in its incorporation of prior knowledge in

deriving the posterior probability. A proper prior should be interpretable so that knowledge of a human expert or from other information sources can be meaningfully transformed into probabilistic distributions (or densities). Further, it should be mathematically convenient for the computation of the posterior. Conjugate priors are often considered to be suitable choices regarding these two aspects. Conjugate priors refer to the class of prior distributions (or density functions) that, when combined with the target likelihood function (i.e., probabilistic model of the data), produce posterior of the same form as the prior. Therefore, conjugate priors are convenient to use in the sense that the output we want (i.e., the posterior distribution) is in the same format and has the same metric as the input we impose (i.e., the prior distribution). For these reasons, conjugate priors are used throughout this thesis.

The Dirichlet priors $\boldsymbol{\alpha}$ and B are conjugate priors for our multinomial models on the bicluster and the background. They express our prior knowledge about the bicluster and the background in the form of pseudocounts. The parameters in $\boldsymbol{\alpha}$ and B are treated in the same way as the counts of the discrete expression levels in the data. A Dirichlet parameter vector $\boldsymbol{u} = [u_1, u_2, v_3]^{\mathrm{T}}$ can be decomposed into a term of counts l and a term of frequency $\boldsymbol{v}$

$$\boldsymbol{u} = l \cdot [v_1, v_2, v_3]^{\mathrm{T}} = l \cdot \boldsymbol{v}, \tag{5.36}$$

where $\sum_{i=1}^{3} v_i = 1$. The frequency term, $\boldsymbol{v}$ represents the frequencies of observing the three discrete levels; and the term of counts l is the amount of data from which $\boldsymbol{v}$ is observed. Thus, by changing l of a Dirichlet prior, we impose the strength of our prior knowledge on observing the frequency pattern $\boldsymbol{v}$.

5.6 Biclustering for global pattern discovery of pathologies

In this section, we show that our method can be used to discover global patterns embedded in the microarray data by using non-informative priors.

5.6.1 Construction of priors

Global pattern discovery means that we have no knowledge about the pathological classes of the tumors under study. This means that little prior knowledge should be imposed on the model of the bicluster. In this way, we allow the algorithm to discover natural patterns of biclusters that are embedded in the data. Therefore, we use weak priors (i.e., non-informative priors) for the bicluster, which means that by default, we set

$$\boldsymbol{\beta}_j = \frac{1}{\sqrt{n}} \cdot \frac{h(\mathcal{D})}{n \cdot m}, \quad j = 1 \ldots m. \tag{5.37}$$

Note that $\frac{h(\mathcal{D})}{n \cdot m}$ is the frequencies to observe the discrete levels in the entire data, and $\frac{1}{\sqrt{n}}$ is a factor which shrinks the value of $\boldsymbol{\beta}_j$ implying that we have little belief that the multinomial distribution of the bicluster resembles the frequency term $\frac{h(\mathcal{D})}{n \cdot m}$. (This factor $\frac{1}{\sqrt{n}}$ is chosen based on our experience after applying the algorithm on several data sets.)

On the other hand for the background, we assume that the frequency pattern discovered at the background resembles that of the entire data. Thus, we use

$$\boldsymbol{\alpha} = \sqrt{n} \cdot \frac{h(\mathcal{D})}{n \cdot m}. \tag{5.38}$$

Again, the factor $\sqrt{n}$ is chosen based on our experience. It implies our stronger belief that the multinomial model of the background resembles the frequency term $\frac{h(\mathcal{D})}{n \cdot m}$.

Though the above two equations provide a good starting point for the biclustering algorithm, we find that tuning the prior parameters $\boldsymbol{\alpha}$ and B by a coefficient (i.e., tuning on the l component of the Dirichlet prior vector in Equation 5.36) helps to direct the size and the consistency of the discovered pattern. Therefore, we open two parameters for user input—s^{α} and s^{β}—which tune $\boldsymbol{\alpha}$ and B respectively as follows,

$$\boldsymbol{\alpha} = s^{\alpha} \cdot \sqrt{n} \cdot \frac{h(\mathcal{D})}{n \cdot m}, \tag{5.39}$$

$$\boldsymbol{\beta}_j = s^{\beta} \cdot \frac{1}{\sqrt{n}} \cdot \frac{h(\mathcal{D})}{n \cdot m}, \quad j = 1 \ldots w. \tag{5.40}$$

5.6.2 Experiments on synthetic data

We use a synthetic data set to show how our algorithm works in practice and to illustrate the influence of the parameters.

Data

We embedded a pattern of 25 rows by 8 columns (see Figure 5.3(d)) into a data set of size 100 by 30 (see Figure 5.3(b)). The pattern was described by eight sharp multinomial distributions, while the background was generated from a multinomial distribution close to a uniform distribution,

$$\boldsymbol{\Phi}^{\text{true}} = \begin{bmatrix} 0.05 & 0.9 & 0.03 & 0.05 & 0.03 & 0.9 & 0.05 & 0.05 \\ 0.9 & 0.07 & 0.07 & 0.9 & 0.07 & 0.07 & 0.9 & 0.9 \\ 0.05 & 0.03 & 0.9 & 0.05 & 0.9 & 0.03 & 0.05 & 0.05 \end{bmatrix}, \tag{5.41}$$

$$\Psi^{\text{true}} = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix}. \tag{5.42}$$

The task for our algorithm is to find the location of the embedded bicluster (i.e., to give an estimate on $\mathcal{D}_R$ and $\mathbf{C}_m$).

Discovery of a bicluster

We ran the Gibbs sampling procedure (i.e., Loop 1 in Figure 4.7) for 500 iterations on the data set under the parameter settings $s^\alpha = 0.3$ and $s^\beta = 5$, and terminate the algorithm (so that Loop 3 in Figure 4.7 is not executed). We calculated the autocorrelation time (see Equation 4.19) for the samples of each of the Bernoulli parameters in the full conditoinal distributions

$$P(\mathcal{D}_R[i] = 1 \,|\, \mathcal{D}_R[\bar{i}]^{(t)}, \mathbf{C}_m^{(t)}, \mathcal{D}) \quad \text{for } i = 1 \ldots n,$$

and

$$P(C_j = 1 \,|\, \mathcal{D}_R^{(t)}, \mathbf{C}_{\bar{j}}^{(t)}, \mathcal{D}) \quad \text{for } j = 1 \ldots m.$$

In Figure 5.4, we use a density plot to represent the autocorrelation time of the 130 Bernoulli parameters under concern. As the plot shows, most of the Bernoulli parameters have an autocorrelation time smaller than 6. Combined with information from Figure 5.5 (a), which shows that the log-likelihood appear to converge after less than 10 iterations, we consider the Markov chains to have converged well after 50 iterations, and that with the 500 iterations in total, enough samples have been collected to carry out the Monte Carlo integration. Figure 5.5 (b) and (c) monitor the Bernoulli parameters of the full conditional distributions, which confirms our decision to use samples drawn from the last 450 iterations to simulate the posterior distributions of the labels.

Both the data and the result of the biclustering procedure are summarized in Figure 5.3. Figure 5.3(a) illustrates the posterior probability for each position in the data matrix that it belongs to the bicluster by a heatmap, $P(\mathcal{D}_R[i] = 1, C_j = 1 \,|\, \mathcal{D})$, which is obtained by

$$\begin{aligned} & P(\mathcal{D}_R[i] = 1, C_j = 1 \,|\, \mathcal{D}) \\ = & \sum_{t=1}^{T} P(\mathcal{D}_R[i] = 1 \,|\, \mathcal{D}_R[\bar{i}]^{(t)}, \mathbf{C}_m^{(t)}, \mathcal{D}) \cdot P(C_j = 1 \,|\, \mathcal{D}_R^{(t)}, \mathbf{C}_{\bar{j}}^{(t)}, \mathcal{D}). \end{aligned} \quad (5.43)$$

The posterior probability is reflected by the brightness associated to every position in the plot, where the two extremes, white and black, imply respectively the probabilities of 1 and 0. The inner bars around the main plot in Figure 5.3(a) indicate posterior probabilities $P(\mathcal{D}_R[i] = 1, | \mathcal{D})$ and $P(C_j = 1 \,|\, \mathcal{D})$, which can be calculated from Equation 4.14. The outer bars mark the embedded positions of the bicluster by a white tag. Figure 5.6 provides a further examination of the posterior probability that a row or a column belongs to the bicluster. It also shows that size of the final bicluster can also be adjusted by setting different thresholds on the posterior probabilities (in addition to adjusting s^α and s^β).

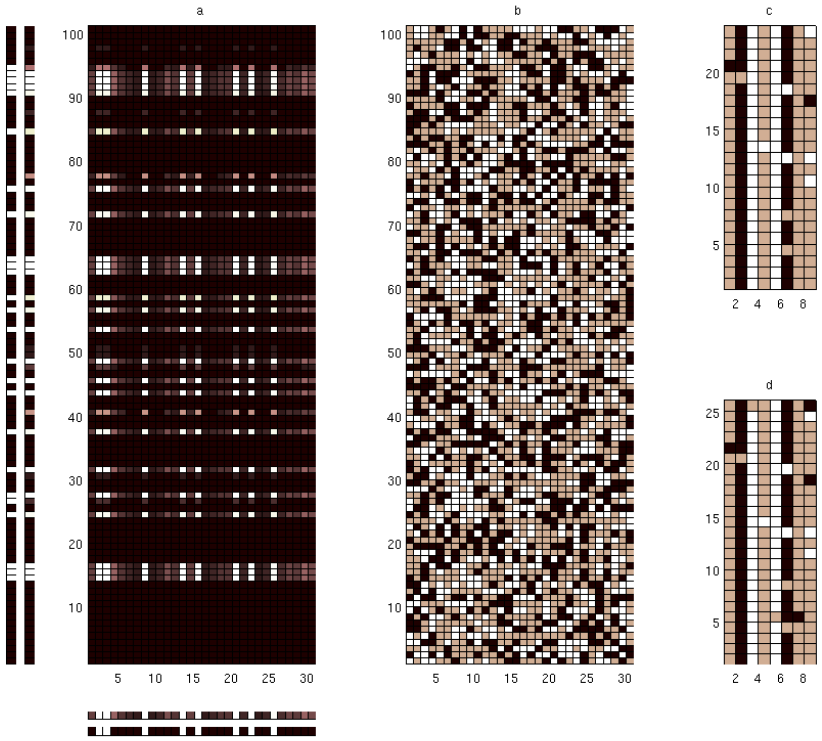


Figure 5.3: Results from the synthetic data set. (a) Main plot: The posterior probability that a position of the data matrix belongs to the bicluster. Inner bars: expected values of the random variables $\mathcal{D}_R$ and $\mathbf{C}_m$. Outer bars: positions of the embedded pattern. (b) The data matrix. (c) Pattern of the bicluster revealed by the Gibbs sampling algorithm. (d) Pattern of the embedded bicluster. All the positions where the bicluster is embedded have a high posterior probability to belong to the bicluster (see the inner bars of subplot (a)). We determine the position of the bicluster by putting a threshold on the posterior probability of the random variables $\mathcal{D}_R$ and $\mathbf{C}_m$. The pattern of the retrieved bicluster (subplot (c)) highly resembles that of the embedded bicluster (subplot (d)). All the columns where the embedded bicluster locates are identified, and two relatively noisy rows of the bicluster are left out of the discovered bicluster.

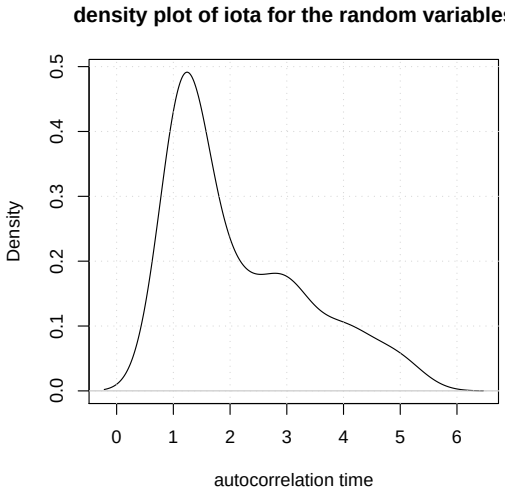


Figure 5.4: Density plot of the autocorrelation time of the Bernoulli parameters for the full conditional distribution of the 130 random variables under consideration (i.e., those that are included in $\mathcal{D}_R$ and $\mathbf{C}_m$). The figure shows that the autocorrelation time for most of the random variables is smaller than 6.

Figure 5.6 further illustrate the value of these posterior probabilities. In both of the plots in Figure 5.6, a seperation of the posterior probabilities at 0.5 is obvious. Therefore we use 0.5 as the threshold both for the rows $\mathcal{D}_R$ and the columns $\mathbf{C}_m$, and consider the positions of the target bicluster to be the ones that possess expected values higher than the thresholds in both dimensions. The final pattern of the bicluster revealed by our algorithm is shown in Figure 5.3(c), which resembles the embedded bicluster (see Figure 5.3(d)). Two rows of the embedded bicluster are not recovered, Row 26 and Row 94 of the data matrix, which can be explained by the fact that these two rows deviate most from the conserved pattern.

A more detailed look shows that there is quite variability in the configuration of the biclusters retrieved at different iterations. However, these biclusters overlapped with each other most frequently at the positions of our final decision, which is reflected by Figure 5.3 (a). (More illustrations will be provided in this respect in the following section.) 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.

Influence of s^α and s^β

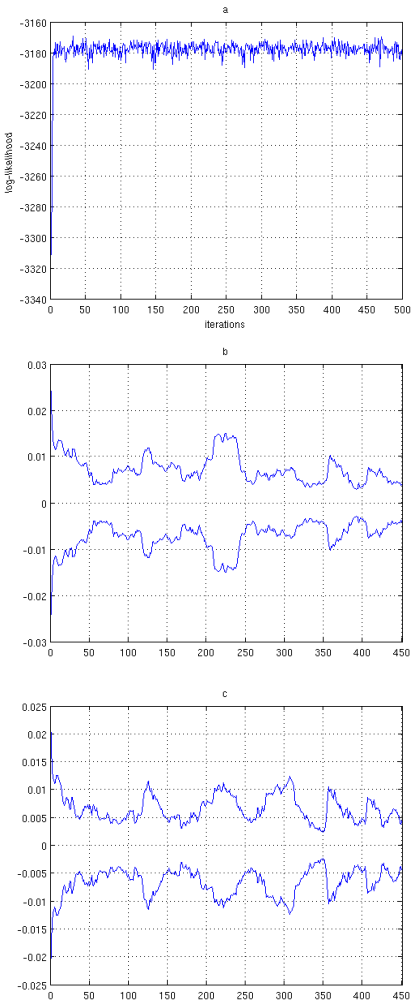


Figure 5.5: (a) Trace of the log likelihood of the synthetic data evaluated at the end of each iteration during the whole Gibbs sampling procedure. (b) and (c) reflects the evolution of the $P(\mathcal{D}_R[i] = 1 \mid \mathcal{D})$ and $P(C_j \mid \mathcal{D})$. For every random variable, we estimated $P(\mathcal{D}_R[i] = 1 \mid \mathcal{D})$ or $P(C_j \mid \mathcal{D})$ over every possible window of 50 iterations to obtain the trace of the posteriors (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 $\mathcal{D}_R$ or $\mathbf{C}_m$. Shown in (b) and (c) are the plus and minus one standard deviation. The plot shows a fast convergence of the log-likelihood and the posterior probabilities of the random variables—all of which seem to converge after less than 10 Gibbs sampling iterations.

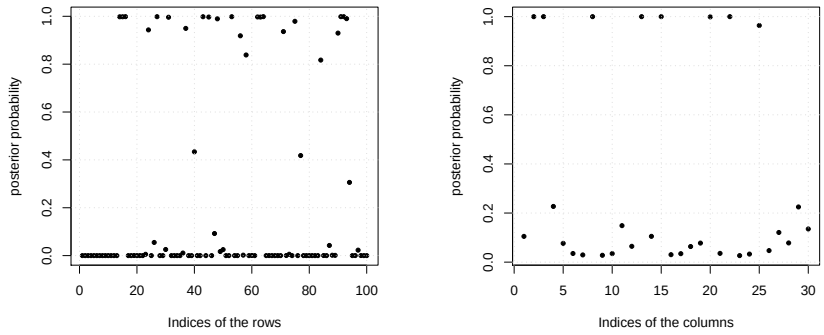
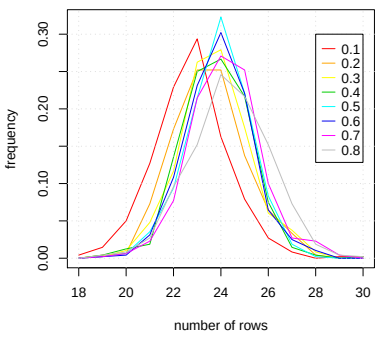


Figure 5.6: Plots of posterior probabilities of the random variables—left: $\mathcal{D}_R$, right: $\mathbf{C}_m$.

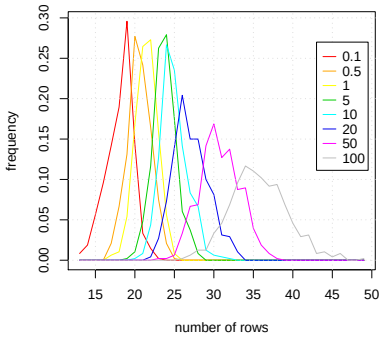
The user input parameters s^α and s^β (see Equation 5.39 and Equation 5.40) influence not only the size but also the diversity of the bicluster found at each iteration. To illustrate the impact of the two parameters, we performed the biclustering algorithm in two experiments with the following settings: (1) $s^\alpha = 0.3$ remains fixed, while s^β ranges over 0.1, 0.5, 1, 5, 10, 20, 50, and 100; (2) s^α ranges over 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, and 0.8, while $s^\beta = 5$ remains fixed.

Figure 5.7 illustrates the impact of s^α and s^β on the size of the bicluster found in each iteration. The figure shows that s^α has limited influence on the number of rows discovered in each bicluster (see Figure 5.7(a)). However, it has a high influence on the number of columns included in the bicluster. A bigger s^α puts a more stringent criterion on the selection of the columns, and as a result, fewer columns are included in the bicluster (see Figure 5.7(c)). When $s^\alpha = 0.1$, almost all the columns are selected for the bicluster in every iteration; when $s^\alpha = 0.9$ (results not shown here), the algorithm failed to converge because it happened frequently that no column was included in the bicluster. In contrast, s^β has little influence on the number of rows selected for the bicluster (see Figure 5.7(d)). Rather, it influences the number of columns selected for the bicluster in each iteration. The number of rows in the bicluster found for an iteration increases with the increment of s^β (see Figure 5.7(b)). This phenomenon is explained by Equation 5.40—a larger s^β imposes a stronger prior that the pattern of the bicluster resembles a uniform distribution, which is actually the distribution of the background.

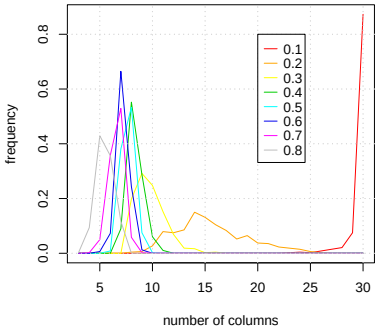
Figure 5.8 depicts the influence of s^α and s^β on the posterior probabilities that a row or a column may belong to the bicluster. Again, the figures show that s^α has limited influence on the posterior probabilities $P(\mathcal{D}_R \mid \mathcal{D})$ (see Figure 5.8(a)) and that s^β has little influence on $P(\mathbf{C}_m \mid \mathcal{D})$ (see Figure 5.8(d)). However, with



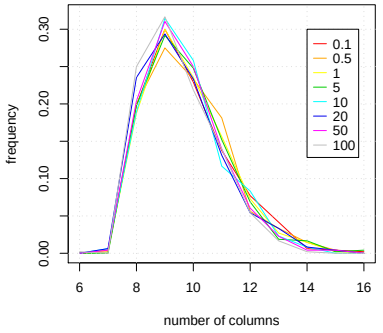
(a) Influence of s^{α} on the number of rows



(b) Influence of s^{β} on the number of rows

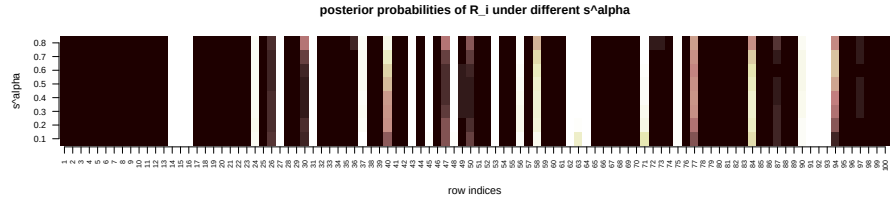


(c) Influence of s^{α} on the number of columns

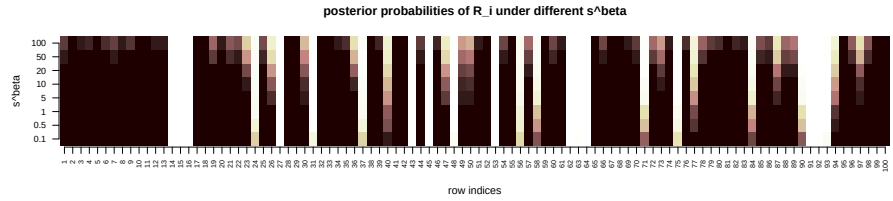


(d) Influence of s^{β} on the number of columns

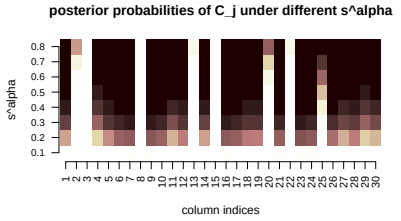
Figure 5.7: Frequency plots of the number of rows (or the number of columns) in the found bicluster of an iteration under different parameter settings. The differently color lines correspond to different parameter settings. The figure shows that s^{α} has limited influence on the number of rows discovered in each bicluster (see Subplot 5.7(a)). However, it has a high influence on the number of the columns included in the bicluster. A bigger s^{α} puts a more stringent criterion on the selection of the columns, and as a result, fewer columns are included in the bicluster (see Subplot 5.7(c)). On the other hand, s^{β} has little influence on the number of rows selected for the bicluster (see Subplot 5.7(d)). Rather, it influences the number of columns selected for the bicluster in each iteration. The number of rows in the bicluster found for an iteration increases with the increment of s^{β} (see Subplot 5.7(b)).



