

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

GIBBS SAMPLING ON BAYESIAN MODELS FOR THE BICLUSTEING PROBLEMS OF 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 THE BICLUSTEING PROBLEMS OF MICROARRAY DATA

Jury:
Prof. dr. ir. x, voorzitter
Prof. dr. ir. B. De Moor, promotor
Prof. dr. ir. Y. Moreau, co-promotor
Prof. dr. ir. J. Vanderwalle
Prof. dr. ir. x
Prof. dr. ir. x
Prof. dr. ir. x
Prof. dr. ir. x
Prof. P. Rouzé (INRA, VIB, U.Gent)

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

door

Qizheng SHENG

November 2005

©Katholieke Universiteit Leuven – Faculteit Toegepaste Wetenschappen
Arenbergkasteel, B-3001 Heverlee (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

D/2003/7515/24

ISBN 90-5682-414-7

Contents

Contents i

Publication List vii

1 Introduction 1

1.1 Biological background 1

1.2 Technological background 4

1.3 Biclustering problems for microarray data 4

1.4 Bayesian models for microarray data 5

1.5 Gibbs sampling for Bayesian models on microarray data 6

1.6 Organization of the thesis 7

2 Microarray: a gene expression profiling technology 9

2.1 Microarray technologies 10

2.1.1 cDNA microarrays 10

2.1.2 Affymetrix GeneChip 11

2.1.3 Comparison between spotted arrays and *in situ* synthesized arrays 13

2.2 Noise and artifacts in microarray data 14

2.3 Preprocessing of microarray data 15

2.3.1 Quality assessment 15

2.3.2 Background correction 16

2.3.3 Normalization 18

2.4 Specific characteristics of microarray data 22

3 Clustering microarray data **25**

3.1 Some preliminaries . . . 27
3.1.1 Clustering genes vs. clustering experiments . . . 27
3.1.2 Standardization of gene expression profiles . . . 27
3.1.3 Distance metrics . . . 27
3.2 Classical clustering methods . . . 29
3.2.1 Hierarchical clustering . . . 29
3.2.2 *K*-means clustering . . . 31
3.2.3 Self-organizing maps . . . 32
3.3 Model-based approaches for gene expression data . . . 33
3.3.1 Mixture model of normal distributions . . . 34
3.3.2 Mixture model of t distributions and mixture of factor analysis . . . 35
3.4 Biclustering algorithms . . . 37
3.4.1 Gene shaving . . . 39
3.4.2 Cheng and Church's approach . . . 39
3.4.3 Probabilistic relational models for microarray data . . . 40
3.5 Assessing cluster quality . . . 41

4 Gibbs sampling on Bayesian hierarchical models **45**

4.1 Gibbs sampling . . . 47
4.1.1 The Markov chain property . . . 48
4.1.2 The Monte Carlo property . . . 49
4.1.3 Checking the convergence . . . 50
4.2 Bayesian hierarchical model for the biclustering problem . . . 52
4.2.1 Bayesian hierarchical models . . . 52
4.2.2 Biclustering: an incomplete-data problem . . . 53
4.3 Gibbs sampling for the biclustering problem . . . 59
4.3.1 The target posterior joint distribution . . . 60
4.3.2 The manipulation of Λ^r and Λ^c . . . 61
4.3.3 Full conditional distributions of the missing data and the structural variables . . . 62
4.3.4 The Gibbs sampling scheme for the biclustrering problem 64

4.3.5 From samples to the final pattern 65
4.3.6 Multiple biclusters 66

5 Biclustering experiments in microarray data 67
5.1 The discretization of microarray data 68
5.2 The Model 69
5.3 Full conditional distributions 70
5.4 Importance of the priors 75
5.5 Biclustering for global pathological pattern discovery 76
5.5.1 Construction of priors 76
5.5.2 Experiments on synthetic data 77
5.5.3 Case study: biclustering expriments on leukemia patients 84
5.6 Query-driven biclustering for pathological pattern discovery . . 94
5.6.1 Construction of priors 94
5.6.2 Experiments on synthetic data 98
5.6.3 Case study: query-driven biclustering of leukemia patients 104

6 Biclustering genes based on microarray data 109
6.1 Model structure 111
6.2 Gaussian-Wishart model 112
6.3 Full conditional distributions 113
6.4 Construction of the priors 117
6.5 Biclusters for transcriptional regulatory modules 119
6.6 Experiments on synthetic data 119
6.7 Transcriptional module discovery on *Saccharomyces cerevisiae* . . 125