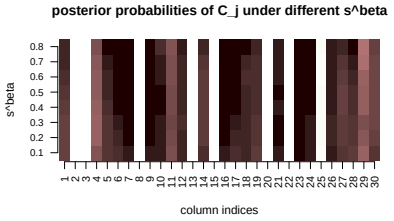
(a) Influence of s^{α} on the posterior of $\mathcal{D}_R$



(b) Influence of s^{β} on the posterior of $\mathcal{D}_R$



(c) Influence of s^{α} on the posterior of $\mathbf{C}_m$



(d) Influence of s^{β} on the posterior of $\mathbf{C}_m$

Figure 5.8: Heatmap illustrating the posterior probability that a row or a column in the data matrix belongs to the bicluster under different parameter settings. The brighter the spot in the heatmap, the larger the probability that the corresponding row or column (shown along the x-axis) in the data matrix may belong to the bicluster under the corresponding parameter setting (shown along the y-axis). The figures show that s^{α} has limited influence on the posterior probabilities $P(\mathcal{D}_R \mid \mathcal{D})$ (see subplot 5.8(a)) and that s^{β} has little influence on $P(\mathbf{C}_m \mid \mathcal{D})$ (see subplot 5.8(d)). However, with a smaller s^{α}, the posterior probability for a column to belong to the bicluster increases (see subplot 5.8(c)). Similarly, the probability for a row to be selected for a bicluster increases with s^{β} (see subplot 5.8(d)).

a smaller s^α, the posterior probability for a column to belong to the bicluster increases (see Figure 5.8(c)). (Note that when $s^\alpha = 0.1$, every the column in the data set almost has the posterior probability of 1 to be in the bicluster.) Similarly, the probability for a row to be selected for a bicluster increases with the increment of s^β (see Figure 5.8(d)).

Combining the information from Figure 5.7 and Figure 5.8, we can infer that when s^α is relatively large (i.e., when the discovered bicluster at the end of each iteration of the Gibbs sampling procedure contains a small number of columns), the selected columns for the bicluster are relatively consistent from one iteration to another, (or in other words, the diversity of the selected columns is relatively small). When s^α is small, the diversity of the selected columns at the end of each iteration is larger, but the columns selected in different iterations always overlap with each other most frequently at those selected columns when a larger s^α is used. The same inference can be made for s^β, but note that in the contrast, the increment of s^β raises the diversity of selected rows at each iteration. Therefore, by adjusting the input parameters s^α and s^β, users of the algorithm can fine-tune the stringency of the target bicluster.

5.6.3 Case study: biclustering experiments on leukemia patients

We applied our algorithm on a data set on leukemia patients, see [4] for a detailed description of the data. In this paper, Armstrong *et al.* show 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 explored the possibility to use our algorithm to find gene expression fingerprints of expression profiles for the three patient groups. The data set consists of expression data from Affymetrix chips (U95a or U95aV2) for 12,600 genes collected from 72 leukemia tumor samples (from 72 patients), of which 28 were clinically diagnosed as ALL, 20 as MLL, and 24 as AML.

We preprocess the data according to the original paper* [4]. First, a threshold of 100 and a ceiling of 1,600 were put on the original data to eliminate data points with noisy and non-reproducible low values and unreliable high values. Next, a variation filter was imposed so that 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 tumor samples. This reduced data set was then discretized according to the equal frequency principle as described in Section 5.2.

*We favored the preprocessing procedure described by the original paper than RMA or VSN (see Chapter 1), because both RMA and VSN result in a larger number of Affymetrix control probes that exhibit a big variation (among the top 15 percent for all the probes) in expression values across the tumor samples.

By masking the patients found after each run, the algorithm succeeded in discovering three biclusters one after another for the data set. The first bicluster selected 25 patients all of whom are (out of the 28) AML patients, and 444 genes (because of the large number of selected genes, Figure 5.9 illustrate the pattern of the discovered bicluster for 100 genes randomly chosen from the 444 genes). The second bicluster included 19 (out of 24) ALL patients, and 119 genes (see Figure 5.10). The third bicluster consisted of 17 (out of 20) MLL patients and 34 genes (see Figure 5.11). The patterns displayed in these figures demonstrate the ability of our algorithm to group patients based on their expression behavior over a subset of genes, and thus discover expression fingerprint for the patient groups.

These results are obtained by putting a threshold of 0.5 on the posterior probabilities of $\mathcal{D}_R$ and $\mathbf{C}_m$. For the patients, the posterior probabilities for $\mathcal{D}_R$ are quite polarized (being either very close to 1, or very close to 0). However, a different threshold on the posterior probabilities of $\mathbf{C}_m$ helps to fine-tune the selection of genes.

To test the significance of the found biclusters, we performed the algorithm on 100 permuted data sets of the test data. Tests were done under three sets of pseudocounts. 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 scattered around and did not have a consistent core. This result demonstrates that the patterns found by Gibbs biclustering are statistically highly significant.

We checked the genes whose difference in their expression levels for ALL patients and MLL patients is explained by their biological characters according to the paper of Armstrong *et al.* (2002), to see if these genes are also revealed by our algorithm. Table 5.1 provides a comparison between our discovery and the descriptions in the original paper [4] of those genes. The table shows that most of the genes mentioned in the original paper are recovered by at least one of the found biclusters. In general, the expression of these genes are in accordance with the description in Armstrong *et al.* (2002). However, judging from the results of our method on this data set, ALL patients have much more consistent patterns for most of the referred genes (than the MLL patients). For example, for the genes that have a function early B-cell development (i.e., MME, CD24, DNTT, TCF3, TCF4, POU2AF1, and LIG4) evidence revealed by our algorithm suggests that these genes are overexpressed in ALL, while the evidence that they are underexpressed in MLL is relatively weak. Similarly, for the genes that encode certain adhesion molecules (i.e., LGALS1, ANXA1, and CD44) our result provides more evidence that these genes are underexpressed for ALL patients than that they are overexpressed in MLL.

The paper of Armstrong *et al.* (2002) also compared gene expressions between

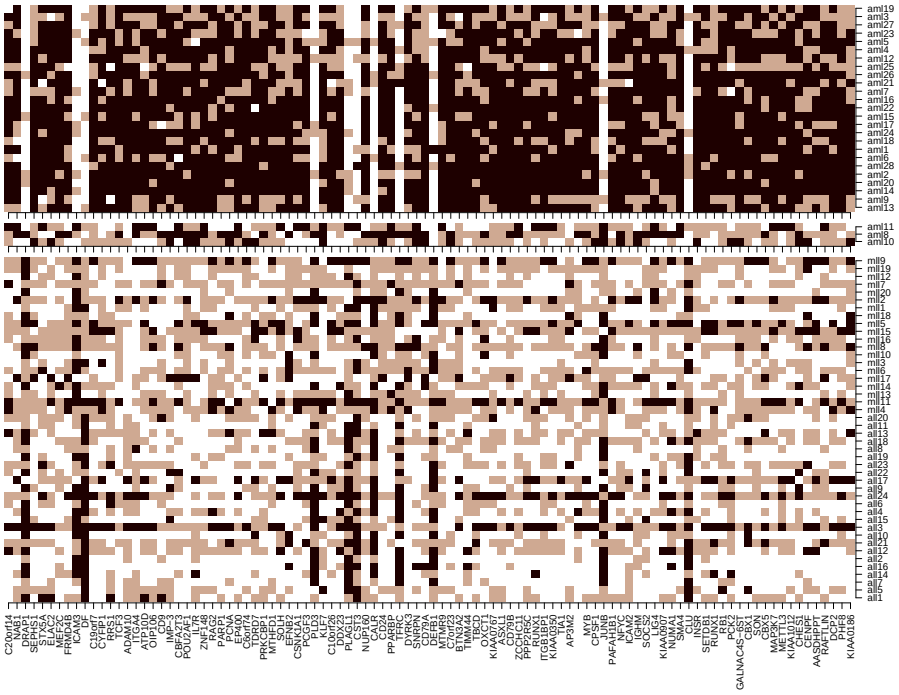


Figure 5.9: The first bicluster discovered consists of 25 out of 28 AML patients, whose discrete gene expression pattern is illustrated in the top figure as a heatmap, where the rows represent the patients, and the columns represent the genes (100 genes are randomly selected from the 444 genes that are included in the bicluster for the purpose of illustration). Black is used to represent the discrete level "low", gray for "medium", and white for "high". The heatmap in the middle of the figure shows the discrete expression pattern of the 3 left-out AML patients over the selected genes. The bottom heatmap shows the pattern of the ALL patients and the MLL patients over the selected genes, where the patients are reordered so that they are grouped according to their pathological categories.

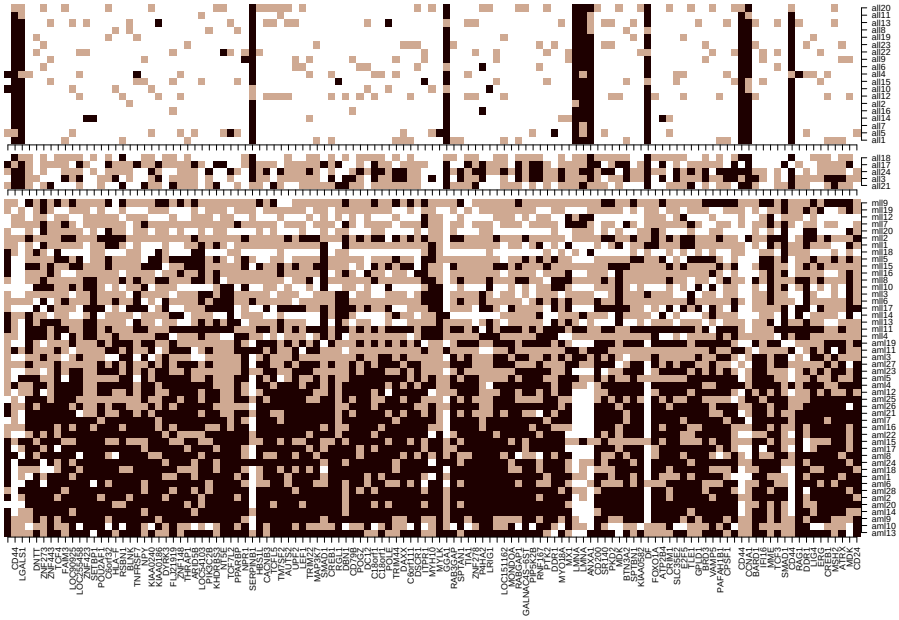


Figure 5.10: The found bicluster discovered consists of 19 out of 24 ALL patients, whose discrete gene expression pattern is illustrated in the top figure as a heatmap, where the rows represent the patients, and the columns represent the genes. Black is used to represent the discrete level ”low”, gray for ”medium”, and white for ”high”. The heatmap in the middle of the figure shows the discrete expression pattern of the 5 left-out ALL patients over the selected genes. The bottom heatmap shows the pattern of the AML patients and the MLL patients over the selected genes, where the patients are reordered so that they are grouped according to their pathological categories.

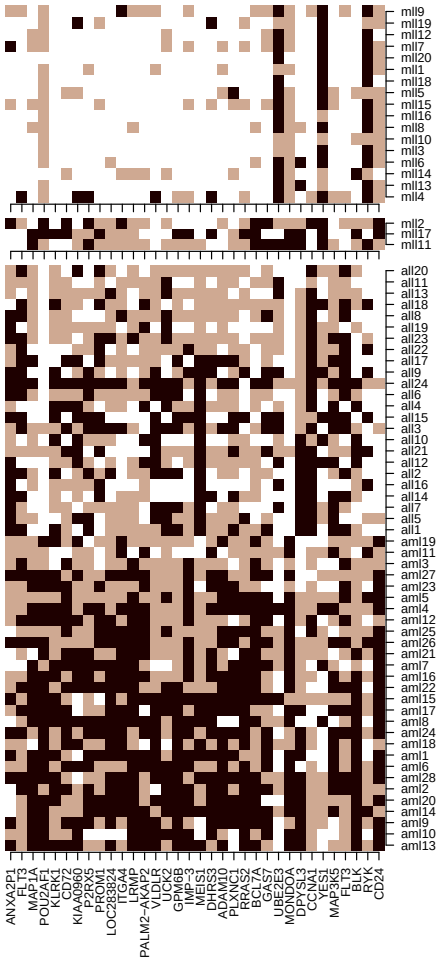


Figure 5.11: The third bicluster discovered consists of 17 out of 20 MLL patients, whose discrete gene expression pattern is illustrated in the top figure as a heatmap, where the rows represent the patients, and the columns represent the genes. Black is used to represent the discrete level ”low”, gray for ”medium”, and white for ”high”. The heatmap in the middle of the figure shows the discrete expression pattern of the 3 left-out MLL patients over the selected genes. The bottom heatmap shows the pattern of the AML patients and the ALL patients over the selected genes, where the patients are reordered so that they are grouped according to their pathological categories.

Gene	Armstrong *et al*. (2002)		Discovered biclusters[a]		
	Description	MLL vs. ALL	MLL	ALL	AML
MME	Genes expressed in early B cells	Under-expressed in MLL	–	high	low
CD24			medium	high	low
CD22			–	–	low
DNTT			–	high	low
TCF3	Genes required for appropriate B-cell development	Under-expressed in MLL	–	high	low
TCF4			–	high	–
POU2AF1			medium	high	low
LIG4			–	high	–
SMARCA4	Correlated with B-precursor ALL	Under-expressed in MLL	Gene filtered out by the variation filter		
LGALS1	Genes encoding certain adhesion molecules	Relatively over-expressed in MLL	–	low	–
ANXA1			–	low	–
ANXA2			–	–	–
CD44			–	low	–
SPN			Gene filtered by the variation filter		
PROML1	Genes expressed in progenitors	Highly over-expressed in MLL	high	–	–
FLT3			high	–	–
LMO2			Gene filtered out by the variation filter		
CCNA1	Myeloid-specific genes	highly over-expressed in MLL	high	low	–
SERPINB1			–	low	–
CAPG			–	–	–
RNASE3			–	–	–
NKG2D	Natural killer cell-associated gene	Highly over-expressed in MLL	Gene filtered out by the variation filter		
CD79B	Genes that mark early B-lymphoid commitment	MLL < ALL[b]	–	high	low
CD19			–	–	low
MME		not expressed in MLL	–	high	–
IL7R		MLL = ALL[c]	–	–	low

[a]The biclusters discovered by our algorithm are associated with the patient groups that they represent. Shown under this column are the discretized expression levels of the corresponding gene for the majority of patients included in the bicluster. A "–" means that the gene is not revealed by the corresponding bicluster.

[b]Expressed in MLL, though with lower levels than in ALL.

[c]Expressed at similar levels in ALL and MLL.

Table 5.1: Comparison between MLL expression pattern and ALL expression pattern for some biologically related genes discussed in Armstrong *et al.* (2002) according to both the original paper and the biclusters found by our algorithm.

Gene	Armstrong *et al.* (2002)		Discovered biclusters[a]		
	Description	High in[b]	MLL	ALL	AML
MME	Lymphoid-specific genes	ALL	–	high	–
CD24			medium	high	low
DNTT			–	high	low
LIG4			–	high	–
RPOML1	Hematopoietic progenitors	MLL	high	–	–
FLT3			high	–	–
LMO2			Gene filtered out by the variation filter		
DF	myeloid-specific genes	AML	–	low	high
CTSD			Gene filtered out by the variation filter		
ANPEP			–	–	high

[a]The biclusters discovered by our algorithm are associated with the patient groups that they represent. Shown under this column are the discretized expression levels of the corresponding gene for the majority of patients included in the bicluster. A "–" means that the gene is not revealed by the corresponding bicluster.

[b]High expression levels in the corresponding patient category according to Armstrong *et al.* (2002).

Table 5.2: Biclustering results on biologically relevant genes with characteristic expression patterns for one of the three pathological groups according to Armstrong *et al.* (2002).

the three types of leukemia sub-types—MLL, ALL, and AML. They reached a conclusion that conventional ALL samples express high levels of lymphoid-specific genes; AML samples express high levels of myeloid-specific genes; whereas MLL samples express high levels of genes associated with hematopoietic progenitors [4]. We also compared our results with their discovery in this regard. Our results confirmed the discovery of Armstrong *et al.*, see Table 5.2.

To further validate if the genes selected for the biclusters are leukemia related, we calculate the enrichment of the gene ontology (GO) terms [24] of biological processes based on a hypergeometric distribution (see Section 3.7). Putting a limit of 0.05 on the p-values of the GO terms, the 444 genes included in the bicluster that characterizes AML patients are over-represented in 162 GO terms of biological process. Figure 5.12 shows the hierarchical structure of these over-represented GO terms[b]. The majority of these over-represented terms are leukemia related, such as cell differentiation, regulation of cytokine, hemopoiesis, immune response, chemotaxis, monocyte activation, neutrophil activation, regulation of DNA recombination, and somatic cell DNA recombi-

[b]The GO graphs in this thesis are generated by BioConductor [41] package "GOstats" and R [98] package "Rgraphviz".

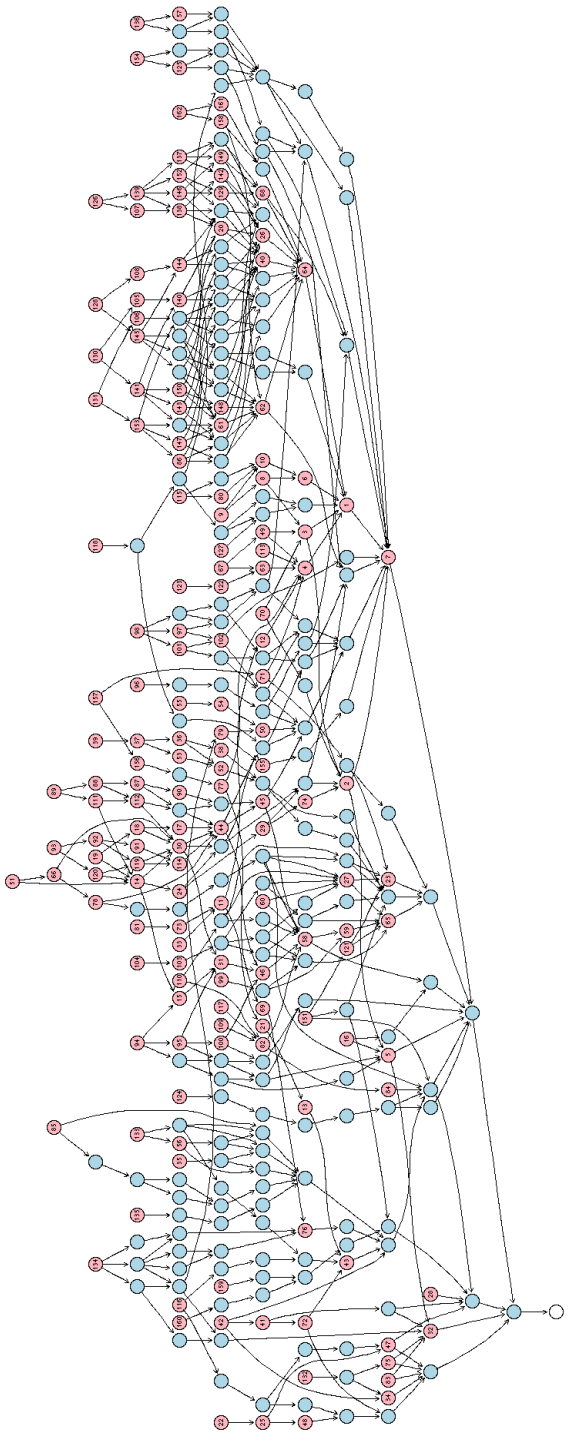


Figure 5.12: GO graph for over-represented GO terms (pink nodes) of biological process for the genes that are included in the bicluster that characterize AML patients.

rank[a]	GO term	p-value	k[b]	f[c]
9	regulation of transcription, DNA-dependent	7.49e-05	73	1025
14	positive regulation of cytokine biosynthesis	1.16e-03	4	11
16	regulation of vascular permeability	2.15e-03	2	2
19	positive regulation of tumor necrosis factor-alpha biosynthesis	2.15e-03	2	2
22	hemocyte development	3.02e-03	3	7
25	hemocyte differentiation (sensu Arthropoda)	4.66e-03	3	8
28	membrane fusion	6.71e-03	4	17
33	chromatin modification	8.68e-03	7	51
35	vesicle targeting	1.21e-02	2	4
39	induction of positive chemotaxis	1.21e-02	2	4
42	cell growth	1.33e-02	9	82
47	cell differentiation	1.48e-02	14	158
48	hemopoiesis	1.71e-02	7	58
51	positive regulation of interleukin-2 biosynthesis	1.96e-02	2	5
55	protein amino acid methylation	1.96e-02	2	5
60	humoral immune response	2.26e-02	12	135
63	DNA recombination	2.49e-02	6	49
67	regulation of DNA recombination	2.85e-02	2	6
70	negative regulation of nucleobase, nucleoside, nucleotide and nucleic acid metabolism	2.98e-02	9	94
73	chromatin assembly or disassembly	3.49e-02	7	67
81	nucleosome assembly	3.98e-02	5	41
83	regulation of gene expression, epigenetic	4.16e-02	3	17
85	intracellular copper ion transport	4.64e-02	1	1
87	fractalkine biosynthesis	4.64e-02	1	1
89	positive regulation of fractalkine biosynthesis	4.64e-02	1	1
93	positive regulation of interleukin-1 beta biosynthesis	4.64e-02	1	1

[a]The rank of significance of the GO term (biological process ontology), based on its p-value, among all the significantly enriched GO terms.

[b]Number of LocusLink IDs, corresponding to genes in the bicluster that are annotated with the specified GO term (biological process ontology).

[c]Number of LocusLink IDs corresponding to probes on the chip of Affymetrix U95A V2 that are annotated with the specific GO term (biological process ontology).

Table 5.3: Some GO terms (from the biological process ontology) that are highly related to leukemia that are associated with genes selected for the bicluster corresponding to AML patients.

rank[a]	GO term	p-value	k^b	f^c
94	positive regulation of cytokine secretion	4.64e-02	1	1
96	constitutive protein ectodomain proteolysis	4.64e-02	1	1
98	chondroitin sulfate biosynthesis	4.64e-02	1	1
99	defense response to Gram-negative bacteria	4.64e-02	1	1
104	detection of triacylated bacterial lipoprotein	4.64e-02	1	1
109	monocyte activation	4.64e-02	1	1
110	neutrophil activation	4.64e-02	1	1
113	DNA integration	4.64e-02	1	1
115	transcription termination from Pol II promoter	4.64e-02	1	1
116	paranodal junction formation	4.64e-02	1	1
117	microglial cell activation	4.64e-02	1	1
118	alanyl-tRNA aminoacylation	4.64e-02	1	1
123	generation of antibody gene diversity	4.64e-02	1	1
124	release of cytochrome c from mitochondria	4.64e-02	1	1
126	UDP catabolism	4.64e-02	1	1
127	RNA methylation	4.64e-02	1	1
128	UMP biosynthesis	4.64e-02	1	1
130	dTDP biosynthesis	4.64e-02	1	1
131	dTTP biosynthesis	4.64e-02	1	1
132	regulation of body size	4.64e-02	1	1
133	Golgi to endosome transport	4.64e-02	1	1
134	negative regulation of cyclin dependent protein kinase activity	4.64e-02	1	1
135	UDP-N-acetylgalactosamine transport	4.64e-02	1	1
136	glutamine catabolism	4.64e-02	1	1
151	cellular response to starvation	4.64e-02	1	1
154	histidine biosynthesis	4.64e-02	1	1
157	mismatch repair	4.82e-02	3	18
159	calcium-mediated signaling	5.00e-02	2	8
160	androgen receptor signaling pathway	5.00e-02	2	8
162	cGMP biosynthesis	5.00e-02	2	8

[a]The rank of significance of the GO term (biological process ontology), based on its p-value, among all the significantly enriched GO terms.

[b]Number of LocusLink IDs, corresponding to genes in the bicluster that are annotated with the specified GO term (biological process ontology).

[c]Number of LocusLink IDs corresponding to probes on the chip of Affymetrix U95A V2 that are annotated with the specific GO term (biological process ontology).

Table 5.4: Some GO terms (from the biological process ontology) that are highly related to leukemia that are associated with genes selected for the bicluster corresponding to AML patients (continued).

nation. Table 5.3 and 5.4 list those terms that are represented by the leaves of the GO graph. Some of the branches of the GO graph are extended a bit deeper in Table 5.3 and Table 5.4 to illustrate their relevance to leukemia.

The majority of over-represented GO terms of biological processes for those genes that are included in the bicluster characterizing ALL patients are highly related to ALL (in this case we put the limit on p-values < 0.1). Figure 5.13 illustrates the hierarchical structure of these over-represented GO terms. Table 5.5 and Table 5.6 list the terms on the leaves of the GO graph, as well as the two GO terms with the lowest p-value—B-cell differentiation and hemopoiesis—because of their close relation with ALL and leukemia. Together, Figure 5.13, Table 5.5 and Table 5.6 show that the over-represented GO terms, which include B-cell differentiation, hemopoiesis, immune response, transcription regulation, B-cell activation, lymphocyte differentiation and apoptosis, evidently portray an ALL theme.

Although the 34 selected genes for the bicluster representing MLL patients exhibit a strong pattern on the 17 MLL patients included in the bicluster, the majority of the GO terms of biological processes for these genes are not relevant to leukemia.

To conclude, the genes recovered by our algorithm for each of the three biclusters provide a strong expressional fingerprint for the patients selected for the corresponding bicluster. Furthermore, the genes that are included in the biclusters representing AML and (especially) ALL patients have a meaningful interpretation of pathology. However, pathological evidence is relatively weak for the genes that are selected for the bicluster representing MLL patients. The difference in the selected differentially expressed genes between our results and those represented in Armstrong *et al.* (2002) [4] might because of the discretization procedure and consequently the different criterion for judging differentially expressed genes.

In general, the experiments on this data set illustrate the ability of the algorithm to discover biclusters of consistent gene behaviors over subsets of patients, which are embedded in a microarray data set. Moreover, the biclusters discovered by the algorithm confirm the possibility to use data produced by microarray technology to fingerprint gene expression patterns to facilitate pathological discoveries in leukemia.

5.7 Query-driven biclustering for pattern discovery in pathology

As we mentioned before, the task of query-driven biclustering of experiments is the following. Given a set of patients (usually in a small amount), or tumor samples, which we know to share the same pathology (we refer to this set

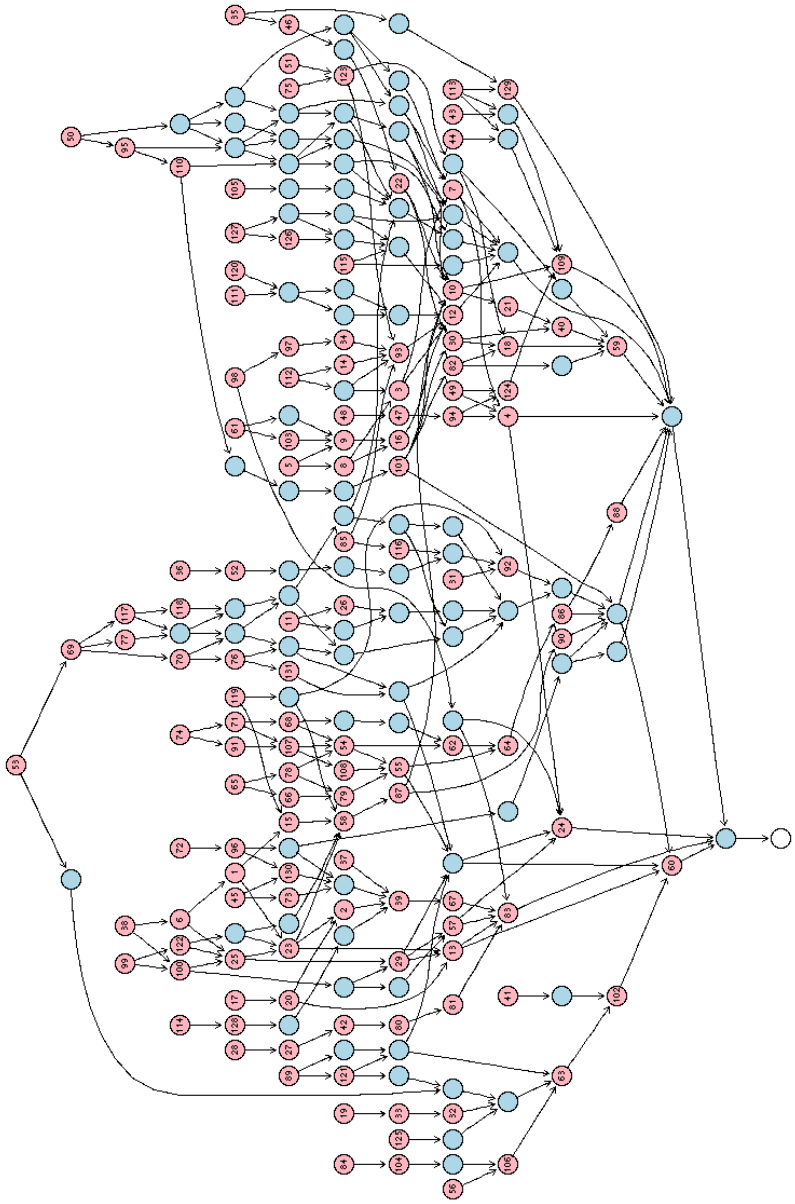


Figure 5.13: GO graph for over-represented GO terms (pink nodes) of biological process for the genes that are included in the bicluster that characterize ALL patients.

rank[a]	GO term	p-value	k[b]	f[c]
1	B-cell differentiation	3.72e-4	3	13
2	hemopoiesis	4.77e-4	5	58
5	regulation of transcription, DNA-dependent	7.47e-4	23	1025
11	calciumion transport	1.61e-3	4	45
14	DNA recombination	2.22e-3	4	49
17	hemocyte development	2.58e-3	2	7
19	androgen receptor signaling pathway	3.41e-3	2	8
28	spermatogenesis	7.63e-3	4	69
31	cytokinesis	8.85e-3	4	72
35	immunoglobulin secretion	1.13e-2	1	1
36	male meiosis I	1.13e-2	1	1
37	adrenal gland development	1.13e-2	1	1
38	positive regulation of B-cell differentiation	1.13e-2	1	1
41	cell-matrix adhesion	1.51e-2	3	46
43	diuresis	2.26e-2	1	2
44	natriuresis	2.26e-2	1	2
45	regulation of dendrite morphogenesis	2.26e-2	1	2
48	regulation of vasodilation	2.26e-2	1	2
49	regulation of vascular permeability	2.26e-2	1	2
50	positive regulation of interferon-gamma biosynthesis	2.26e-2	1	2
51	postreplication repair	2.26e-2	1	2
53	release of cytoplasmic sequestered NF-kappaB	2.26e-2	1	2
56	Wnt receptor signaling pathway	2.43e-2	3	55
61	transcription initiation from Pol II promoter	2.99e-2	2	24
65	induction of apoptosis	3.11e-2	4	105
72	regulation of neuronal synaptic plasticity	3.37e-2	1	3
74	inhibition of caspase activation	3.37e-2	1	3
75	single strand break repair	3.37e-2	1	3
84	transforming growth factor beta receptor signaling pathway	3.98e-2	2	28
85	DNA replication	4.03e-2	4	114
98	DNA methylation	5.55e-2	1	5
99	positive regulation of T-cell differentiation	5.55e-2	1	5
105	protein ubiquitination	5.99e-2	4	130

[a]The rank of significance of the GO term (biological process ontology), based on its p-value, among all the significantly enriched GO terms.

[b]Number of LocusLink ID's, corresponding the genes in the bicluster, which are annotated with the GO term of biological process.

[c]Number of LocusLink ID's corresponding to the probes on the chip of Affymetrix U95A V2 that are annotated with the GO term of biological process.

Table 5.5: Some Go terms of biological processes that are highly related to leukemia that are associated with genes selected for the bicluster corresponding to ALL patients.

rank[a]	GO term	p-value	k[b]	f[c]
111	mRNA cleavage	6.63e-2	1	6
112	regulation of DNA recombination	6.63e-2	1	6
113	fluid secretion	6.63e-2	1	6
114	negative regulation of angiogenesis	6.63e-2	1	6
115	nucleotide catabolism	6.63e-2	1	6
119	B-cell proliferation	7.69e-2	1	7
120	mRNA polyadenylylation	7.69e-2	1	7
125	calcium-mediated signaling	8.74e-2	1	8
127	cGMP biosynthesis	8.74e-2	1	8

[a]The rank of significance of the GO term (biological process ontology), based on its p-value, among all the significantly enriched GO terms.

[b]Number of LocusLink ID's, corresponding to the genes in the bicluster that are annotated with the GO term of biological process.

[c]Number of LocusLink ID's corresponding to the probes on the chip of Affymetrix U95A V2 that are annotated with the GO term of biological process.

Table 5.6: Some Go terms of biological processes that are highly related to leukemia that are associated with genes selected for the bicluster corresponding to ALL patients (continued).

of patients or tumor samples hereafter as the seeds), we want to query the microarray data to recruit other patients (or tumor samples) that belong to the same pathological group and in the mean time to identify a gene expressional fingerprint for the pathology. This type of tools is useful for biologists and doctors to retrieve information from microarray data when only a small number of (tumor) samples can be confirmed by traditional means to belong to the specific pathological type of interest.

5.7.1 Construction of priors

To incorporate information from the seeds into the Bayesian framework of our biclustering algorithm, we use the frequency information from the seeds to construct the prior of the bicluster, and impose it to the bicluster model as a soft query, so that the Gibbs sampling procedure is directed to the discovery of the target pathological type. In the meantime, the soft query means that seed patients (or seed tumor samples) that do not share the common pattern for the discovered bicluster will be excluded.

More specifically, given a set of seed patients (or seed tumor samples) whose indices in the data matrix are collected in $\mathbf{a}$, we calculate the frequencies of the three discrete levels observed on the seed patients for gene j, and use it as the base for $\boldsymbol{\beta}_j$:

$$\boldsymbol{\beta}_j = s^{\beta} \cdot \frac{h(\mathcal{D}[\mathbf{a}, j]) + 0.001}{n}. \tag{5.44}$$

Note that 0.001 is a pseudocount added to the counts to avoid zero frequency.

However, for the background model, we keep the parameter setting as described in Section 5.6.1, assuming that the background model is similar to the frequency model observed in the whole data set, i.e.

$$\boldsymbol{\alpha} = \sqrt{n} \cdot \frac{h(\mathcal{D})}{n \cdot m}. \tag{5.45}$$

In addition, to accelerate the convergence of the Gibbs sampling procedure, we initialize the algorithm by assigning all the seed patients (seed rows in the data matrix) to the bicluster and assigning the rest of the patients to the background.

$$\mathcal{D}_R[i] = \begin{cases} 1, & i \in \mathbf{a} \\ 0, & i = 1 \ldots n,\ i \notin \mathbf{a}. \end{cases} \tag{5.46}$$

However, for the columns of the data matrix $\mathbf{C}_m$ (i.e., for the genes), we initialize by randomly assigning them to either the bicluster or the background.

5.7.2 Experiments on synthetic data

We embedded three biclusters to a noisy background described by a distribution close to the uniform distribution. The data set contains 200 rows and 40 columns. Bicluster 1 is 40 by 7 in size, Bicluster 2 is 25 by 10, and Bicluster 3 is 35 by 8. The consistency of the data in the three biclusters (i.e., the sharpness of the multinomial distributions that generate the data in the biclusters) also differs from one to another. The resulting data is shown in Figure 5.14, where the rows and the columns of the data set are rearranged to manifest the biclusters.

We first performed biclustering for global pattern discovery on the data as described in Section 5.6. We used five different sets of parameters, under each of which the biclustering algorithm was performed 10 times, to see which embedded bicluster is first recovered each time (i.e., Loop 3 in Figure 4.7 is not executed). As illustrated in Table 5.7, Bicluster 3 remains as a dominant pattern for the biclustering discovery.

Parameters	Bicluster 1	Bicluster 2	Bicluster 3
$s^{\text{bcl}} = 7, s^{\text{bgd}} = 0.2$	0	0	10
$s^{\text{bcl}} = 7, s^{\text{bgd}} = 0.4$	0	2	8
$s^{\text{bcl}} = 5, s^{\text{bgd}} = 0.3$	1	0	9
$s^{\text{bcl}} = 3, s^{\text{bgd}} = 0.2$	0	0	10
$s^{\text{bcl}} = 3, s^{\text{bgd}} = 0.4$	2	2	6

Table 5.7: Number of times each bicluster are discovered by the biclustering algorithm under five different parameter settings.

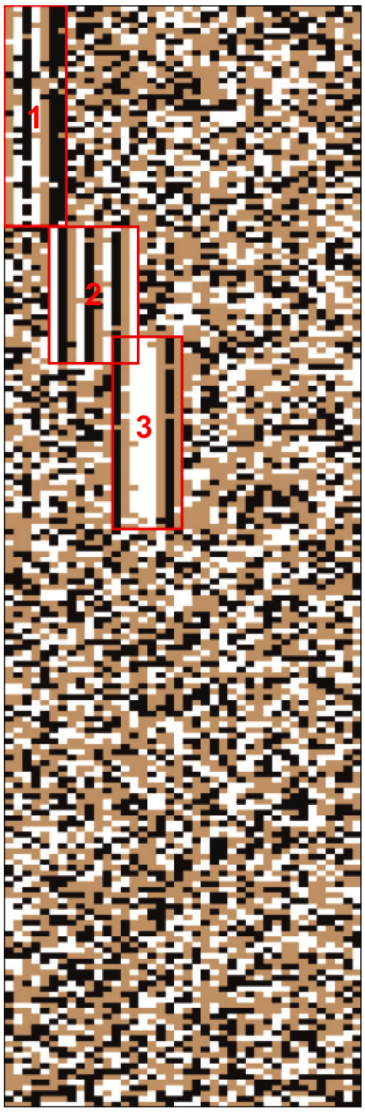


Figure 5.14: Synthetic data set with three biclusters of different size and consistency are embedded. The rows and the columns of the data set are reordered to clearly show the biclusters.

s^β	Seed 1			Seed 2			Seed 3		
	Bcl. 1	Bcl. 2	Bcl. 3	Bcl. 1	Bcl. 2	Bcl. 3	Bcl. 1	Bcl. 2	Bcl. 3
0.1	16	4	0	12	3	5	11	6	3
0.5	16	3	1	13	2	5	10	5	5
1	19	1	0	13	2	5	10	4	6
5	16	4	0	16	0	4	5	0	15[a]
10	20	0	0	18	0	2	0	0	20[a]
20	20	0	0	18	0	2	-	-	-
50	20	0	0	19	0	1	-	-	-
100	20	0	0	20	0	0	-	-	-

[a]Degenerate bicluster that has a major overlap with Bicluster 3.

Table 5.8: Number of times (out of 20) that each embedded bicluster is recovered under different values of s^β for each set of seeds.

We now explore the ability of the algorithm to discover the two non-dominant biclusters—Bicluster 1 and Bicluster 2—when a set of seed rows is imposed to the algorithm as a query to direct the discovery. As a first experiment, we examine the performance of the algorithm under different sets of seeds extracted from Bicluster 1, where the consistency of the seed rows in each set varies from one to another. To construct the different sets of seeds, we calculated the frequency that each discrete levels (i.e., 1, 2, or 3) is observed under each column of Bicluster 1. This frequency model is concluded as $\mathbf{\Phi}^{\text{true}}$,

$$\Phi_j^{\text{true}} = \frac{h\{\mathcal{D}[\mathbf{r}^{\text{true}}, \delta_{40}(j)]\} + 0.001}{200}, \quad j \in \mathbf{c}^{\text{true}}, \tag{5.47}$$

where $\mathbf{r}^{\text{true}}$ and $\mathbf{c}^{\text{true}}$ are the row and column indices indicating where Bicluster 1 is embedded in the data. Similarly, we calculated the frequency model of the background, Ψ^{true}, using the background data. Then, each row is scanned to obtain a similarity score (between the row and the embedded bicluster) which is essentially a likelihood ratio,

$$d[i] = \prod_{j \in \mathbf{c}^{\text{true}}} \left(\frac{\mathbf{\Phi}^{\text{true}}}{\Psi^{\text{true}}} \right)^{h\{\mathcal{D}[i,j]\}}, \quad i = 1, \ldots, 200. \tag{5.48}$$

The scores of the rows in the whole data set is plotted in Figure 5.15. We ranked the rows according to their scores, and composed three sets of seeds accordingly: Seed 1 consists of 5 rows of the highest similarity scores, Seed 2 is composed of 5 rows ranked in the middle of the 200 rows in the data set, and Seed 3 contains 5 genes with the lowest similarity scores. The data of the seeds are illustrated as heatmaps in Figure 5.16.

For each set of seeds, we first test the frequency with which Bicluster 1 is found back as the first bicluster discovered by the algorithm under different parameter settings. We found that the influence of s^α remains the same as explained

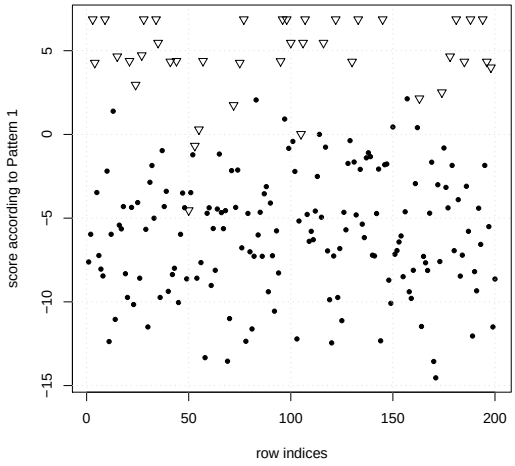
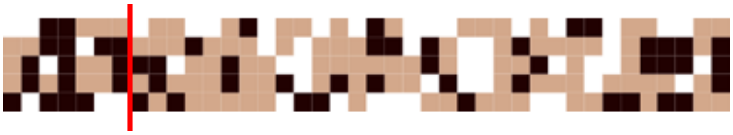


Figure 5.15: Scores of the rows in the whole data set measuring their similarity with the embedded Bicluster 1. The triangles represent the rows where Bicluster 1 is embedded, and the dots represent the rest of the rows in the data set.

(a) Seed 1 for Bicluster 1

(b) Seed 2 for Bicluster 1



(c) Seed 3 for Bicluster 1

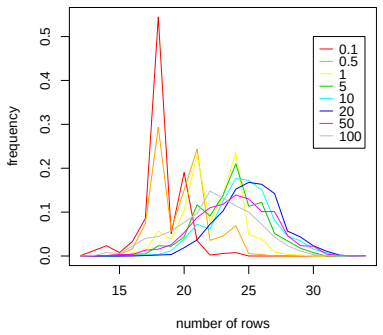
Figure 5.16: Patterns of the three seeds that are used to guide the discovery of Bicluster 1. The rows in each pattern represent the rows in the data set that are used as seeds. The column in each pattern is rearranged so that the first seven columns corresponds to those where Bicluster 1 is embedded.

in Section 5.6.2. That is, a larger s^{α} results in fewer columns selected for the bicluster, and data under the selected columns exhibit higher consistency for those rows that are included in the bicluster. Therefore, we focus our discussion below on the influence of s^{β}. To illustrate the influence of s^{β}, we fixed $s^{\alpha} = 0.3$, and performed the algorithm under 8 different values of s^{β}. For each set of seeds and under the same parameter settings, we ran the algorithm 20 times, each time only to identify one cluster in the bicluster. Out of these 20 runs, we count the number of times that Bicluster 1 is recovered. The result is presented in Table 5.8.

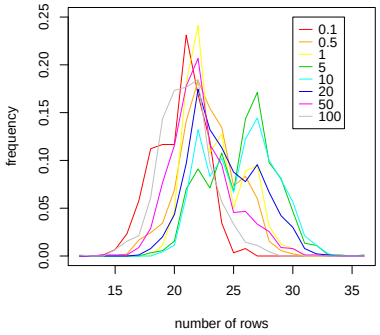
Table 5.8 shows that in general the introduction of seeds increases the chance to discover the target bicluster (i.e., Bicluster 1). When the set of seed rows compose a consensus pattern under the columns of the true bicluster (as in the case of Seed 1), and with a large s^{β} (e.g., $s^{\beta} \geq 5$ in this case), the intended bicluster is almost always found back. When the pattern of the seed rows exhibits high consistency under the true biclustering columns, (e.g., Seed 1 and Seed 2), the frequency that the algorithm retrieves the embedded pattern decreases together with of s^{β}. The decrement in the consistency of seed rows reduces the frequency of finding the intended pattern. When the set of seeds carries little information about the bicluster—as in the case of Seed 3, where the similarity scores (see Equation 5.48) of the seed rows (to Bicluster 1) are no higher than the maximum score of the rest of the rows (see Figure 5.15)—a smaller s^{β}, however, helps to recover the embedded bicluster. That is because a big s^{β} only emphasizes the noise of the seeds, and thus suggests that the target bicluster is also a noisy one. This explanation is also reflected by the fact that a degenerated version of Bicluster 3 (a bicluster consisting of some 50 rows and 10 columns, with the rows and columns greatly overlapping with those in Bicluster 3) is found when a large s^{β} (i.e., larger than 10) is used.

We also examined the influence of s^{β} on the number of rows selected for the bicluster in each iteration of the Gibbs sampling procedure in those cases when the target bicluster (i.e., Bicuster 1) is recovered. Figure 5.17 shows the frequency that, the bicluster found in an iteration contains a certain number of rows. For the two extreme situations (i.e., Seed 1 and Seed 3) the increment of s^{β} causes the bicluster discovered during the whole Gibbs sampling procedure to cover a larger range of row size. However, in the case of Seed 2, the range for the number of rows discovered by the algorithm increased first and then fell back as s^{β} grows larger. Despite their individual difference, the bicluster discovered during the whole Gibbs sampling procedure always overlap at the position of the target bicluster (i.e., Bicluster 1). Changing the threshold on the final estimate of $\mathcal{D}_R$ (see Equation 4.53) helps to fine-tune the final selection of rows.