7 Discussions and conclusions 137
7.1 Achievements of the work 137
7.2 Limitations of the work 139
7.3 Future directions 140

Bibliography 143

Curriculum vitae 153

Notations

Mathematical notations

X	scaler random variable
x	specific value of random variable X
$\mathbf{X}_m$	set of random variables with set-length equals m
$\mathbf{x}$	specific values for the set of random variables $\mathbf{X}_m$
$\mathcal{X}$	set
$p(\cdot)$	density function
$P(\cdot)$	probability distribution
$\mathrm{E}(\cdot)$	estimation

Fixed symbols

n	number of rows in a microarray data matrix
m	number of columns in a microarray data matrix

Acronyms

BIC	Bayesian information score
cDNA	complementary DNA
CPD	conditional probability distribution
EST	expressed sequence tag
EM	expectation-maximization
HMM	hidden Markov model
IM	ideal mismatch
IQR	interquartile range
MAP	maximum a posterior
MM	mismatch (probe)
mRNA	messenger RNA
PCA	principle component analysis

PM	perfect-match (probe)
PRM	probabilistic relation model
RMA	robust multichip average
SB	(biweight) specific background
SOM	self-organizing maps
VSN	variation stabilization 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, Beglium), 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, Yves Moreau, and Bart De Moor, Intoduction to bioinformatics, *First Annual Conference on Life System Modelling and Simulation*, Oct. 21- 23, 2004, Shanghai, China.

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

Chapter 1

Introduction

1.1 Biological background

The study of molecular biology is centered around the dogma which can be explained as follows. 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 nucleotide at the same position on the other strand by a strict rule of basic pairing, i.e., guanine can only be paired with cytosine and vice versa, while adenine (A) can only be paired with thymine (T), and vice versa (see Figure 1.1). Genes are fragments of the DNA sequence that carry such information.

The first step of a protein synthesis procedure is the transcription of its corresponding gene, to a message 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 procedure 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 in a ribosome where 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. In this way, a peptide chain is assembled by the ribosome. The peptide chain is later folded into the resulting protein.

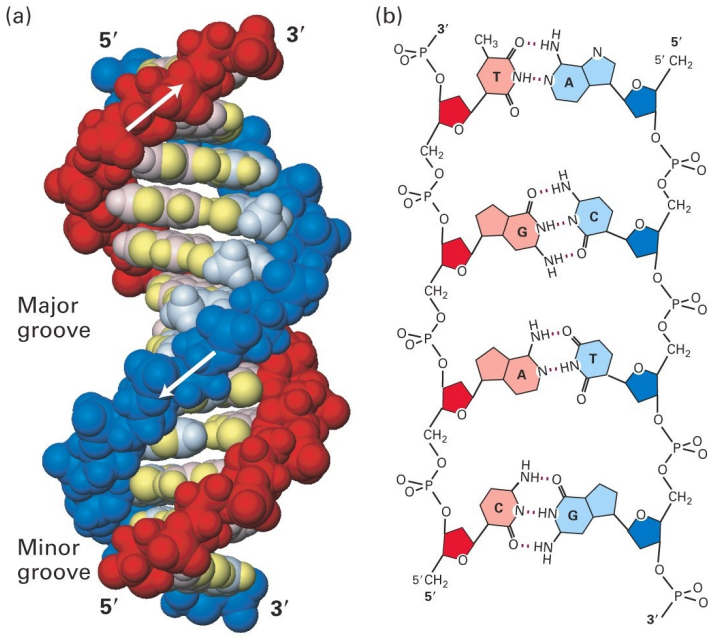


Figure 1.1: (A): 3D illustration of the structure of DNA molecules. (B) Rule of base pairing for the four nucleotides—basic components of DNA molecules. The figure is obtained from Scott *et al*. (2003).