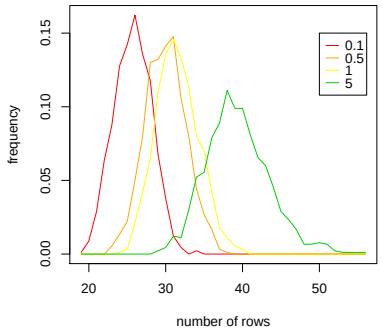
Note that the algorithm can exclude seeds that are not consistent with the discovered bicluster. For instance, when Seed 3 is used, those rows often have very low posterior probabilities to be included in the final bicluster.



(a) Seed 1



(b) Seed 2



(c) Seed 3

Figure 5.17: Frequency plots of the number of rows contained in the bicluster found at the end of an iteration in the Gibbs sampling procedure, for different set of seeds, and under the 8 different values of s^β. For the two extreme situations (i.e., Seed 1 and Seed 3) increasing s^β causes the bicluster discovered during the whole Gibbs sampling procedure to cover a larger range of row sizes. However, in the case of Seed 2, the range for the number of rows discovered by the algorithm increased first and then fell back as s^β grows larger.

# Unique rows in the seed	Bicluster 1	Bicluster 2	Bicluster 3
0	9	0	11
1	16	0	4
2	15	0	5
3	17	0	3
4	20	0	0
5	20	0	0

Table 5.9: Number of times (out of 20) that each embedded bicluster is discovered when a query is imposed to discover Bicluster 2. The seed is made up of two components, one of those rows where only Bicluster 2 is embedded, and the other of those rows where Bicluster 2 overlaps with Bicluster 3. Each set of seeds consists of five rows. The first row in the table shows the number of rows in the seed where Bicluster 2 is uniquely embedded.

In the second experiment, we examine the impact of seed rows that overlap with the dominant bicluster in the data set. Our target bicluster in this case is Bicluster 2. We set the parameters $s^{\alpha} = 0.2$, and $s^{\beta} = 10$, and use five rows in each set of seeds. Each set of seed rows is composed of two parts, one of those rows where only Bicluster 2 is embedded, and the other of those rows where Bicluster 2 overlaps with Bicluster 3. However the proportion of the two components is different from one set of seeds to another. As Table 5.9 illustrates, the increase in the proportion of unique rows in the seed raises the chance to find the intended bicluster.

5.7.3 Case study: query-driven biclustering of leukemia patients

The paper of Yeoh *et al.* (2002) [115] demonstrates that distinct expression profiles can identify each of the prognostically important leukemia subtypes of pediatric acute lymphoblastic leukemia (ALL), including T-ALL, E2A-PBX1, BCR-ABL, TEL-AML1, MLL rearrangement, and hyperdiploid > 50 chromosomes. In addition, they found a novel ALL subgroup based on its unique expression profile.

The data of Yeoh *et al.* (2002) [115] contains gene expression profiles measured on 327 patients, of whom 15 are BCR-ABL patients, 27 are E2A-PBX1 patients, 64 are hyperdiploid-over-50-chromosomes patients, 20 are MLL-rearrangement patients, 43 are T-ALL patients, 79 are TEL-AML patients, and 79 are from other pathological categories. We preprocessed the data as described in the original paper [115]. Further, we only retained the genes, whose variations was are among the 15% highest for further analysis, which leaves 1894 genes in the data set. The resulting data was then discretized as described in Section 5.2.

Patient group	1st bicl.	2nd bicl.	3rd bicl.	4th bicl.	5th bicl.
TEL-AML1	4/20	3/11	1/7	0/7	0/5
T-ALL	6/20	2/11	2/7	0/7	0/5
Hyperdiploid > 50	1/20	1/11	3/7	2/7	0/5
E2A-PBX1	0/20	1/11	1/7	1/7	0/5
E2A-PBX1 + MLL	0/20	0/11	0/7	2/7	0/5
Last 1/3 patients[a]	9/20	3/11	0/7	0/7	0/5
First 1/3 patients[b]	0/20	1/11	0/7	2/7	5/5

[a]Selected patients are represented in the last 1/3 columns in the data matrix.
[b]Selected patients are represented in the first 1/3 columns in the data matrix.

Table 5.10: The frequency of recovering biclusters representing different patient groups by performing the biclustering algorithm to the data using non-informative priors. The columns of the table correspond to the biclusters found sequentially by performing the biclustering experiment. In total 20 experiments were performed, and the masking-and-biclustering procedure was terminated when a bicluster not corresponding to any of the patient groups is found. The denominators gives the number of times that the algorithm succeeded to recover a i^{th} bicluster during the 20 experiments (where i refers to the number in the title of the column). The numerators are the number of times that the bicluster corresponding to a certain patient type is recovered.

We first applied the biclustering algorithm (as described in Section 5.6) to discover global patterns in the data, to get an idea of the behavior of the pattern discovery under non-informative prior on the data. Clusters found in earlier rounds are masked to discover multiple patterns. The whole procedure is terminated when a biclusters not in correspondence to any of the patient group is found, as illustrated in Figure 4.7. This entire procedure was repeated 20 times in total. Out of these 20 rounds, we counted the number of times that a bicluster was recovered. Table 5.10 summarizes the frequency of discovering a bicluster representing a certain patient group. The table shows that biclusters whose columns depend on their position in the data set (which is a result from the fact that the genes included in these bicluster do not show sufficient variations in their continuous expression profiles, see Figure 5.2 for an explanation) are more dominant than the biclusters representing the MLL group, the BCR-ABL group, and the novel patient group.

For each of the ALL pathological subgroups that are not recovered in the above biclustering experiments (i.e., the MLL group, the BCR-ABL group, the novel group of patients) we randomly selected five patients as seeds, and applied the query-driven biclustering to the unmasked data. Our algorithm successfully revealed biclusters corresponding to the targeted patient classes. The bicluster presented in Figure 5.18 includes 20 patients, 18 of whom are clinically identified as MLL patients (while the other two belong to the hyperdiploid >

50 group), and 57 genes, fingerprinting the gene expression profiles of the 20 patients. Moreover, the 57 genes are highly related to MLL according to both Yeoh *et al.* (2002) [115] and Armstrong *et al.* (2002) [4]. Figure 5.19 illustrates the bicluster where 12 out of the 14 patients of the novel group are included. This bicluster includes 84 genes, which have a strong fingerprint over the 12 selected patients. For the BCR-ABL patients, the algorithm found a bicluster of 9 BCR-ABL patients and 71 genes, 17 out of which overlaps with those that are discovered by Yeoh *et al.* (2002), see Figure 5.20.

This example shows the power of applying informative prior on the hierarchical Bayesian model for the biclustering problem of experiments. If such prior knowledge is available, the ability of the algorithm to find non-dominant patterns embedded in the microarray data can be greatly enhanced.

5.8 Conclusion

In this chapter, we give a full explanation of the Bayesian hierarchical model for the problem of biclustering experiments. We discussed two ways of constructing the prior distributions either to discover respectively global biclusters embedded in the data or to direct the discovery toward a bicluster corresponding to a certain type of experiment. By using two synthetic data sets, we illustrate the influence of user input parameters on the resulting bicluster. In addition, we also illustrate the usefulness of algorithm on two leukemia data sets.

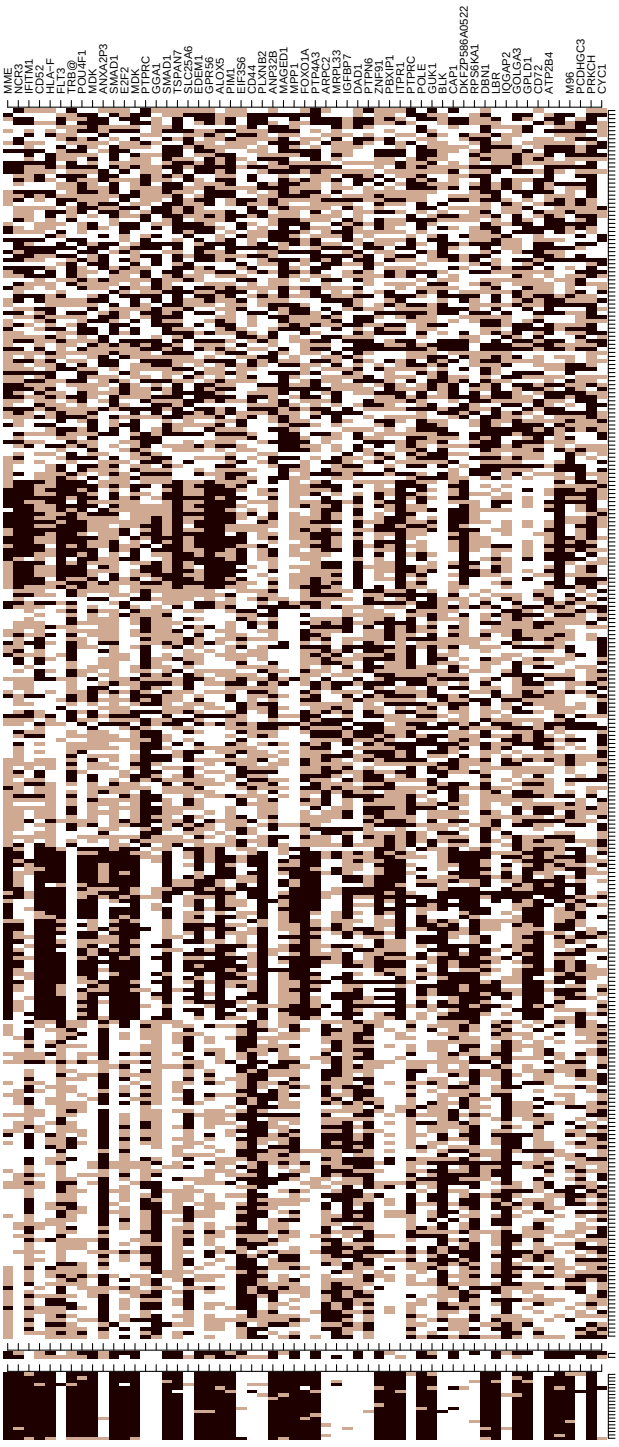


Figure 5.18: The left-most heatmap shows the pattern of the bicluster, which includes 18 MLL patients and 2 patients from the hyperdiploid > 50 group (represented by the columns) and 57 genes (represented by the rows). Black represents the discrete expression level of "low", brown for "middle", and white for "high". The heatmap in the middle of the figure shows the discrete expression pattern of the two left-out MLL patients over the selected genes. The right-most heatmap shows the pattern of the rest of the patients over the selected genes, where the patients are reordered so that they are grouped according to their pathological categories. The names of the 57 genes are also provided.

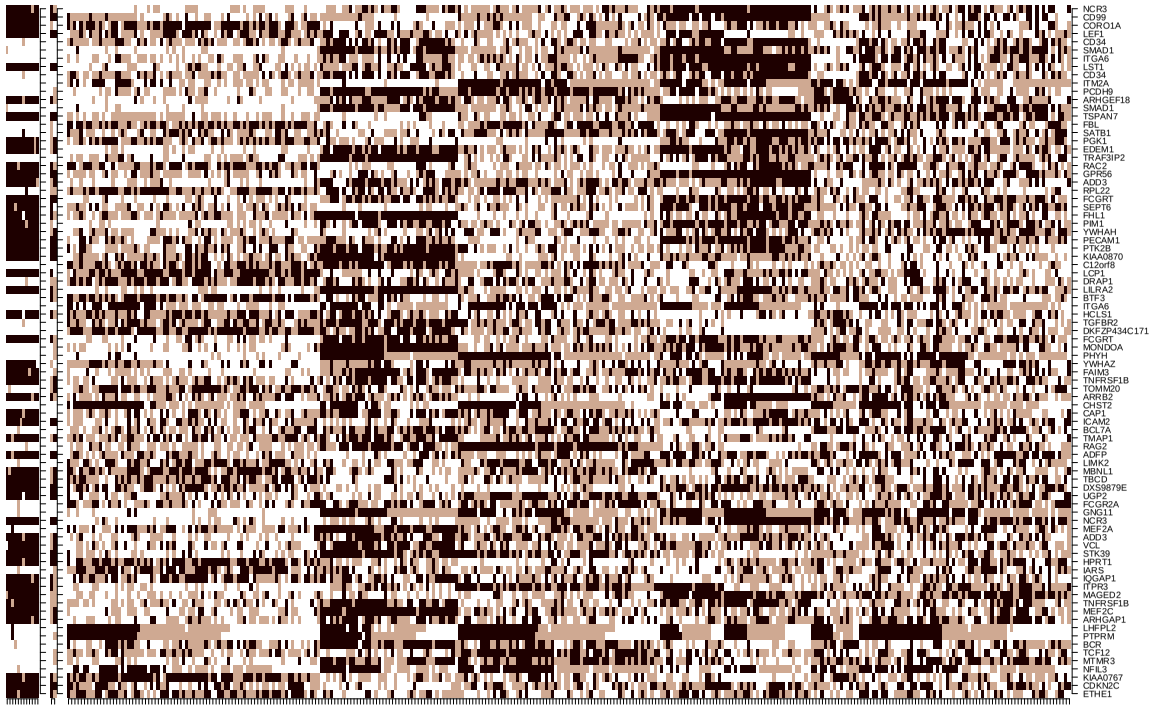


Figure 5.19: The left-most heatmap shows the pattern of the bicluster which, includes 12 patients of the novel group (represented by the columns) and 84 genes (represented by the rows). Black represents the discrete expression level of ”low”, brown for ”middle”, and white for ”high”. The heatmap in the middle of the figure shows the discrete expression pattern of the two left-out patients of the novel group over the selected genes. The right-most heatmap shows the pattern of the rest of the patients over the selected genes, where the patients are reordered so that they are grouped according to their pathological categories. The names of the 84 genes are also provided.

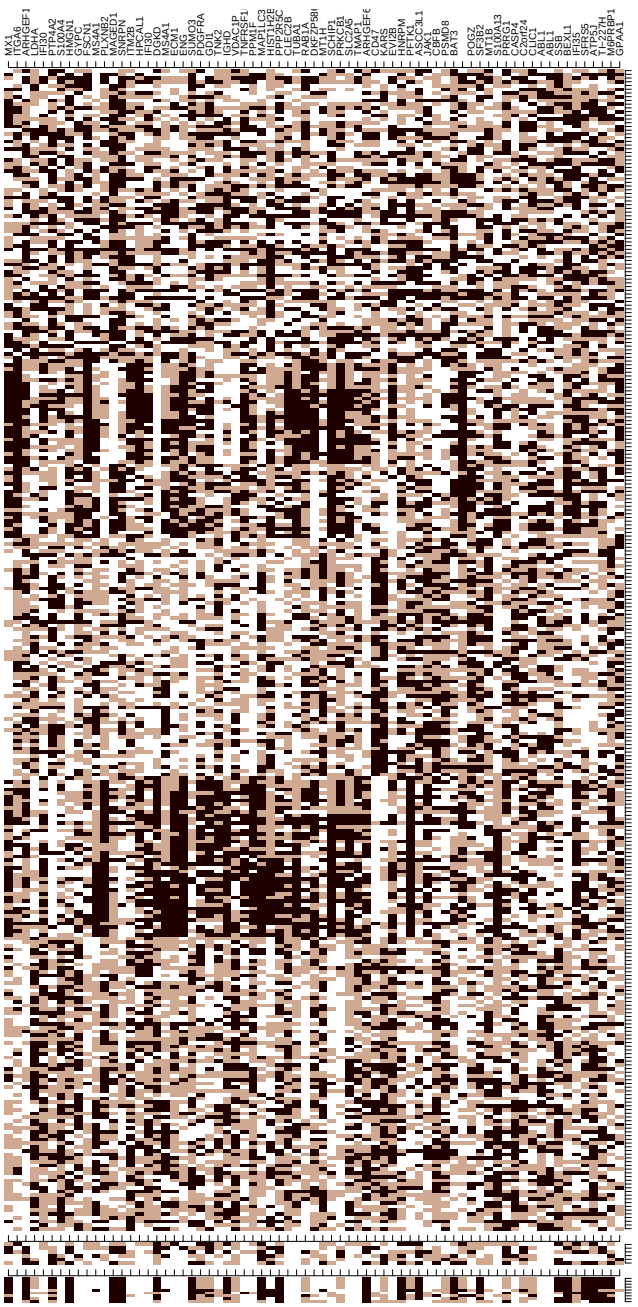


Figure 5.20: The left-most heatmap shows the pattern of the bicluster, which includes 10 BCR-ABL patients (represented by the columns) and 71 genes (represented by the rows). Black represents the discrete expression level of "low", brown for "middle", and white for "high". The heatmap in the middle of the figure shows the discrete expression pattern of the 5 left-out BCR-ABL over the selected genes. The right-most heatmap shows the pattern of the rest of the patients over the selected genes, where the patients are reordered so that they are grouped according to their pathological categories. The names of the 15 genes are also provided.

Chapter 6

Biclustering genes in microarray data

In this chapter, we develop a dedicated Bayesian hierarchical model for the problem of biclustering genes. We emphasize the importance of priors in this type of model, and their usefulness in biological discovery. We illustrate the efficiency of our algorithm in assisting the discovery of regulatory transcriptional modules on a combined data set of yeast.

6.1 Introduction

The same Gibbs sampling strategy as described in Chapter 4 can be used for the biclustering of genes. However, in contrast to the biclustering of experiments, we treat the preprocessed microarray as it is—i.e., no transposition is performed before carrying out the Gibbs biclustering procedure. Because now the row dimension of the data is at least of thousands of rows, a Gaussian distribution becomes a feasible candidate for describing the data. Fitting a normal model to gene expression measurements under a certain condition is considered reasonable especially when an appropriate preprocessing procedure has been applied to the microarry data.

The nature of our biclustering algorithm partitions the genes in the microarray data into two components based on their expression profiles under a selected subset of conditions. The partition is made by associating the genes either to the bicluster model or to the background model. Because of the high complexity of the underlying biological processes, a probabilistic mixture model for a microarray data set often comprises a vast number of modes. In our case, we divide the probabilistic mixture into a bicluster component and a background

component, which means that each combination of the modes in the mixture model can be selected as a bicluster (and consequently the rest of the modes combined as the background).

As described later on in the chapter, the same hierarchical model structure is used for both the bicluster and the background. This means that when a non-informative prior is used, the Gibbs sampling procedcure tends to partition the genes corresponding to the two global modes in the maximum likelihood function. Genes included in either of the partitions are often of little interest to biologists.

The strength of our algorithm lies in its ability to answer this type of specific questions that interest biologists. Thanks to the Bayesian hierarchical model that we use, the question of interest for the biologists, once transformed to mathematical language, can be imposed to the model to direct the discovery of the bicluster. By introducing a prior, methods based on Bayesian models help to zoom in on the local area of interest of the likelihood landscape, and raise the corresponding area in the posterior distribution.

We consider one possible situation where the biologists have at hand a specific set of genes (called the "seed genes" hereafter), which they know to be related to some common biological function. The question for their query to the microarray data is "which other genes in this data set share similar expression profiles as the seed genes and thus might be involved in the same function? In the meantime, in which experimental conditions is this biological function involved?" Otherwise stated, given the seed genes, we want to recruit genes (presented in the microarray data set) that share similar expression profiles under a subset of conditions. In addition, the few seed genes whose profiles are not compatible with the discovered pattern should be rejected if present. We discuss in this chapter the methodology to construct a prior model by using the seed genes. When the prior model is strong enough, the posterior mean estimate of the target joint posterior distribution (see Equation 4.42) provides an answer to the query of biologists.

In this chapter, we discuss the Gibbs sampling strategy for tackling the biclustering problem of genes in the following four aspects:

- *Model structure*: minor modifications in the general structure to improve results on time-series experiments
- *Gauss-Wishart model*: the hierarchical Bayesian models describing the bicluster and the background
- *Full conditional distributions*: the distributions from which samples of the hidden variable R, the structural variables $\mathbf{C}$, and the model parameters are drawn during the Gibbs sampling procedure
- *Construction of the priors*: the incorporation of prior information into the Bayesian hierarchical model

For a case study, we illustrate the ability of our biclustering strategy to assist the discovery of regulatory modules in transcriptional networks. We first consider the possible layout of the biclusters in a microarray data set that could correspond to different transcriptional modules from a biological point of view. Then, we test the performance of our algorithm on a synthetic data set in which the discussed layout of biclusters are embedded. Finally, we use a yeast data set to illustrate our strategy for transcriptional regulatory module discovery as a whole.

6.2 Model structure

For this chapter, we distinguish between the phrases "experiment" and "condition" by defining an experiment as a column of the microarray data matrix, and a condition as a group of experiments. This distinction is useful, for example, when the microarray data is obtained from time-series experiments. In this case, different columns in a microarray data set may correspond to experiments that are performed under the same condition but at different time points. When performing the biclustering algorithm, we might want to assign all the experiments from the same condition to one bicluster by using one hidden variable to describe the association of these experiments to the bicluster. Yet in the meantime, to allow flexibility in the model, we would use different distributions to describe different experiments.

We use $\{X_j \,|\, j = 1, \ldots, m\}$ to denote random variables that describe the expression values of the genes under the corresponding experiment (i.e., corresponding column in the data set). We introduce another set of variables $\{\mathbf{Y}_k \,|\, k = 1 \ldots q\}$ (where q is the number of the conditions) to denote the expression of the genes under condition k. Using $\mathbf{e}_k$ to denote the set of indices of the experiments that belong to Condition k, we have,

$$\mathbf{Y}_k = \{X_l \,|\, l \in \mathbf{e}_k\}. \tag{6.1}$$

For example, in Figure 6.1,

$$\mathbf{e}_4 = [5, 6], \tag{6.2}$$

$$\mathbf{Y}_4 = \{X_5, X_6\}. \tag{6.3}$$

Adjusting our Bayesian hierarchical model introduced in Section 4.3 for this change, the nodes for observed data now represent $\mathbf{Y}_1, \ldots, \mathbf{Y}_q$ (see Figure 6.1 for an example). Consequently, the notation for the set of Bernoulli variables to describe whether a node belongs to the bicluster changes to $\mathbf{C}_q = \{C_k \,|\, k = 1, \ldots, q\}$. To be explicit, the bicluster model now is

$$p(\mathbf{Y}_k) = \begin{cases} p(\mathbf{Y}_k \,|\, R = 1) = f(\mathbf{\Theta}_k^{\text{bcl}}) & C_k = 1,\ R = 1 \\ p(\mathbf{Y}_k \,|\, R = 0) = f(\mathbf{\Theta}_k^{\text{bgd}}) & C_k = 1,\ R = 0 \\ p(\mathbf{Y}_k) = f(\mathbf{\Theta}_k^{\text{bgd}}) & C_k = 0 \end{cases}, \quad k = 1 \ldots q, \tag{6.4}$$

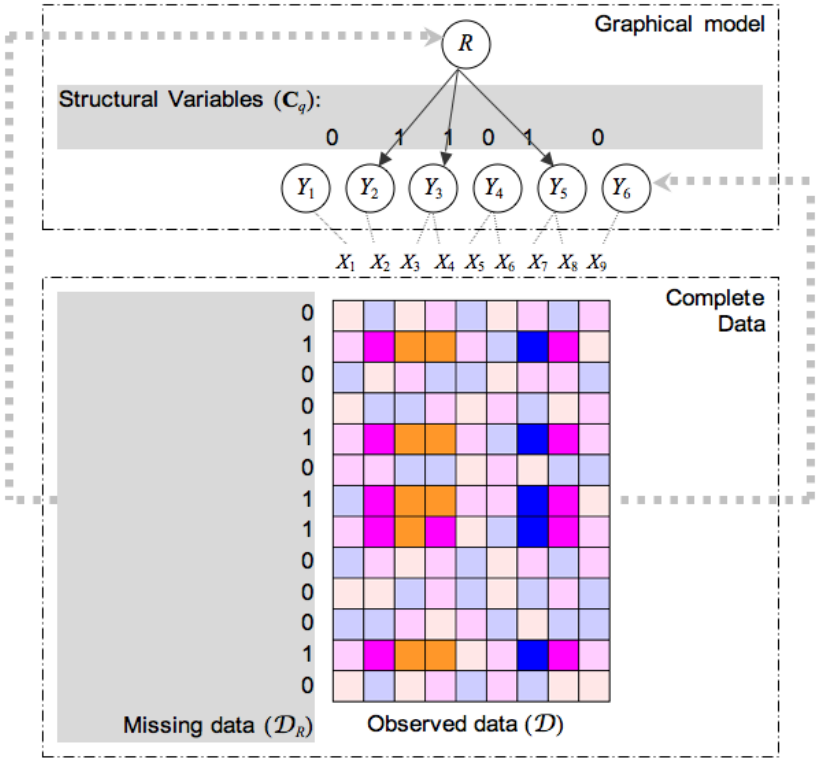


Figure 6.1: Data level of a Bayesian model for biclustering genes. The colored matrix represents microarray data, where an embedded bicluster is highlighted. Each column in the data represents an experiment, which is described by random variable X_i, ($i = 1, \ldots, 9$). The experiments performed under the same condition are grouped together. The conditions are represented by random variables $\{\mathbf{Y}_k \,|\, k = 1 \ldots 6\}$. Random variables $\mathbf{Y}_k$, ($k = 1, \ldots, 6$) are involved in the graphical model for biclustering as shown in the upper part of the figure.

where a $\mathbf{\Theta}_k$ parameterizes the multivariate distribution $p(\mathbf{Y}_k) = p(\mathbf{X}_{\mathbf{e}_k})$.

To facilitate the notation, we keep the symbols $\mathbf{r}$, $\bar{\mathbf{r}}$, $\mathbf{c}$ and $\bar{\mathbf{c}}$ as described in Equations 5.14 to 5.17. More specifically, $\mathbf{r}$ and $\bar{\mathbf{r}}$ are respectively the indices of the rows (i.e., genes) that belong to the bicluster and the background; $\mathbf{c}$ and $\bar{\mathbf{c}}$, however, now denote the indices of the conditions (instead of the columns in the data matrix, or the experiments) that belong to the bicluster and the background respectively. For the experiments, we use

$$\mathbf{e} = [j \mid j \in \mathbf{e}_k, \forall k = \{1, \ldots, q\} \wedge C_k = 1] \tag{6.5}$$

to denote the entire set of indices of the experiments (i.e., columns in the data) whose corresponding conditions belong to the bicluster, and

$$\bar{\mathbf{e}} = [j \mid j \in \mathbf{e}_k, \forall k = \{1, \ldots, q\} \wedge C_k = 0] \tag{6.6}$$

to notate the entire set of indices of the experiments whose corresponding condition are in the background. Take Figure 6.1 for example,

$$\mathbf{e} = [2, 3, 4, 7, 8], \tag{6.7}$$
$$\bar{\mathbf{e}} = [1, 5, 6, 9]. \tag{6.8}$$

6.3 The Gauss-Wishart model

We use Gaussian distributions to describe the expression data. This choice is not only inspired because normal models are analytically convenient, but also because of the previous success in applying normal mixture models to the clustering problems of microarray data [116, 71]. Furthermore, as we have mentioned in the beginning of this chapter, the assumption for fitting a normal distribution to the gene expression measurements in a given situation is considered to be reasonable especially when a proper preprocessing procedure has been applied to the microarray data [7].

For each $\mathbf{Y}_k$ ($k \in \mathbf{c}$) we assume that the covariance matrix of $\mathbf{X}_{\mathbf{e}_k}$ is diagonal. That is to say that we use a single normal distribution to model the expression values of the genes that belong to the bicluster under each experiment,

$$X_j \sim \mathcal{N}\left(\mu_j^{\text{bcl}}, (\sigma_j^{\text{bcl}})^2\right), \quad j \in \mathbf{e}. \tag{6.9}$$

To provide flexibility to the model, we allow μ_j (for $j \in \mathbf{e}$) to be drawn from different distributions. Because of the use of conjugate priors, the μ_j's follow normal distributions,

$$\mu_j^{\text{bcl}} \sim \mathcal{N}\left(\varphi_j^{\text{bcl}}, (\tau_j^{\text{bcl}})^2\right), \quad j \in \mathbf{e}. \tag{6.10}$$

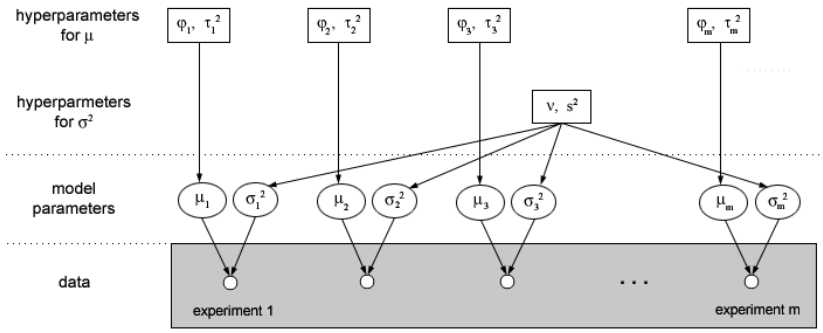


Figure 6.2: The same hierarchal model structure used for describing both the bicluster and the background for the problem of biclustering genes.

However, for the variance of the experiments in the bicluster, we assume that the variances of different experiments in the bicluster are modeled by the same prior distribution. Again, conjugate priors are used,

$$(\sigma_j^{\text{bcl}})^2 \sim \text{Inverse-}\chi^2\left(\nu^{\text{bcl}}, (s^{\text{bcl}})^2\right), \quad j \in \mathbf{e}. \tag{6.11}$$

In this way, we assume that both the correlations within $\mathbf{Y}_k$ (i.e., correlations between the experiments in the same condition, X_j for $j \in \mathbf{e}_k$) as well as the correlation between the $\mathbf{Y}_k$'s (for $k = 1 \ldots q$) are explained by the prior on σ_j^2 for $j \in \mathbf{e}$.

The hierarchical model structure for experiments in the bicluster is illustrated in Figure 6.2. Note that the hierarchical model for the bicluster as explained above (see Equation 6.9, Equation 6.11, and Equation 6.10) is represented as a whole by $\mathbf{\Theta}^{\text{bcl}}$ in Equation 6.4.

We use the same hierarchical structure for the background (as illustrated in Figure 6.2),

$$X_j \sim \mathcal{N}\left(\mu_j^{\text{bgd}}, (\sigma_j^{\text{bgd}})^2\right) \quad j = 1 \ldots m, \tag{6.12}$$

$$\mu_j^{\text{bgd}} \sim \mathcal{N}\left(\varphi_j^{\text{bgd}}, (\tau_j^{\text{bgd}})^2\right) \quad j = 1 \ldots m, \tag{6.13}$$

$$(\sigma_j^{\text{bgd}})^2 \sim \text{Inverse-}\chi^2\left(\nu^{\text{bgd}}, (s^{\text{bgd}})^2\right) \quad j = 1 \ldots m. \tag{6.14}$$

Equations 6.13 to 6.14 contains the hierarchical model for data in the background represented as $\mathbf{\Theta}^{\text{bgd}}$ in Equation 6.4.

We will also use

$$\boldsymbol{\mu}^{\text{bcl}} = \{\mu_j^{\text{bcl}} \mid j \in \mathbf{e}\} \tag{6.15}$$

$$\boldsymbol{\mu}^{\text{bgd}} = \{\mu_j^{\text{bgd}} \mid j = 1 \ldots m\}. \tag{6.16}$$

The same system of notations is applied to σ^{bcl}, σ^{bgd}, φ^{bcl}, φ^{bgd}, τ^{bcl} and τ^{bgd}.

6.4 Full conditional distributions

As explained in Section 4.4.2 (see Equation 4.42) the target joint distribution for this biclustering problem is

$$p(\mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}} \,|\, \mathcal{D}, \xi^{\text{bcl}}, \xi^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}),$$

where

$$\begin{aligned}
\boldsymbol{\Theta}^{\text{bcl}} &= \{\boldsymbol{\mu}^{\text{bcl}}, (\boldsymbol{\sigma}^{\text{bcl}})^2\} && (6.17)\\
\boldsymbol{\Theta}^{\text{bgd}} &= \{\boldsymbol{\mu}^{\text{bgd}}, (\boldsymbol{\sigma}^{\text{bgd}})^2\} && (6.18)\\
\xi^{\text{bcl}} &= \{\boldsymbol{\varphi}^{\text{bcl}}, (\boldsymbol{\tau}^{\text{bcl}})^2, \nu^{\text{bcl}}, (s^{\text{bcl}})^2\} && (6.19)\\
\xi^{\text{bgd}} &= \{\boldsymbol{\varphi}^{\text{bgd}}, (\boldsymbol{\tau}^{\text{bgd}})^2, \nu^{\text{bgd}}, (s^{\text{bgd}})^2\}. && (6.20)
\end{aligned}$$

This means that in the Gibbs sampling procedure, we need to iteratively sample from the full conditional distribution of each of the random variables involved in Equation 6.17 to Equation 6.20. In what follows, we show how to derive the full conditional distributions for each of these variables.

Given a fixed model structure $\mathbf{C}_q$ and a known value for the hidden variable R, the derivation of the full conditional distributions of the parameters for the Gauss-Wishart model is straightforward because of the use of conjugate priors (see explanations in Section 5.5). These conditional distributions are in the same form as the prior. In either the bicluster model or the background model, the conditional distribution for μ_j remains a normal distribution, which is illustrated in the following equation:

$$\begin{aligned}
& p(\mu_j \,|\, \mathcal{D}[\mathbf{u}, j], \sigma_j^2, \phi_j, \tau_j^2)\\
&\propto\ p(\mathcal{D}[\mathbf{u}, j] \,|\, \mu_j, \sigma_j^2) \cdot P\Big(\mu_j \,|\, \phi_j, \tau_j^2\Big)\\
&\propto\ \exp\left\{-\frac{|\mathbf{u}|}{2\sigma_j^2}\Big[\mu_j - \bar{\mu}_j\Big]^2\right\} \cdot \exp\left\{-\frac{1}{2\tau_j^2}\Big[\mu_j - \phi_j\Big]^2\right\}\\
&=\ \exp\left\{-\frac{1}{2}\left[\frac{\mu_j^2}{\frac{\sigma^2}{|\mathbf{u}|}} + \frac{\bar{\mu}_j^2}{\frac{\sigma^2}{|\mathbf{u}|}} - 2\frac{\mu_j \cdot \bar{\mu}_j}{\frac{\sigma^2}{|\mathbf{u}|}} + \frac{\mu_j^2}{\tau^2} + \frac{\phi_j^2}{\tau^2} - 2\frac{\mu_j \cdot \phi_j}{\tau^2}\right]\right\}\\
&\propto\ \exp\left\{-\frac{1}{2}\left[\mu_j^2\left(\frac{1}{\frac{\sigma^2}{|\mathbf{u}|}} + \frac{1}{\tau^2}\right) - 2\mu_j\left(\frac{\bar{\mu}_j}{\frac{\sigma^2}{|\mathbf{u}|}} + \frac{\phi_j}{\tau^2}\right)\right]\right\}, && (6.21)
\end{aligned}$$

where $\bar{\mu}_j$ is the sample mean $\bar{\mu}_j = \frac{\sum_{i \in \mathbf{u}} \mathcal{D}[i,j] - \mu_j}{|\mathbf{u}|}$, and $|\mathbf{u}|$ denotes the length of the

vector of indices $\mathbf{u}$. Let $a = \frac{1}{\tau_j^2}$ and $b = \frac{1}{\frac{\sigma_j^2}{|\mathbf{u}|}}$,

$$\begin{aligned} p(\mu_j \,|\, \mathcal{D}[\mathbf{u}, j],\, \sigma_j^2,\, \phi_j,\, \tau_j^2) &\propto \exp\left\{-\frac{1}{2}\Big[\mu_j^2\big(a+b\big) - 2\mu_j\big(a\cdot\phi_j + b\cdot\bar{\mu}_j\big)\Big]\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left[\mu_j^2\big(a+b\big) - \frac{2\mu_j\cdot\big(a\cdot\phi_j + b\cdot\bar{\mu}_j\big)\cdot(a+b)}{(a+b)} + \right.\right. \\ &\qquad \left.\left. \frac{\big(a\cdot\phi_j + b\cdot\bar{\mu}_j\big)^2\cdot(a+b)}{(a+b)^2}\right]\right\}. \end{aligned} \tag{6.22}$$

Using $\hat{\sigma}^2 = \frac{1}{a+b}$ and $\hat{\mu}_j = \frac{a\cdot\phi_j + b\cdot\bar{\mu}_j}{a+b}$, we finally derive

$$\begin{aligned} p(\mu_j \,|\, \mathcal{D},\, \mathcal{D}_R,\, \mathbf{C}_q,\, \sigma_j^2,\, \phi_j,\, \tau_j^2) &= N(\hat{\mu}_j, \hat{\sigma}_j^2) \\ \hat{\mu}_j = \frac{\frac{\varphi_j}{\tau_j^2} + \frac{\bar{\mu}_j}{\frac{\sigma_j^2}{|\mathbf{u}|}}}{\frac{1}{\tau_j^2} + \frac{1}{\frac{\sigma_j^2}{|\mathbf{u}|}}} \quad &\text{and} \quad \hat{\sigma}_j^2 = \frac{1}{\frac{1}{\tau_j^2} + \frac{1}{\frac{\sigma_j^2}{|\mathbf{u}|}}}. \end{aligned} \tag{6.23}$$

The posterior distributions for σ_j^2 is a scaled inverse-χ^2 distribution,

$$p(\sigma_j^2 \,|\, \mathcal{D}[\mathbf{u}, j],\, \mu_j,\, \nu,\, s^2) = \frac{p(\mathcal{D}[\mathbf{u}, j] \,|\, \mu_j,\, \sigma_j^2)\cdot p(\sigma_j^2 \,|\, \nu,\, s^2)}{p(\mathcal{D}[\mathbf{u}, j] | \mu_j,\, \nu,\, s^2)}. \tag{6.24}$$

The numerator of Equation 6.24 is,

$$\begin{aligned} & p(\mathcal{D}[\mathbf{u}, j] \,|\, \mu_j,\, \sigma_j^2)\cdot p(\sigma_j^2 \,|\, \nu,\, s^2) \\ &= \frac{1}{(\sqrt{2\pi}\sigma_j)^{|\mathbf{u}|}} \exp\left\{-\frac{\sum_{i\in\mathbf{u}}(\mathcal{D}[i,j]-\mu_j)^2}{2\sigma_j^2}\right\}\cdot\frac{2^{-\nu/2}}{\Gamma(\nu/2)}s^\nu\sigma_j^{-2(\frac{\nu}{2}+1)}\exp\left(-\frac{\nu s^2}{2\sigma_j^2}\right) \\ &= Z\cdot\sigma_j^{-2(\frac{\nu+|\mathbf{u}|}{2}+1)}\exp\left\{-\frac{\sum_{i\in\mathbf{u}}(\mathcal{D}[i,j]-\mu_j)^2+\nu s^2}{2\sigma_j^2}\right\}, \end{aligned} \tag{6.25}$$

where $Z = \frac{1}{\sqrt{2\pi}}\cdot\frac{2^{-\nu/2}}{\Gamma(\nu/2)}s^\nu$. The denominator of Equation 6.24 is,

$$\begin{aligned} & p(\mathcal{D}[\mathbf{u}, j] | \mu_j,\, \nu,\, s^2) \\ &= \int p(\mathcal{D}[\mathbf{u}, j] \,|\, \mu_j,\, \sigma_j^2)\cdot p(\sigma_j^2 \,|\, \nu,\, s^2)\,\mathrm{d}\sigma_j^2 \\ &= Z\cdot\int\sigma_j^{-2(\frac{\nu+|\mathbf{u}|}{2}+1)}\exp\left\{-\frac{\sum_{i\in\mathbf{u}}(\mathcal{D}[i,j]-\mu_j)^2+\nu s^2}{2\sigma_j^2}\right\}\mathrm{d}\sigma_j^2. \end{aligned} \tag{6.26}$$

Replacing $\sum_{i\in\mathbf{u}}(\mathcal{D}[i,j]-\mu_j)^2+\nu s^2$ by t, we have

$$\begin{aligned}
p(\mathcal{D}[\mathbf{u},j]|\mu_j,\nu,s^2) &\propto \int\left(\frac{1}{\sigma_j^2}\right)^{\frac{\nu+|\mathbf{u}|}{2}+1}\exp\left(-\frac{t}{2\sigma_j^2}\right)\mathrm{d}\sigma_j^2\\
&= \int\frac{2}{t}\left(\sigma_j^2\right)^2\left(\frac{1}{\sigma_j^2}\right)^{\frac{\nu+|\mathbf{u}|}{2}+1}\exp\left(-\frac{t}{2\sigma_j^2}\right)\mathrm{d}\left(\frac{t}{2\sigma_j^2}\right)\\
&= \int\left(\frac{t}{2}\right)^{-1}\left(\frac{1}{\sigma_j^2}\right)^{\frac{\nu+|\mathbf{u}|}{2}-1}\exp\left(-\frac{t}{2\sigma_j^2}\right)\mathrm{d}\left(\frac{t}{2\sigma_j^2}\right)\\
&= \left(\frac{t}{2}\right)^{-\frac{\nu+|\mathbf{u}|}{2}}\int\left(\frac{t}{2\sigma_j^2}\right)^{\frac{\nu+|\mathbf{u}|}{2}-1}\exp\left(-\frac{t}{2\sigma_j^2}\right)\mathrm{d}\left(\frac{t}{2\sigma_j^2}\right)\\
&= \left(\frac{t}{2}\right)^{-\frac{\nu+|\mathbf{u}|}{2}}\Gamma\left(\frac{\nu+|\mathbf{u}|}{2}\right) \qquad (6.27)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&p(\sigma_j^2\,|\,\mathcal{D}[\mathbf{u},j],\,\mu_j,\,\nu,\,s^2) \qquad (6.28)\\
&= \frac{\left(\frac{\sum_{i\in\mathbf{u}}(\mathcal{D}[i,j]-\mu_j)^2+\nu s^2}{2}\right)^{\frac{\nu+|\mathbf{u}|}{2}}}{\Gamma\left(\frac{\nu+|\mathbf{u}|}{2}\right)}\cdot\sigma_j^{-2\left(\frac{\nu+|\mathbf{u}|}{2}+1\right)}\exp\left\{-\frac{\sum_{i\in\mathbf{u}}(\mathcal{D}[i,j]-\mu_j)^2+\nu s^2}{2\sigma_j^2}\right\}.
\end{aligned}$$

Using

$$\hat{\nu}=\nu+|\mathbf{u}| \qquad (6.29)$$

and

$$\hat{s}^2=\frac{1}{\nu+|\mathbf{u}|}\sum_{i\in\mathbf{u}}(\mathcal{D}[i,j]-\mu_j)^2+\frac{\nu}{\nu+|\mathbf{u}|}s^2, \qquad (6.30)$$

we have,

$$p(\sigma_j^2\,|\,\mathcal{D}[\mathbf{u},j],\,\mu_j,\,\nu,\,s^2)=\frac{\left(\frac{\hat{\nu}}{2}\right)^{\frac{\hat{\nu}}{2}}}{\Gamma\left(\frac{\hat{\nu}}{2}\right)}\hat{s}^{\hat{\nu}}\sigma_j^{-2\left(\frac{\hat{\nu}}{2}+1\right)}\exp\left(-\frac{\hat{\nu}\cdot\hat{s}^2}{2\sigma_j^2}\right). \qquad (6.31)$$

That is to say,

$$p(\sigma_j^2\,|\,\mathcal{D},\,\mathcal{D}_R,\,\mathbf{C}_q,\,\mu_j,\,\nu,\,s^2)=\text{Inverse-}\chi^2(\hat{\nu},\hat{s}^2). \qquad (6.32)$$

Thanks to the conditional independence assumptions of our model as described in Section 6.3, in Equations 6.21 to 6.32, μ_j is only conditioned on σ_j^2 in the same model (i.e., the model either for the bicluster or for the background)

in addition to $\mathcal{D}$, $\mathcal{D}_R$ and $\mathbf{C}_q$; and vice versa. Note that the vector of indices $\mathbf{u}$ carries the information of both $\mathcal{D}_R$ and $\mathbf{C}_q$: for $\boldsymbol{\mu}^{\text{bcl}}$ and $(\sigma^2)^{\text{bcl}}$,

$$\mathbf{u} = \mathbf{r} \quad , j \in \mathbf{e}; \tag{6.33}$$

for $\boldsymbol{\mu}^{\text{bgd}}$ and $(\sigma^2)^{\text{bgd}}$,

$$\mathbf{u} = \begin{cases} \bar{\mathbf{r}} & j \in \mathbf{e} \\ [1, \ldots, n]^{\mathrm{T}} & j \in \bar{\mathbf{e}} \end{cases} \tag{6.34}$$

Equation 4.49 is directly applicable for the evaluation of the full conditional distribution of $\mathcal{D}_R[i]$ $(i = 1 \ldots n)$, which now becomes,

$$\gamma_i^{\text{r}} = \prod_{j \in \mathbf{e}} \frac{P\left(\mathcal{D}[i,j] \,|\, \mu_j^{\text{bcl}}, (\sigma^2)_i^{\text{bcl}}\right)}{P\left(\mathcal{D}[i,j] \,|\, \mu_j^{\text{bgd}}, (\sigma^2)_j^{\text{bgd}}\right)} \cdot \frac{v_{\bar{i}} + \zeta_1^{\text{r}}}{n - 1 - v_{\bar{i}} + \zeta_0^{\text{r}}} \quad i = 1, \ldots, n. \tag{6.35}$$

For the conditions, when applying Equation 4.52 to our model here, we have to keep in mind that Equation 6.34 indicates that μ_j^{bgd} and $(\sigma^2)_j^{\text{bgd}}$ are evaluated differently depending on whether experiment j is included in the bicluster or not. We use μ_j^{bgd1} and $(\sigma^2)_j^{\text{bgd1}}$ to denote the parameters of the background model evaluated in the former case, and μ_j^{bgd0} and $(\sigma^2)_j^{\text{bgd0}}$ for the latter case. Now, Equation 4.52 is evaluated as

$$\begin{aligned}
\gamma_k^{\text{c}} \;=\; & \frac{p(\mathcal{D} \,|\, \mathcal{D}_R,\, C_k = 1,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\Theta}^{\text{bcl}},\, \boldsymbol{\Theta}^{\text{bgd}})}{p(\mathcal{D} \,|\, \mathcal{D}_R,\, C_j = 0,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\Theta}^{\text{bcl}},\, \boldsymbol{\Theta}^{\text{bgd}})} \\
& \cdot \frac{p(\boldsymbol{\Theta}^{\text{bcl}},\, \boldsymbol{\Theta}^{\text{bgd}} \,|\, C_k = 1,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\xi}^{\text{bcl}},\, \boldsymbol{\xi}^{\text{bgd}})}{p(\boldsymbol{\Theta}^{\text{bcl}},\, \boldsymbol{\Theta}^{\text{bgd}} \,|\, C_k = 0,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\xi}^{\text{bcl}},\, \boldsymbol{\xi}^{\text{bgd}})} \cdot \frac{w_{\bar{j}} + \zeta_1^{\text{c}}}{m - w_{\bar{j}} + \zeta_0^{\text{c}}} \\
=\; & \prod_{j \in \mathbf{e}_k} \left\{ \frac{p\left(\mathcal{D}[\mathbf{r}, j] \,|\, \mu_j^{\text{bcl}},\, (\sigma^2)_j^{\text{bcl}}\right) \cdot p\left(\mathcal{D}[\bar{\mathbf{r}}, j] \,|\, \mu_j^{\text{bgd1}},\, (\sigma^2)_j^{\text{bgd1}}\right)}{p\left(\mathcal{D}[\cdot, j] \,|\, \mu_j^{\text{bgd0}}, (\sigma^2)_j^{\text{bgd0}}\right)} \right. \\
& \cdot \frac{p\left(\mu_j^{\text{bcl}} \,|\, \phi_j^{\text{bcl}}, (\tau^2)_j^{\text{bcl}}\right) \cdot p\left(\mu_j^{\text{bgd1}} | \phi_j^{\text{bgd}}, (\tau^2)_j^{\text{bgd}}\right)}{p\left(\mu_j^{\text{bgd0}} \,|\, \phi_j^{\text{bgd}}, (\tau^2)_j^{\text{bgd}}\right)} \\
& \left. \cdot \frac{p\left((\sigma^2)_j^{\text{bcl}} \,|\, \nu^{\text{bcl}}, (s^2)^{\text{bcl}}\right) \cdot p\left((\sigma^2)_j^{\text{bgd1}} | \nu^{\text{bgd}}, (s^2)^{\text{bgd}}\right)}{p\left((\sigma^2)_j^{\text{bgd0}} \,|\, \nu^{\text{bgd}}, (s^2)^{\text{bgd}}\right)} \right\} \\
& \cdot \frac{w_{\bar{j}} + \zeta_1^{\text{c}}}{m - w_{\bar{j}} + \zeta_0^{\text{c}}}
\end{aligned} \tag{6.36}$$

The first term in both Equation 6.35 and Equation 6.36 are likelihood ratios (also see the explanations for Equation 4.49 and Equation 4.52). Note that by using

likelihood ratios, the missing values in the microarray data can be neglected from the evaluation of the conditional distributions, which is equivalent to assuming that these data points have the same possibility be to generated by the bicluster model as by the background model.

6.5 Construction of the priors

Note that conjugate priors are used for the hierarchical models (see Section 6.3) for the same reason as explained in Chapter 5—their interpretability and their analytical convenience.

As we explained in the beginning of this chapter, the imposition of priors plays an utmost important role to guide the discovery of the bicluster toward the answer to the specific question that interests the biologist. Given a set of seed genes, biologists want to recruit other genes whose expression profile is similar to the seed genes, but only under a subset of conditions. Mathematically translated, the objective is to find a set of genes that have small variance under a subset of conditions, in addition the mean profile of these genes should strictly follow that of the seed genes under the selected conditions.

To impose our requirement that the mean of the genes under each experiment in the bicluster should strictly follow that of the mean of seed genes, we define

$$\varphi_j^{\text{bcl}} = \frac{\sum_{i \in \mathbf{a}} \mathcal{D}[i, j]}{|\mathbf{a}|}, \tag{6.37}$$

which is the mean of the seed genes under all the experiment j in the data set—$\mathbf{a}$ stands for the vector of indices of the seed genes, and $|\mathbf{a}|$ is the number of genes in the seed. Moreover, we use a very small value for $\boldsymbol{\tau}^{\text{bcl}}$, for example,

$$\tau_j^{\text{bcl}} = 10^{-4},\ j \in \mathbf{e}. \tag{6.38}$$

By setting

$$(s^2)^{\text{bcl}} = \frac{1}{\nu^{\text{bcl}}} \tag{6.39}$$

for the prior on $(\sigma^2)^{\text{bcl}}$, the scaled inverse-χ^2distribution becomes an inverse-χ^2 distribution, which means that no prior knowledge on the exact value of the posterior variance is imposed, and that the posterior parameters for $(\sigma^2)^{\text{bcl}}$ are of smaller values for those experiments under which the selected genes have a smaller sample variance. Raising ν^{bcl} implies a stronger belief that the posterior variance is close to the sample variance of the selected genes, the effect of which is equivalent to increasing the number of genes in the bicluster without changing the sample variance.

Similarly, we assume that the mean profile of the genes in the bicluster should be close to that of those genes that are not used as seed genes. Therefore, for

the prior on $\boldsymbol{\mu}^{\text{bgd}}$, we set $\boldsymbol{\varphi}_j^{\text{bgd}}$ to the mean of the expression levels of all the genes under experiment j,

$$\boldsymbol{\varphi}_j^{\text{bcl}} = \frac{\sum_{i \in \bar{\mathbf{a}}} \mathcal{D}[i, j]}{n - |\mathbf{a}|}; \tag{6.40}$$

and we use

$$\tau_j^{\text{bgd}} = 10^{-4},\ k \in \mathbf{e}. \tag{6.41}$$

For the priors on $(\sigma^2)^{\text{bgd}}$, we set

$$(s^2)^{\text{bgd}} = \frac{1}{\nu^{\text{bgd}}}. \tag{6.42}$$

In addition, weak priors are used for the labels, because we have little knowledge beforehand about how many genes and conditions the bicluster would contain. We typically set

$$\zeta_0^{\text{r}} = \zeta_1^{\text{r}} = 0.5 \tag{6.43}$$

$$\zeta_0^{\text{c}} = \zeta_1^{\text{c}} = 0.5. \tag{6.44}$$

In this way, ν^{bcl} and ν^{bgd} are the only two hyperparameters opened to the user for controlling the stringency of the bicluster.

6.6 Biclusters for transcriptional regulatory modules

The assumption for regulatory module discovery in transcriptional data (i.e., microarray data) is that genes governed by the same regulatory program (i.e., having the same set of regulators) share similar expression profiles under the working conditions of the regulators. A transcriptional module refers to the set of the genes that are coregulated, the conditions under which the coregulation occurs and the expression profiles of these genes under the specific set of conditions. By this definition, it is clear that a transcriptional module corresponds to a bicluster in the microarray data. The regulatory module however includes both the transcriptional module and its governing regulatory program.

Depending on the combination of regulators, the same genes can be involved in different transcriptional modules of different size (i.e., biclusters including different additional genes and different working conditions). A hypothetical example is given in Figure 6.3.

A gene expression data set can be subdivided in several overlapping context dependent modules. Modules found in many conditions can be expected to contain few genes with a very specific function. Indeed, the more conditions

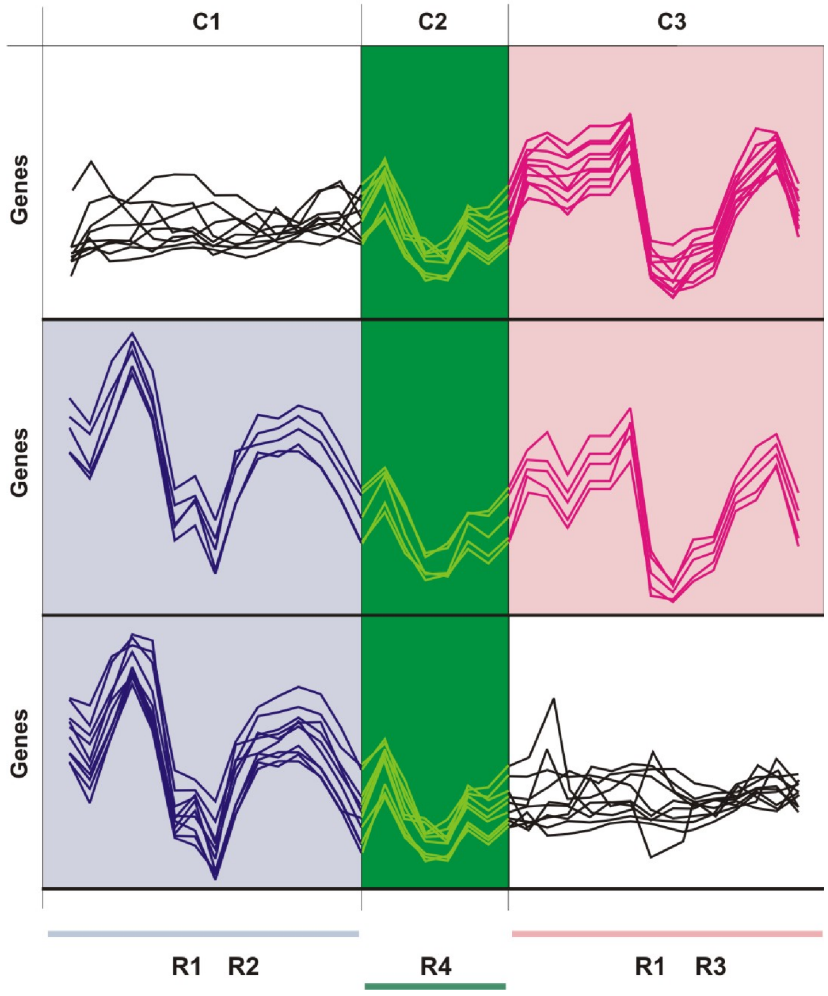


Figure 6.3: C1, C2, and C3 represent three unrelated conditions. Ri represents different regulators active in the respective condition dependent regulatory programs. R1 and R2 are active in C1; R1 and R3 are active in C3; and R4 is active in C2. (A) Distinct partially overlapping modules exist. Modules consisting of a few genes, tightly coexpressed in many conditions can be hypothesized to be associated with a highly specific function (horizontal middle panel). They consist of genes that respond to the same regulatory program and are coexpressed under all conditions. As modules are extended with more genes, the number of conditions can be expected to decrease. Genes within such extended modules only share part of the regulatory program (i.e., the one that is active under the selected conditions—top and bottom panel).

	ϕ	τ^2	ν	s^2
Background	0	0.3	200	1.0
Bicluster 1	0	2	50	0.25
Bicluster 2	0	1.5	50	0.36
Overlap (50 ×30)	–	–	50	0.16

Table 6.1: Parameter settings for the generation of the synthetic data set.

genes appear to be coexpressed in, the more similar their regulatory program tends to be and the more connected their role in the pathway becomes. In a module, the number of genes will usually increase with a decreasing number of conditions. Obviously, there will be more genes that only share part of their regulatory program, (i.e., the part that is active under the set of conditions tested). The fewer the number of conditions included in the module one considers, the less stringent the requirements on the overlap in the regulatory program becomes (see Figure 6.3).

6.7 Experiments on synthetic data

To demonstrate the performance of the algorithm, we have embedded two overlapping biclusters—(1) Bicluster 1 of 300 rows by 80 and (2) Bicluster 2 of 400 rows by 50 columns—into a noisy background of 2000 rows by 100 columns. The two biclusters overlap each other by 50 rows and 30 columns. The data of the two biclusters at the overlapping area share the same mean for each column. However, the variance under each column in the overlapping areas is generated by a different inverse-χ^2 distribution other than those that generated the variances for the columns under Bicluster 1 and Bicluster 2. The parameters used to generate the data are listed in Table 6.1. Part of the resulting data set is illustrated in Figure 6.4, which includes 100 rows from Bicluster 1, 150 rows from Bicluster 2 (note that all the 50 rows at the overlapping region are all included) and 100 rows from the background data are selected for illustrate. The rows and columns in Figure 6.4 are rearranged so that the embedded biclusters are clearly visible.

In this way, the synthetic data resembles the one that we discussed for Figure 6.3, and yet contains different structures that allow us to explore the influence of the user input parameters ν^{bcl} and ν^{bgd} on the resulting biclusters. We want to investigate which part of the three biclusters is recovered under different parameter settings.

To illustrate the performance of the algorithm under different parameter settings ν^{bcl} and ν^{bgd}, we use four sets of seed rows that are extracted from different parts of the data set. Seed 1 consists of five rows in the non-overlapping area of Bicluster 1, Seed 2 consists of five rows in the non-overlapping area of Bi-

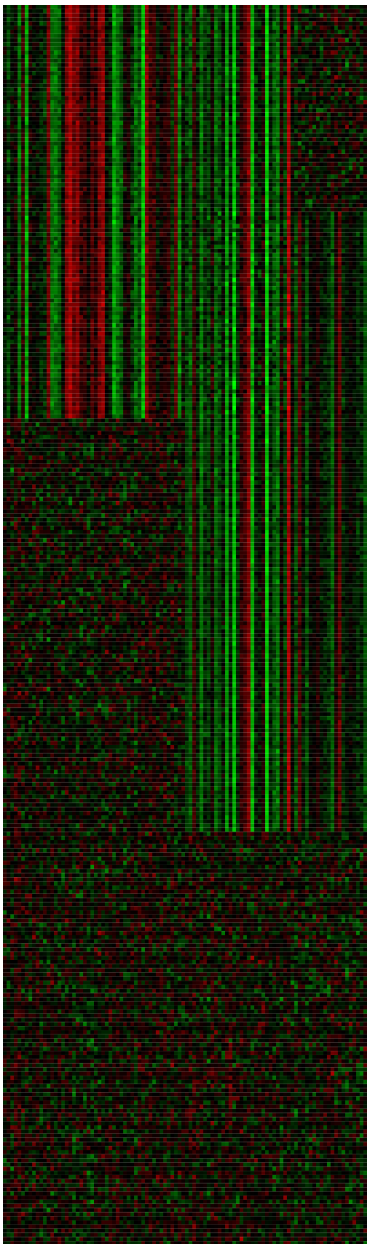


Figure 6.4: Part of the synthetic continuous data set. The values of the data points are reflected by the color scale. The rows and the columns are rearranged to manifest the embedded biclusters.

cluster 2, Seed 3 is made up of five rows out of the 50 rows where Bicluster 1 overlaps with Bicluster 2, and finally Seed 4 is composed of two rows from the overlapping area of Bicluster 1 and Bicluster 2, two of the non-overlapping rows of Bicluster 2, and two random rows from the background data.

We found that our algorithm retrieved reproducible patterns for each set of seeds under each of the parameter settings that we tested for this synthetic data set. We performed the algorithm 10 times on each of Seed 1, 2, 3, and 4, under each parameter setting. Each time we aim to find one bicluster (which means that Loop 3 in Figure 4.7 is not performed). However, for these 10 rounds on a particular set of seed—take Seed 1 for example—a different random set of seed rows is generated according to the requirement described above for Seed 1. In Table 6.2 and Table 6.3, we report the size of the bicluster discovered under each parameter setting, as well as the number of times that the corresponding bicluster was found out of the 10 rounds using the same parameter setting.

These two tables tells us that increasing ν^{bgd} raises the stringency in selecting the columns for the bicluster. With a larger ν^{bgd}, fewer columns are selected. In contrast, an increase in ν^{bcl} helps to increase the diversity of the resulting biclusters. The table suggests that every mode in the posterior distribution is amplified under a larger μ^{bcl}. This is why biclusters corresponding to local maxima solutions (i.e., the "small" biclusters, see the footnotes of Table 6.2 for their definition) are more frequently found with a larger ν^{bcl}. However, comparing biclusters containing the same number of rows but discovered by different ν^{bcl} for the same seed, we suppose that a larger ν^{bcl} reinforces the power of the seeds, and consequently relatively fewer columns are selected for the bicluster. In addition, the result on Seed 2 suggests that a combination of large ν^{bgd} and small ν^{bcl} encourages the resulting bicluster to include more rows and fewer columns.

The two tables also illustrate the influence of the choice of seeds. Bicluster 2 has a weaker pattern compared with Bicluster 1, because the means of its rows under the non-overlapping columns (with Bicluster 1) have a smaller amplitude than those of Bicluster 1 (see Figure 6.4). That is to say that the patterns of both Bicluster 1 and the overlapping bicluster (containing all the rows in the Bicluster 1 and Bicluster 2, and only those overlapping columns of Bicluster 1 and Bicluster 2), which corresponds to the vertical middle panel in Figure 6.3, are more dominant than Bicluster 2. Therefore, seed rows picked at any region of Bicluster 1 (i.e., either from the non-overlapping rows or from the overlapping ones), most frequently, directs the result to the embedded pattern of Bicluster 1. However, the discovery of Bicluster 2 is highly influenced by the selection of the seeds. When the seeds are selected from the overlapping rows, the result is directed toward the discovery of Bicluster 1. When the seeds are made up from those "pure" rows for Bicluster 2, Bicluster 2 can be identified very frequently. Finally, when the seeds are mixed (i.e., extracted from both the non-overlapping rows, the overlapping ones, and even added with some

Seed 1		ν^{bgd}							
		1		10		50		100	
		size	#[a]	size	#	size	#	size	#
ν^{bcl}	1	300×80	10	300×78	10	300×48	10	300×25	10
	10	300×79	10	300×77	10	300×49	10	300×25	10
	50	300×75 small[b] –[c]	2 2 6	300×71	10	300×43	10	300×23	10
	100	small –	1 9	650×27 300×51	1 9	300×36	10	300×19	10

Seed 2		ν^{bgd}							
		1		10		50		100	
		size	#	size	#	size	#	size	#
ν^{bcl}	1	400×59 300×80	7 3	650×69 400×49	2 8	400×45	10	650×15	10
	10	400×50 350×78	7 3	400×50	10	400×45	10	650×15	10
	50	400×50 small –	6 1 3	400×50	10	400×35	10	650×12 400×14	1 9
	100	650×30 – –	1 9	400×46	10	400×23	10	650×11 400×13	3 7

[a]The number of times that the corresponding bicluster was found out of the 10 runs using the same parameter settings.

[b]The Gibbs sampling converged to a very small bicluster by the end of the 500 iterations. The bicluster is typically of one row or one column in size, or of size such as 4 by 4.

[c]The algorithm failed to converge after 50 re-initializations (see Loop 2 in Figure 4.7; i.e., no bicluster was found.

Table 6.2: The size of the bicluster discovered using each set of seeds under different parameter settings.

Seed 3		ν^{bgd}							
		1		10		50		100	
		size	#	size	#	size	#	size	#
ν^{bcl}	1	300×80	10	300×76	10	300×50	10	300×26	10
	10	300×80	10	300×77	10	300×49	10	300×26	10
	50	400×45 small –	1 1 8	400×40 300×70	1 9	300×44	10	300×24	10
	100	– –	 10	300×60 small	2 8	300×37	10	300×16	10

Seed 4		ν^{bgd}							
		1		10		50		100	
		size	#	size	#	size	#	size	#
ν^{bcl}	1	650×85	10	650×81 400×48	9 1	650×41 300× 4	10 1	650× 8	9
	10	650×81 400×50	8 2	650×73 400×50	7 3	650×38	10	650× 7 300× 4	8 2
	50	 400×44 small –	 7 1 2	650×50 400×42 small	4 2 4	650×28	10	650× 5	10
	100	650×50 small –	1 6 3	650×33 400×30 300×32 small	3 1 1 5	650×15 300× 8	8 2	650× 5 300× 5 –	4 2 4

Table 6.3: The size of the bicluster discovered using each set of seeds under different parameter settings (continued).

random rows from the background), the discovery is mostly directed toward the finding of the overlapping bicluster.

These experiments on synthetic data set illustrate the flexibility of the algorithm to discovery different regions of overlapped transcription modules, and give some intuitive directions on how to manipulate the findings from the input of the algorithm—the seeds and the parameters ν^{bcl} and ν^{bgd}.

6.8 Transcriptional module discovery in *Saccharomyces cerevisiae*

To discover a transcriptional regulatory module, we use the method of De Bie *et al.* (2005) [15] to reveal regulatory programs as well as sets of seed genes that are governed by these regulatory programs. De Bie *et al.* (2005) [15] combine three independent data sources, namely genome-wide location data (ChIP-chip data), motif information as obtained by phylogenetic shadowing, and gene expression profiles. Seed genes are identified from the input information as those that share the same combination of regulators and motifs, and whose expression profiles have a large correlation. However, this method is only suitable when the microarray data is obtained under homogeneous conditions. Our biclustering method further extends the work by applying the seed genes (typically of the size of 3 to 15 genes) to microarray data collected from a heterogeneous compendium for the discovery of transcriptional modules under the regulatory program of interest.

We use three sets of seed genes found by the method of De Bie *et al.* (2005) when it is applied to the data from Kellis *et al.* (2003) [58] (motif data), Lee *et al.* (2002) [66] (ChIP-chip data [80]), and Spellman *et al.* (1998) [93] (micorarray data). The final data contains 6157 genes, 267 experiments and 70 conditions. Two sets of seed genes (referred to as Seed 1 and Seed 2 hereafter) are composed of cell cycle related genes, see Table 6.4, Table 6.5, and Table 6.6. To be more specific, Seed 1 is related to the early G1 phase of the cell cycle and Seed 2 is related to the G2 phase of the cell cycle. The other set of seed genes (i.e., Seed 3) is involved in ribosome biogenesis, a more general function, see Table 6.7.

We applied our algorithm to the combined data set on *Saccharomyces cerevisiae* from Gasch *et al.* (2000) [37] (with stress-response experiments), Spellman *et al.* (1998) [93] and Cho *et al.* (1998) [22] (both with cell cycle-related experiments) for the three sets of seeds. Each of the original data set was centered and rescaled so that measurements under every microarray experiment have mean of 0 and standard deviation of 1. Then, the gene profiles in each data set were centered and rescaled in the same way. The resulting data sets were then put alongside each other. We expect that for Seed 1 and Seed 2 the algorithm would identify all the experimental conditions under the data from Spellman

Regulators	Gene ID	Names	MIPS	Description
Mbp1 Swi4 Swi6 Stb1	YCR065W	HCM1	11.02.03.04	transcriptional control
	YDL003W	MCD1	10.03.01	mitotic cell cycle and cell cycle control
			10.03.04.03	chromosome condensation
	YDR097C	MSH6	10.01.05	DNA recombination and DNA repair
	YER095W	RAD51	10.01.05	DNA recombination and DNA repair
			10.01.05.01	DNA repair
			10.03.02	meiosis
			34.11.03.07	pheromone response, mating-type determination, sex-specific proteins
	YGL038C	OCH1	01.05.01	C-compound and carbohydrate utilization
			14.07	protein modification
	YGR109C	CLB6	10.01.03	DNA synthesis and replication
			10.03.01	mitotic cell cycle and cell cycle control
			10.03.02	meiosis

Table 6.4: Information about Seed 1. The first column gives names of the regulators. The second column lists the systematic names of the seed genes, while the third column lists the gene names. The fourth column lists the MIPS functional categories of the corresponding gene, and the last column describes the functional categories.

Regulators	Gene ID	Names	MIPS	Description
	YGR221C	TOS2	40.01	cell growth / morphogenesis
			42.04	cytoskeleton
			43.01.03.05	budding, cell polarity and filament formation
	YKL113C	RAD27	10.01.03	DNA synthesis and replication
			10.01.05.01	DNA repair
Mbp1	YLR103C	CDC45	10.01.03	DNA synthesis and replication
Swi4			10.03.01	mitotic cell cycle and cell cycle control
Swi6	YML027W	YOX1	11	TRANSCRIPTION
Stb1	YMR179W	SPT21	11.02.03.04	transcriptional control
	YMR199W	CLN1	10.03.01	mitotic cell cycle and cell cycle control
	YPL267W	YPL267W	99	UNCLASSIFIED PROTEINS
	YPR120C	CLB5	10.01.03	DNA synthesis and replication
			10.03.01	mitotic cell cycle and cell cycle control
			10.03.02	meiosis

Table 6.5: Information about Seed 1 (continued).

Table 6.6: Information about Seed 2.

Regulators	Gene ID	Names	MIPS	Description
Ndd1	YJL051W	–	99	UNCLASSIFIED PROTEINS
Fkh2	YGL021W	ALK1	32.01	stress response
Mcm1	YLR190W	–	99	UNCLASSIFIED PROTEINS

Table 6.7: Information about Seed 3.

Regulators	Gene ID	Names	MIPS	Description
	YGR148C	RPL24B	12.01	ribosome biogenesis
	YGL189C	PRS26A	12.01	ribosome biogenesis
	YER056C-A	RPL34A	12.01	ribosome biogenesis
	YER131W	RPS26B	12.01	ribosome biogenesis
	YGL031C	RPL24A	12.01	ribosome biogenesis
	YGL103W	PRL28	12.01	ribosome biogenesis
	YER102W	RPS8B	12.01	ribosome biogenesis
	YLR167W	RPS31	12.01	ribosome biogenesis
Fkl1 Yap5 Rap1			14.13.01	cytoplasmic and nuclear protein degradation
	YLR029C	RPL15A	12.01	ribosome biogenesis
	YLR333C	RPS25B	12.01	ribosome biogenesis
	YOL127W	RPL25	12.01	ribosome biogenesis
	YOL040C	RPS15	12.01	ribosome biogenesis
	YLR344W	RPL26A	12.01	ribosome biogenesis
	YLR441C	RPS1A	12.01	ribosome biogenesis
			34.11.03.07	pheromone response, mating-type determination, sex-specific proteins

et al. (1998) [93] and Cho *et al.* (1998) [22], and recruit additional genes related to cell cycle regulation. For Seed 3, we expect the algorithm to find a bicluster consisting of most of the experimental conditions in the data set.

For each set of seeds, we ran the biclustering algorithm for 1000 iterations to have a sufficient number of samples for evaluating the posterior distributions for the random variables. The number of burn-in iterations was determined as described in Chapter 4. A gene or a condition was selected (to be in the bicluster) if in 95% of the collected samples (i.e., iterations), the gene or the condition had a probability of more than 0.9 to be in the bicluster. Figure 6.5, Figure 6.6, and Figure 6.7 illustrate the three biclusters we found for Seed 1, Seed 2, and Seed 3. Finally, we validated the bicluster by calculating the functional enrichment of the bicluster using a hypergeometric distribution [97],

where the functional categories of the genes are obtained from MIPS [72]. In Table 6.8 to 6.10 we only report the functional categories whose p-values are lower than 0.01 (as well as those p-values). The input parameters ν^{bcl} and ν^{bgd} are reported as well.

In accordance with our expectation, the genes recruited for Seed 1 are mainly enriched in cell cycle related functional categories. In addition, only the cell cycle experimental conditions are recruited by the bicluster.

Seed 2 is experimentally detected (i.e., based on the ChIP-chip data) to be regulated by Ndd1, Fkh2, and Mcm1, which are cell cycle regulators. Yet, according to the MIPS database, two of the three genes in Seed 2 are annotated as functionally unknown, and the other gene is only associated to "stress response". The results show that the biclustering algorithm mainly recruited genes that are functionally enriched in categories of "cell cycle and DNA processing" and "cell type differentiation". Thus, the biclusters discovered by our algorithm confirm that the three seed genes might have cell cycle related functions.

Seed 3 is composed of 14 genes that are in the functional category of "ribosome biogenesis". The algorithm recruited genes that are highly enriched in the same functional category, 119 out of the 132 selected genes are found to have the function "ribosome biogenesis". For those 13 genes that are selected for the bicluster but are not associated with "protein synthesis" according to MIPS, we consulted the *Saccharomyces* Genome Database [6] (SGD) and found that 10 of these genes are rather dubious ORFs that overlap with various known protein synthesis genes on the other strand of the DNA (see Table 6.11).

Although Seed 3 is obtained by applying the method of De Bie *et al.* (2005) [15] to the data set from Spellman *et al.* (1998) [93], the cell cycle related experimental conditions are seldom selected to be in the bicluster, while almost all the stress response related conditions from Gasch *et al.* (2000) [37] are selected. This result shows that data set from Gasch *et al.* (2000) [37] might be a better data set to look at for the study of "ribosome biogenesis" than those from Spellman *et al.* (1998) [93] and Cho *et al.* (1998) [22], justified by either some biological explanation or the experimental noise presented in the data, which needs to be further investigated.

6.9 Conclusion

In this chapter, we give a full explanation of the Bayesian hierarchical model for the problem of biclustering genes. We show here how our priors can be constructed to incorporate information from other data sources to direct the discovery of biclusters to answer specific questions in systems biology. We illustrate the effectiveness of our algorithm in assisting the discovery of regulatory transcriptional on a combined data set on yeast.

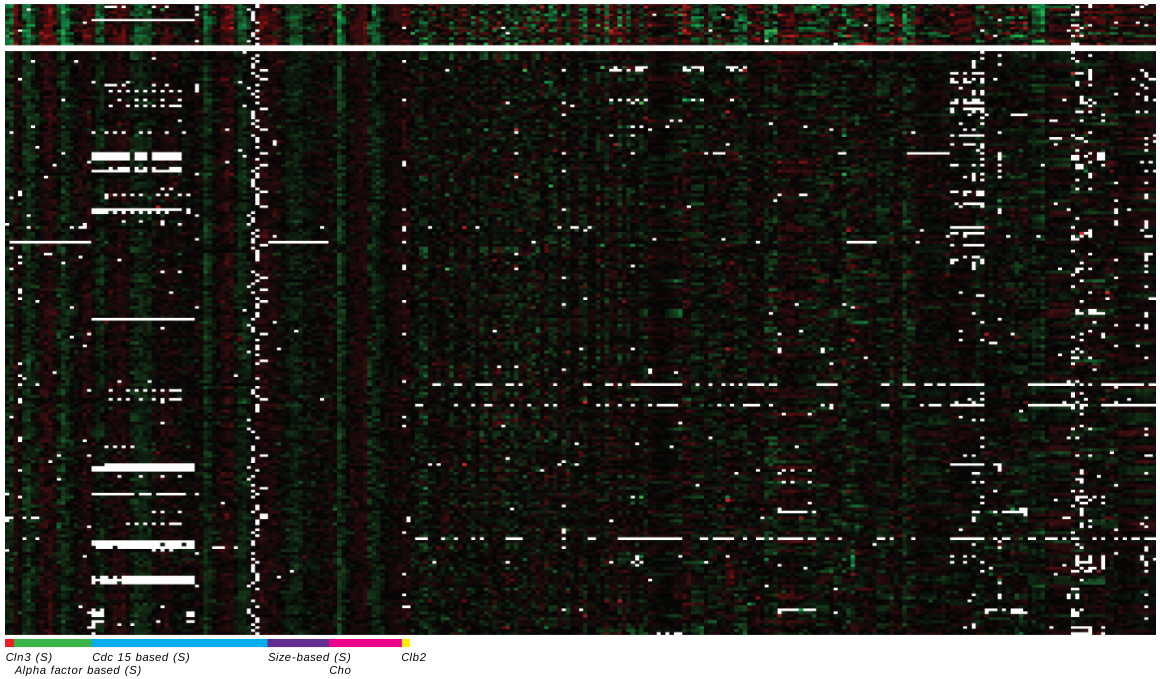


Figure 6.5: Result obtained for Seed 1: The top heatmap represents the expression profiles of the seed genes. The bottom heatmap illustrates the expression profiles of the selected genes under all the experimental conditions. Gene expression profiles are represented by rows, and experimental conditions by columns. The conditions are rearranged according to their posterior probability to belong to the bicluster in descending order. The conditions labeled by color tags are selected for the bicluster. The white spots in the heatmap represent missing values in the data.

Figure 6.6: Result obtained for Seed 2: The top heatmap represents the expression profiles of the seed genes. The bottom heatmap illustrates the expression profiles of the selected genes under all the experimental conditions. Gene expression profiles are represented by rows, and experimental conditions by columns. The conditions are rearranged according to their posterior probability to belong to the bicluster in descending order. The conditions labeled by color tags are selected for the bicluster. The white spots in the heatmap represent missing values in the data.

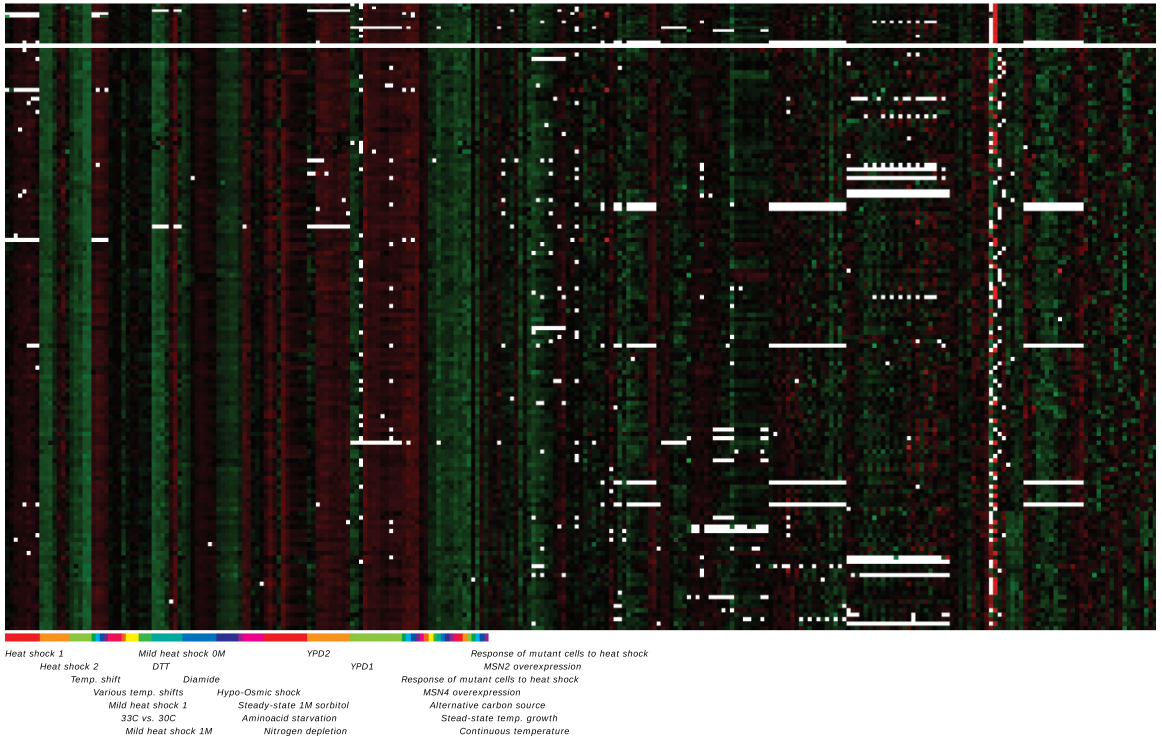


Figure 6.7: Result obtained for Seed 3: The top heatmap represents the expression profiles of the seed genes. The bottom heatmap illustrates the expression profiles of the selected genes under all the experimental conditions. Gene expression profiles are represented by rows, and experimental conditions by columns. The conditions are rearranged according to their posterior probability to belong to the bicluster in descending order. The conditions labeled by color tags are selected for the bicluster. The white spots in the heatmap represent missing values in the data.

Seed 1: 197 genes covering 88 functional categories, and 6 experimental conditions							
ν^{bcl}	ν^{bgd}	MIPS ID	Description	#	p-value	Excl. seed genes	Selected conds.
40	1	**10.01.03**	DNA synthesis and replication	27	1.2e-12	*none*	all the cell cycle conditions of the Spellman data set and the Cho data set
		10.01	DNA processing	44	1.2e-12		
		10	Cell cycle and DNA processing	75	3.9e-12		
		10.01.05	DNA recombination and DNA repair	24	5.56e-9		
		10.03.01	mitotic cell cycle and cell cycle control	33	1.9e-8		
		10.03	cell cycle	43	2e-8		
		10.01.05.01	DNA repair	13	4.8e-6		
		10.03.04	nuclear and chromosomal cycle	9	1.7e-5		
		01.03.07	deoxyribonucleotide metabolism	4	2.8e-4		
		43.01.03.05	budding, cell polarity and filament formation	17	1.3e-3		
		10.03.05.01	spindle pole body/centrosome and microtubule cycle	2	3.0e-3		
		10.03.05	cell cycle dependent cytoskeleton reorganization	2	3.0e-3		
		43.01	fungal/microorganismic cell type differentiation	21	4.1e-3		
		43.01.03	fungal and other eukaryotic cell type differentiation	21	4.1e-3		
		43	CELL TYPE DIFFERENTIATION	21	4.1e-03		
		10.03.04.01	centromere/kinetochore complex maturation	3	4.4e-3		
		42.04	cytoskeleton	10	8.6e-03		
		10.03.02	meiosis	9	9.0e-03		

Table 6.8: Functional enrichment of the bicluster found for Seed 1.

Seed 2: 37 genes covering 47 functional categories, and 35 experimental conditions							
ν^{bcl}	ν^{bgd}	MIPS ID	Description	#	p-value	Excl. seed genes	Selected conds.
5	1	**10.03**	cell cycle	13	1.0e-5	*none*	all the cell cycle conditions and half of the stress conditions
		10.03.03	cytokinesis (cell division) septum formation	4	4.5e-5		
		43.01.03.05	budding, cell polarity and filament formation	8	6.0e-5		
		43.01.03	fungal and other eukaryotic cell type differentiation	9	1.8e-4		
		43.01	fungal/microorganismic cell type differentiation	9	1.8e-4		
		43	CELL TYPE DIFFERENTIATION	9	1.8e-4		
		10.03.01	mitotic cell cycle and cell cycle control	8	1.0e-3		
		10	CELL CYCLE AND DNA PROCESSING	13	1.4e-3		

Table 6.9: Functional enrichment of the bicluster found for Seed 2.

Seed 3: 132 genes covering 28 functional categories, and 39 experimental conditions						
ν^{bcl}	ν^{bgd}	MIPS ID	Description	#	*p*-value	Excl. seed genes	Selected conds.
130	1	**12**	PROTEIN SYNTHESIS	119	2.4e-13	YLR333C	most of the stress conditions
		12.01	ribosome biogenesis	112	1.7e-11		
		12.04	translation	7	4.9e-04		

Table 6.10: Functional enrichment of the bicluster found for Seed 3.

Gene ID	Function
YDR417C	Dubious ORF overlapping with RPL12B
YGL102C	Dubious ORF overlapping with RPL28
YJL188C	Dubious ORF overlapping with RPL39
YLL044W	Dubious ORF overlapping with RPL8B
YLR062C	Dubious ORF overlapping with RPL22A
YLR076C	Dubious ORF overlapping with RPL10
YLR339C	Dubious ORF overlapping with RPL0
YPL142C	Dubious ORF overlapping with RPL33A
YPR044C	Dubious ORF overlapping with RPL43A
YOR277C	Dubious ORF overlapping with CAF20 (whose MIPS functional category ID is 12.04)
YLR150W	Protein that binds G4 quadruplex and purine motif triplex nucleic acid; acts with Cdc13p to maintain telomere structure; interacts with ribosomes and subtelomeric Y' DNA; multicopy suppressor of tom1 and pop2 mutations
YML022W	Adenine phosphoribosyltransferase, catalyzes the formation of AMP from adenine and 5-phosphoribosylpyrophosphate; involved in the salvage pathway of purine nucleotide biosynthesis
YPL273W	S-adenosylmethionine-homocysteine methyltransferase, functions along with Mht1p in the conversion of S-adenosylmethionine (AdoMet) to methionine to control the methionine/AdoMet ratio

Table 6.11: Function (according to SGD) of the genes that are found for the bicluster applying Seed 3, and are not associated to "protein synthesis" according to MIPS.

Chapter 7

Discussion and conclusion

In this thesis, we have developed a general framework based on Bayesian hierarchical models and Gibbs sampling for the biclustering microarray data. We have also refined our methodology into two dedicated models to provide solutions to the problems of both the biclustering experiments and the biclustering of genes, which are two distinct problems because of the asymmetry of microarray data.

In this chapter, we summarize the achievements of this work. In addition we also recognize here the limitations of our algorithms, based on which we propose new challenges for future research on this topic.

7.1 Achievements of the work

Our choice of strategy for biclustering provides our algorithms with several key characteristics that fit the needs of microarray data analysis. We enumerate here the advantages of our algorithms, especially in relation to the use of Bayesian models and Gibbs sampling.

Handling missing values in the data

Because of the use of probabilistic models, the association of each row or each column to the bicluster essentially depends on the likelihood ratio of the related data points (see explanations in Chapter 4). Consequently, missing values in the microarray values are handled in the most natural way by assuming that they are generated equally likely by the background model and by the bicluster model. This handling of missing values is independent of the specific model of use. However, an alternative when using the equal frequency discretization before performing the algorithm for biclustering experiments (as described in

Chapter 5) is to randomly assign the missing values to one of the three bins. We encourage the generic treatment of missing values. But in practice, we find that the biclustering results of the two methods differs little.

Introduction of prior knowledge and integration of information from other data sources

Bayesian models provide a systematic basis for the introduction of prior knowledge and the integration of other data sources. We have demonstrated in Chapter 6 the usefulness of our method in cooperation with other methods to discover gene regulatory modules systems biology. First, useful information from other data sources (in this case, the ChIP-chip data and motif information) are extracted (by the method of De Bie *et al*. (2005) [15]) in the form of seed genes, which are corregulated genes that show similar gene expression profiles in a relatively restricted microarray data set. Then we use these seed genes to build a Gauss-Wishart prior model that is compatible with our Bayesian hierarchical model for the biclustering of genes. The result of our biclustering algorithm reveals highly coexpressed genes under a subset of biological conditions that are highly correlated to the working conditions of the governing regulatory program. We also illustrated (in Chapter 5) the same methodology for cooperating information from a small number of patient examples to direct the discovery of biclusters toward the finding of gene expression fingerprint of subtle traits.

Robust results

Because of the complicated and noisy nature of the microarray data, finding the global mode of the target posterior distribution, which gives answer to the biological question under concern, is a critical issue for choosing the optimization methods for the probabilistic model. (Structural) EM is one of the popular methods for solving the missing data problem the biclustering problem. However, EM is well known for its frequent troubles with local maxima. Taking into account that it is never obvious in advance how many runs of an EM procedure can guarantee the discovery of the global mode, we opt for the Gibbs sampling procedure for the estimation of the Bayesian model. Though it takes longer for the algorithm to converge, Gibbs sampling gives a much higher chance of finding the global maximum solution. We have demonstrated in Chapter 5 and Chapter 6 that our algorithms find relatively frequently the bicluster that corresponds to the global maximum solution for the data. In addition, the final biclusters discovered by our algorithm often only differ in a few genes or a few conditions in size.

Allowing genes to belong to different biclusters

By discovering one bicluster at a time, we avoid the problem of associating a gene only to one bicluster. In the case of global bicluster discovery for the experiments (see Chapter 5), we mask a found bicluster by assigning the experiments of the bicluster permanently to the background for the discovery of any further biclusters, while allowing the genes of the found bicluster to be candidates for the other biclusters. When a query of seeds is used to direct the discovery of bicluster, we assume that there is only one optimal bicluster in the data set for the query. In this case, biclusters discovered by inputs of different seeds can still overlap in either the gene or the experiment dimension.

7.2 Limitations of the work

As any other algorithms, desirable features always mean compromises for the others. We list a few main issues where improvements can be made.

Improvement of the time complexity of the algorithm

First, the time complexity of the algorithm can curb the popularity of the algorithm. The judgment of convergence of Gibbs sampling has long been an issue, to guarantee the convergence and to collect sufficient amount of samples for statistical evaluation, it is common practice to collect a large amount of samples during the sampling procedure. A(n) (alternative) way to accelerate the optimization is favorable for the analysis of the ever-growing-in-size microarray data.

Global discovery for the biclustering of genes

Secondly, although the global bicluster discovery works well for the problem of biclustering experiments, where the number of modes in the posterior distribution is relatively small, our method for biclustering genes is mainly designed for the use for directed discovery of transcriptional modules. Applying non-informative priors to the biclustering of genes often result in discovery of bisecting the data set into two groups whose mean expression profiles are anti-correlated with each other, which corresponds to the two main modes in the posterior distribution. It is desired to improve the algorithm by allowing input of minimum information about a desired bicluster, and therefore allowing the spontaneous discovery of various embedded biclusters in the data.

Overlapping patterns

Another issue concerns the discovery of overlapping patterns. For the global discoveries, overlaps of biclusters are allowed in one dimension (for which we often choose the gene dimension) while is prohibited for the other dimension

(i.e. the experiment dimension in our case). However, biologically speaking, the experiments in different biclusters can also overlap with each other. An example would be that one bicluster groups the experiments according to pathological types of the tumors, and the other includes experiment of the same drug response. Therefore, an improvement in strategy that allows the biclusters to overlap in both dimensions is favorable.

7.3 Future directions

In addition, the work can be extended in following aspects.

Other dedicated models

First, our general framework (i.e., applying Gibbs sampling on Bayesian models for microarray data as described in Chapter 4) can be applied to a variety of dedicated models. What we provided in this thesis are just two of the examples.

***t* models:** Take the biclustering of experiments for instance, other models such as those whose likelihood function is in the form of *t*-distribution can be explored [71], considering their success in detecting differentially expressed genes [9].

HMM models: Further, incorporating dynamic models that are able to discover time dependences in the relations between the genes will bring the usefulness of the algorithm to another level. This ability of the model is important for the study of cell cycle for example. In addition, genes only respond to the regulator when the protein level of the regulator reaches a certain threshold in the cell. The ability of the models to catch time dependency can certainly benefit the research of regulatory modules. Our framework is immediately extendable to accommodate this modification by using a hidden Markov Model (HMM) to replace the multivariate model that describes the gene expression values under each condition (i.e., node $\mathbf{Y}_k$ in Chapter 6). Of course this replacement would introduce more parameters into the model. Nevertheless, additional assumptions to limit the complexity of the model as well as some necessary modification in the methodology can help to control the computational complexity of the algorithm.

Incorporation of priors

Secondly, the Bayesian hierarchical model can allow prior knowledge to be introduced in other ways. For example, if we allow the parameters ζ^{r} (and ζ^{c}) to have different values for different rows (and different columns), information of the seeds can be introduced directly at this level—ζ^{r} of the seed rows would

have a larger value, while the rest has a smaller value. The method that we illustrated in this thesis (i.e., building a Gauss-Wishart model from the seed genes) is a more general approach. If such prior model is available (e.g., from some other inference methods such, as neural networks), it can directly be plugged into the framework of our method. Finally, of course, the seed genes can be obtained from other sources such as text information, GO categories, discoveries in wet labs, and so on.

Appendix: Probabilistic distributions

Hereunder, we provide a list of distributions used in this thesis.

Bernoulli distribution

Notation

$$X \sim \text{Bernoulli}(\lambda)$$

Parameter

λ: Probability of a successful Bernoulli trial.

Distribution

$$P(X = x) = \begin{cases} 1 - \lambda, & x = 0 \\ \lambda, & x = 1 \end{cases}$$

Description

The Bernoulli distribution describes the probabilities of its two possible outcomes—0 and 1. The probability of 1 ("success") is λ and the probability of 0 ("failure") is $1 - \lambda$.

Beta distribution

Notation

$$X \sim \text{Beta}(\alpha, \beta)$$

Parameters

λ: Prior number of counts of successful Bernoulli trials.
β: Prior number of counts of failed Bernoulli trials.

Density function

$$p(X = x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \cdot \Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta}, \quad x \in [0, 1]$$

Description

The Beta distribution is the conjugate prior for the Bernoulli distribution. The outcome of the Beta distribution represents the probability of a Bernoulli trial to be successful.

Multinomial distribution

Notation

$$\boldsymbol{X} \sim \text{Multin}(n; p_1, \ldots, p_k)$$

Parameters

n: Sample size.

$p_1, \ldots, p_k$: Probabilities for each of the outcome k, $\sum_{i=1}^{k} p_i = 1$, $0 \leq p_i \leq 1$.

Distribution

$$P(\boldsymbol{X} = [x_1, \ldots, x_k]) = \begin{pmatrix} n \\ x_1, \cdots, x_k \end{pmatrix} p_1^{x_1} \ldots p_k^{x_k}, \quad \sum_{i=1}^{k} x_i = n,$$

where $\begin{pmatrix} n \\ x_1, \cdots, x_k \end{pmatrix}$ denotes the binomial coefficient.

Description

The multinomial distribution describe the probability to observe each event X_i (for $i = 1, \ldots, k$) x_i times.

Dirichlet distribution

Notation

$$\boldsymbol{X} \sim \text{Dirichlet}(\boldsymbol{\alpha})$$

Parameters

$\boldsymbol{\alpha} = [\alpha_i, \ldots, \alpha_k]$: Prior number of counts of for observing each event i, for $i = 1, \ldots, k$.

Density function

$$p(\boldsymbol{X} = [x_1, \ldots, x_k]) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_k)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_k)} x_1^{\alpha_1 - 1} \cdots x_k^{\alpha_k}, \quad \sum_{i=1}^{k} x_i = n,\ x_i \in [0, 1]$$

Description

The Dirichlet distribution is the conjugate prior for the multinomial distribution. The outcome of the Dirichlet distribution represents the probability of observing each event i, for $i = 1, \ldots, k$.

k-variate Gaussian distribution

Notation

$\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma)$

Parameters

$\boldsymbol{\mu}$: A vector of length k, providing the mean of the distribution. Σ: A symmetric matrix of dimension $k \times k$, specifying the variance of the distribution.

Density function

$$p(\boldsymbol{X} = \boldsymbol{x}) = (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \cdot \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}} \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right\}$$

Description

The Gaussian distribution is also called the normal distribution. It is the most widely used continuous distribution.

Scaled inverse-χ^2 distribution

Notation

$$X \sim \text{Inverse-}\chi^2(\nu, s^2)$$

Parameters

ν: Degree of freedom. s^2: Scale.

Density function

$$p(X = x) = \frac{(\frac{\nu}{2})^{\frac{\nu}{2}}}{\Gamma(\frac{\nu}{2})} s^{\nu} x^{-(\frac{\nu}{2}+1)} \exp - \frac{\nu s^2}{2x}$$

Description

The scaled inverse-χ^2 distribution is the conjugate prior on the variance of

a (one dimensional) Gaussian distribution. The mean, the variance, and the mode of the scaled inverse-χ^2 distribution are

$$\begin{aligned} \mathrm{E}(X) &= \frac{\nu}{\nu-2} s^2 \\ \mathrm{Var}(X) &= \frac{2\nu^2}{(\nu-2)^2(\nu-4)} s^4 \\ \mathrm{mode}(X) &= \frac{\nu}{\nu+2} s^2 \end{aligned}$$

Bibliography

[1] Affymetrix, Inc. *Statistical Algorithm Description Document*, 2002.

[2] J. Allemeersch, S. Durinck, R. Vanderhaeghen, P. Alard, R. Maes, K. Seeuws, T. Bogaert, K. Coddens, K. Deschouwer, P. Van Hummelen, M. Vuylsteke, Y. Moreau, J. Kwekkeboom, A. H. M. Wijfjes, S. May, J. Beynon, P. Hilson, and M. T. R. Kuiper. Benchmarking the CATMA microarray. A novel tool for Arabidopsis transcriptome analysis. *Plant Physiology*, 137:588–601, 2005.

[3] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, 96:6745–6750, 1999.

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

[5] F. Azuaje. A cluster validity framework for genome expression data. *Bioinformatics*, 18(2):319–320, 2002.

[6] R. Balakrishna, K. R. Christie, M. C. Costanzo, K. Dolinsky, S. S. Dwight, S. R. Engel, D. G. Fisk, J. E. Hirschman, E. L. Hong, R. Nash, R. Oughtred, M. Skrzypek, C. L. Theesfeld, G. Binkley, C. Lane, M. Schroeder, A. Sethuraman, S. Dong, S. Weng, S. Miyasato, R. Andrada, D. Botstein, and J. M. Cherry. Saccharomyces genome database. http://www.yeastgenome.org/yeast, July 2005.

[7] P. Baldi and S. Brunak. *Bioinformatics: the Machine Learning Approach*. Adaptive computation and machine learning. The MIT Press, second edition, 2001.

[8] P. Baldi and G. W. Hatfield. *DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling*. Cambridge University Press, 2002.

[9] P. Baldi and A. D. Long. A Bayesian framework for the analysis of microarray expression data: regularized *t*-test and statistical inference of gene changes. *Bioinformatics*, 17(6):509–519, 2001.

[10] Z. Bar-Joseph, D. K. Gifford, and S. Jaakkola. Fast optimal leaf ordering for hierachical clustering. *Bioinformatics*, 17(Suppl. 1):S22–S29, 2001.

[11] Y. Barash and N. Friedman. Context-specific bayesian clustering for gene expression data. *J. Comput. Biol.*, 9:169–191, 2002.

[12] T. Barrett, T. O. Suzek, D. B. Troup, S. E. Wilhite, W. C. Ngau, P. Ledoux, D. Rudnev, A. E. Lash, W. Fujibuchi, and R. Edgar. NCBI GEO: mining millions of expression profiles—database and tools. *Nucl. Acids Res.*, 33(Database issue):D562–D566, 2001.

[13] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *J. Comput. Biol.*, 6:281–297, 1999.

[14] S. Bergmann, J. Ihmels, and N. Barkai. Iterative signature algorithm for analysis of large-scale gene expression data. *Physical Review E*, 67:031902, 2003.

[15] T. De Bie, P. Monsieurs, K. Engelen, B. De Moor, N. Cristianini, and K. Marchal. Discovering regulatory modules from heterogeneous information sources. In R. B. Altman, T. A. Jung, T. E Klein, A. K. Dunker, and L. Hunter, editors, *Procedings of the Pacific Symposium of Biocomputing 2005*, pages 483–494, 2005.

[16] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Doughety, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, N. Hayward, and J. Trent. Molecular classfication of cutaneous malignat melanoma by gene expression profiling. *Nature*, 406:536–540, 2000.

[17] B. M. Bolstad, R. A. Irizarry, M. Åstrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19:185–193, 2003.

[18] A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoekert, J. Aach, W. Ansorge, C. A. Ball, H. C. Causton, T. Gaasterland, P. Glenisson, F. C. P. Holstege, I. F. Kim, V. Markowitz, J. C. Matese, H. Parkison, A. Robinson, U. Sarkans, S. Shulze-Kremer, J. Stewart, R. Taylor, J. Vilo, and M. Virgron. Minimum information about a microarray experiment (MIAME)—toward standards for microarray data. *Nature Genet.*, 29:365–371, 2001.

[19] G. Casella and E. I. George. Explaining the Gibbs sampler. *Am. Stat.*, 46(3):167–174, 1992.

[20] Y. Chen, E. R. Dougherty, and M. L. Bittner. Ratio-based decisions and the quantitative analysis of cDNA microarray images. *J. Biomedical Optics*, 2:364–374, 1997.

[21] Y. Cheng and G. M. Church. Biclustering of expression data. In *ISMB 2000 proceedings*, pages 93–103, 2000.

[22] R. J. Cho, M. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, D. J. Lockhart, and R. W. Davis. A genome-wide transcriptional analysis of mitotic cell cycle. *Molecular Cell*, 2:65–73, 1998.

[23] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *J. Amer. Stat. Assoc.*, 32:829–836, 1979.

[24] The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genet.*, 25:25–29, 2000.

[25] M. K. Cowles and B. Carlin. Markov chain Monte Carlo convergence diagnositics a comparative review. *J. Amer. Stat. Assoc.*, 91:883–904, 1996.

[26] F. Crick. On protein synthesis. In *Symposium of the Society of Experimental Biology*, volume 12, pages 138–163, 1958.

[27] F. Crick. Central dogma of molecular biology. *Nature*, 227:561–563, 1970.

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc.*, B 39:1–38, 1977.

[29] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Willey & Sons, Inc., second edition edition, 2001.

[30] B. P. Durbin, J. S. Hardin, D. M. Hawkins, and M. Rocke. A viariance-stablization transformation for gene-expression microarray data. *Bioinformatics*, 18:S105–S110, 2002.

[31] B. P. Durbin and D. M. Rocke. Variance-stabilizing transformations for two-color microarrays. *Bioinformatics*, 20(5):660–667, 2004.

[32] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95:14863–14868, 1998.

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

[34] N. Friedman. The bayesian structural EM algorithm. In *Proceedings of UAI'98*, 1998.

[35] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303:799–805, 2004.

[36] N. Friedman, L. Gettor, D. Koller, and A. Preffer. Learning probabilistic relational models. In *Proceedings of IJCAI'99*, 1999.

[37] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, 11:4241–4257, 2000.

[38] A. E. Gelfand and F. M. Smith. Sampling-based approaches to calculating maginal densities. *J. Amer. Stat. Assoc.*, 85:398–409, 1990.

[39] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–472, 1992.

[40] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 7:721–741, 1984.

[41] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leischand C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004.

[42] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci. USA*, 97(22):12079–12084, 2000.

[43] J. Geweke. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In J.M. Bernardo, J. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 4*. Oxford University Press, 1992.

[44] D. Ghosh and A. M. Chinnaiyan. Mixture modelling of gene expression data from microarray experiments. *Bioinformatics*, 18(2):275–286, 2002.

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

[46] J. A. Hartigan. *Clustering Algorithms*. Wiley Series i Probability. John Wiley & Sons, Inc., 1975.

[47] T. Hastie, R. Tibshirani, M. B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W. C. Chan, D. Botstein, and P. Brown. "Gene shaving" as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1(2):research0003.1–0003.21, 2000.

[48] W. K. Hasting. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 87:97–109, 1970.

[49] D. Heckerman. A tutorial on learning with Bayesian networks. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 301–354. The MIT Press, 1999.

[50] J. Herrero, A. Valencia, and J. Dopazo. A hierachical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, 17(2):126–136, 2001.

[51] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, 9:1106–1115, 1999.

[52] W. Huber, A. von Heydebreck, H. Sültmann, A. Poustka, and M. Vingron. Variation stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18:S96–S104, 2002.

[53] J. Ihmels, G. Friedlander, S. bergmann, O. Sarig, Y. Ziv, and N. Barkai. Revealing modular organization in the yeast transcriptional network. *Nat. Genet.*, 31:370–777, 2002.

[54] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4:249–264, 2003.

[55] M. I. Jorda, editor. *Learning in Graphical models*. The MIT Press, 1999.

[56] R. Jörsten and Bin Yu. Simultaneous gene clustering and subset selection for sample classification via MDL. *Bioinformatics*, 19(9):1100–1109, 2003.

[57] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: an Introcuction to Cluster Analysis*. John Wiley & Sons, Inc., 1990.

[58] M. Kellis, N. Patterson, M. Endrizzi, B. Birren, and E. S. Lander. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, 423:241–254, 2003.

[59] M. K. Kerr and G. A. Churchill. Bootstrapping cluster analysis: Assessing the reliability of conclusions from microarray experiments. *Proc. Natl. Acad. Sci. USA*, 98(16):8961–8965, 2001.

[60] M. K. Kerr, M. Martin, and G. A. Churchill. Analysis of variance for gene expression microarray data. *J. Comput. Biol.*, 7:819–837, 2000.

[61] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, 13:703–716, 2003.

[62] T. Kohonen. *Self-Organizing Maps*. Springer Series in Information Sciences. Springer, 1995.

[63] D. A. Lashkari, J. L. DeRisi, J. H. McCusker, A. F. Namath, C. Gentile, S. Y. Hwang, P. O. Brown, and R. W. Davis. Yeast microarrays for genome wide parallel genetic and gene expression analysis. *Proc. Natl. Acal. Sci. U.S.A.*, 94:13057–13062, 1997.

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

[65] L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Statistica Sinica*, 12:61–86, 2002.

[66] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, 298:799–804, 2002.

[67] C. Li and W. H. Wong. Model–based analysis of oligonucleotide arrays: expression index computation and oulier detection. *Proc. Natl. Acad. Sci. USA*, 98:31–36, 2001.

[68] J. S. Liu, A. F. Neuwald, and C. E. Lawrence. Bayesian models for multiple local sequece alignment and Gibbs sampling strategies. *J. Amer. Stat.*, 90:1156–1170, 1995.

[69] A. V. Lukashin and R. Fuchs. Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters. *Bioinformatics*, 17(5):405–414, 2000.

[70] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE Transactions on Computational Biology and Bioinformatics*, 1:24–45, 2004.

[71] G. J. McLachlan, R. W. Bean, and D. Peel. A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18(3):413–422, 2002.

[72] H. W. Mewes, C. Amid, R. Arnold, D. Frishman, U. Güldener, G. Mannhaupt, M. Münsterkötter, P. Pagel, N. Strack, V. Stümpflen, J. Warfsmann, and A. Ruepp. MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acid Res.*, 32 Database issue:D41–D44, 2004.

[73] Y. Moreau, F. De Smet, G. Thijs, K. Marchal, and B. De Moor. Functional bioinformatics of microarray data: From expression to regulation. *Proceedings of the IEEE*, 90(11):1722–1743, 2002.

[74] F. Naef, D. A. Lim, N. Patil, and M. Magnasco. DNA hybridization to mismatched templates: a chip study. *Phys. Rev. E. Stat. Nonlin. Soft. Matter. Phys.*, 65:040902, 2002.

[75] H. Parkison, U. Sarkans, M. Shojatalab, N. Abeygunawardena, S. Contrino, R. Coulson, A. Farne, G. G. Lara, E. Holloway, M. Kapushesky, P. Lilja, G. Mukherjee, A. Oezcimen, T. Rayner, P. Rocca-Serra, A. Sharma, S. Sansone, and A. Brazma. Arrayexpress—a public repository for microarray gene expression data at the EBI. *Nucl. Acids Res.*, 33(Database issue):D553–D555, 2005.

[76] D. Pinkel, R. Segraves, D. Sudar, S. Clark, I. Poole, D. Kowbel, C. Collins, W. Kuo, C. Chen, Y. Zhai, S. H. Dairkee, B. Ljung, J. W. Gray, and D. G. Albertson. High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nat. Genet.*, 20:207–211, 1998.

[77] J. Quackenbush. Computational analysis of microarray data. *Nature Reviews*, 2:418–427, 2001.

[78] J. Quackenbush. Microarray data normalization and transformation. *Nat. Genet.*, 32:suppl. 496–501, 2002.

[79] A. E. Raftery and S. Lewis. How many iterations in the Gibbs sampler? In J. M. Bernardo, Berger J., A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 4*. Oxford University Press, 1992.

[80] B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zitlinger, J. Schreiber, N. Hannett, E. Kanin, T. L. Vokert, C. J. Wilson, S. P. Bell, and R. A. Young. Genome-wide location and function of DNA binding proteins. *Science*, 290:2306–2309, 2000.

[81] D. M. Rocke and B. Durbin. A model for measurement error for gene expression arrays. *J. Comput. Biol.*, 8:557–569, 2001.

[82] D. M. Rocke and B. Durbin. Approximate variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics*, 19(8):966–972, 2003.

[83] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467–470, 1995.

[84] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[85] M. P. Scott, P. Matsudaira, H. Lodish, J. Darnell, L. Zipusky, C. a. Kaiser, A. Berk, and M. Krieger. *Molecular Cell Biology*, volume 5th edition. W. H. Freeman, 2003.

[86] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: Identifying regulatory modules and their condition-specific regulators for gene expression data. *Nat. Genet.*, 34(2):166–176, 2003.

[87] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller. Rich probabilisitc models for gene expression. *Bioinformatics*, 17(Suppl. 1):S243–S252, 2001.

[88] E. Segal, H. Wang, and D. Koller. Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, 19:i264–i272, 2003.

[89] Q. Sheng, Y. Moreau, and B. De Moor. Biclustering microarray data by Gibbs sampling. *Bioinformatics*, 19(Suppl. 2):II196–II205, 2003.

[90] G. Sherlock. Analysis of large-scale gene expression data. *Curr. Opin. Immunol.*, 12:201–205, 2000.

[91] F. De Smet, J. Mathys, K. Marchal, G. Thijs, B. De Moor, and Y. Moreau. Adaptive quality-based clustering of gene expression profiles. *Bioinformatics*, 18(5):735–746, 2002.

[92] G. K. Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):Artical 3, 2004.

[93] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehesive identification of cell cycle-regulated genes of the yeast saccaromyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9:3273–3297, 1998.

[94] J. D. Storey and R. Tibshirani. Statistical significance for genomewide studies. *Proc. Natl. Acad. Sci. USA*, 100(10):9440–9445, 2003.

[95] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA*, 96:2907–2912, 1999.

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

[97] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. *Nat. Genet.*, 22:281–285, 1999.

[98] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005.

[99] G. Thijs. *Probabilistic methods to search for regulatory elements in sets of coregulated genes*. PhD thesis, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium, 2003.

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

[101] P. Törönen, M. Kolehmainen, G. Wong, and E. Gastrén. Analysis of gene expression data using self-organizing maps. *FEBS Letters*, 451:142–146, 1999.

[102] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles*. Applied mathematics and computaion. Addison-Wesley Publishing Company, 1979.

[103] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[104] G. C. Tseng, M. K. Oh, L. Rohlin, J. C. Liao, and W. H. Wong. Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects. *Nucleic Acids Res.*, 29:2549–2557, 2001.

[105] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[106] V. G. Tusher, R. Tibshirani, and G. Chu. Significant analysis of microarrays applied to the ionizing radiation response. *Proc. Natl. Acad. Sci. USA*, 98(9):5116–5121, 2001.

[107] J. H. Ward. Hierarchical grouping to optimize an objective function. *Jour. Amer. Stat. Assoc.*, 58:239–244, 1963.

[108] X. Wen, S. Fuhrman, G. S. Michaels, D. B. Carr, S. Smith, J. L. Bakker, and R. Somogyi. Large-scale temporal gene expression mapping of central nervous system development. *Proc. Natl. Acad. Sci. USA*, 95:334–339, 1998.

[109] B. L. Wiens. When log-normal and gamma models give different results: a case study. *Am. Stat.*, 53:89–93, 1999.

[110] Y. Woo, J. Affourtit, S. Daigle, A. Viale, K. Johnson, J. Naggert, and G. Churchill. A comparison of cDNA, oligonucleotide and Affymetrix GeneChip gene expression microarray platforms. *Journal of Biomolecular Techniques*, 15:276–284, 2004.

[111] C. F. J. Wu. On the convergence poperties of the em algorithm. *Annals of Statistics*, 11:95–103, 1983.

[112] Y. H. Yang, M. J. Buckley, and T. P. Speed. Analysis of cDNA microarray images. *Briefings in Bioinformatics*, 2:341–349, 2001.

[113] Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acid Res.*, 30(4):e15, 2002.

[114] J. M. Yeakley, J. Fan, D. Doucet, L. Luo, E. Wickham, Z. Ye, M. S. Chee, and X. Fu. Profiling alternative splicing on fiber-optic arrays. *Nat. Biotechnol.*, 20:353–358, 2002.

[115] E. Yeoh, M. E. Ross, S. A. Shurtleff, W. K. Williams, D. Patel, R. Mahfouz, F. G. Behm, S. C. Raimondi, M. V. Relling, A. Patel, C. Cheng, D. Campana, D. Wilkins, X. Zhou, J. Li, H. Liu, C. Pui, W. E. Evans, C. Naeve, L. Wong, and J. R. Downing. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1:133–143, 2002.

[116] K.Y. Yeung, C. Fraley, A. Murua, A.E. Raftery, and W.L. Ruzzo. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987, 2001.

[117] K.Y. Yeung, D.R. Haynor, and W.L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318, 2001.

Index

Curriculum vitae

Qizheng Sheng was born in Shanghai, People's Republic of China, on May 13, 1977. She obtained a Bachelor's degree in engineering (majoring in automatic control) from Shanghai University, China, in July 1999. She worked as an engineer for the R&D department of SVA (Shanghai Video and Audio) Group during September 1999 till September 2000. In September 2001, she obtained a Master degree in engineering (majoring in data mining and automation) from the Katholieke Universiteit Leuven, Belgium. Since October 2001, she has been pursuing her doctorate in bioinformatics at the Department of Electrical Engineering of Katholieke Universiteit Leuven, under the supervision of Professor Bart De Moor and Professor Yves Moreau.