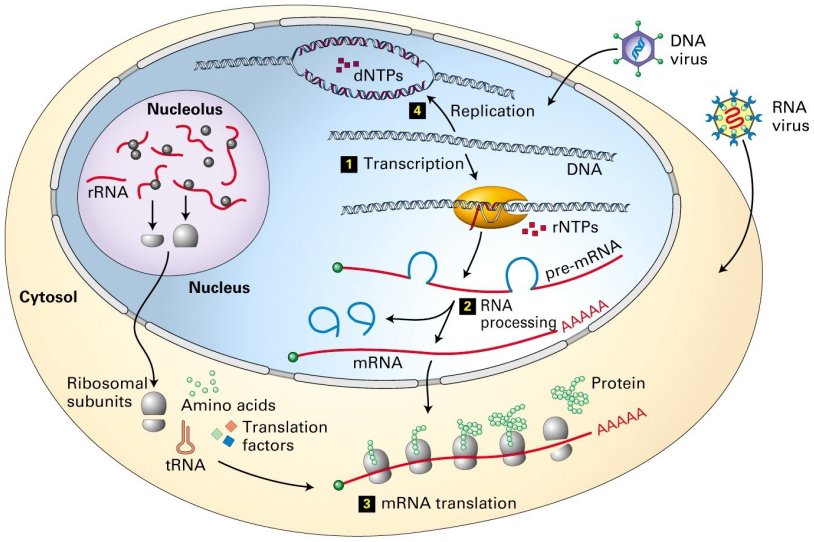


Figure 1.2: Central dogma of molecular biology: the transcription of genes (DNA segments) to mRNAs, and the translation of mRNA to proteins. The figure is obtained from Scott *et al*. (2003).

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 procedure 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 given by Scott *et al*. (2003) [76].)

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 (regulatorial) 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 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 the help from other metabolites. Metabolome is an area of study that concerns the interactions and dynamics of all the metabolites in a cell.

1.2 Technological background

During the past few years, microarray technology has emerged as an effective technique to measure the level of expression of thousands of genes in a single experiment. Nowadays, a microarray chip is able to take a snapshot of the gene expression levels of the whole genome on an area of a couple of square centimeters. Putting together data obtained by performing microarray experiments under different experimental conditions (e.g., tissues, time points, or environmental conditions), expression profiles are obtained for the genes measured on the microarray chips. This technology has been become a major attraction for molecular biologists, in a 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 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 concerns a data mining technology that helps biologists to extract the essential question from microarray data after normalization is performed. Yet, microarray data has several characteristic features that cannot be deprived during the normalization procedure. First, microarray data comprises a huge amount of noise introduced by the underlying biological process as well as the measuring procedure. Secondly, microarray data often has a asymmetric dimension. Microarray data is often put in a matrix whose rows represent the genes and whose columns represent the experimental conditions. While the number of genes can easily reach tens of thousands, the number of experimental conditions often only comprises no more than hundreds.

1.3 Biclustering problems for microarray data

A generalized formulation of the core question of modern molecular biology research is to the function of the genes. Throughout the years, this purpose has evolved from understanding the individual roles that a gene play in the cell by studying the gene by 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, microarray data records the activities of the genes in cooperation so that information about different functional relationships between the genes are reserved.

For medical applications where the the conditions of a microarray studies 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, algorithms are desired to cluster tumor samples based on their gene expression levels for a subset of genes based on microarray data; 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.

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. In the early days of the applications of microarray technology [28, 84], experiments were often conducted under a limited number of homogeneous experimental conditions measured at different time points. Clustering algorithms are performed for these data assuming that genes sharing functional similarities have similar gene expression levels under all the measured conditions. I refer to this type of clustering algorithms as the biclustering algorithms.

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 microarry experiment (MIAME) [16], it is, nowadays, plausible to retrieve data from publicly available repositories of microarray experiments, such as ArrayExpress [67] and GEO [11], and perform analysis for a microarray data set whose experiments form 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. Instead, gene expression profiles should be clustered only under a subset of conditions. Biclustering algorithms, therefore, aims in the clustering of genes and in the mean time identify the working conditions of these genes for the same function that they share.

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. 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. However, the probability distribution provided by these models usually contains many modes, because of the complexity of the underlying biological

process. In clustering, these modes correspond to the different clusters that can be identified in the data. The largest clusters (which are easiest to identify) are often 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 them 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?"

To study the concerted gene activities in a cell and the different relationships between them calls out 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 answering both of the questions. The inference procedure of Bayesian model learning highly resembles that of the human learning process. It allows the introduction of prior knowledge in which form soft queries can be imposed to direct the discovery, and it provides a systematic base that information from different sources can be integrated.

1.5 Gibbs sampling for Bayesian models on microarray data

The likelihood of microarray data is usually highly complex because of the complexity of in the underlying biological process and the non-negligible experimental noise. By introducing a prior, methods based on Bayesian models help to zoom into the local area of interest of the likelihood landscape, and raise the corresponding area in the posterior distribution. However, the other modes in the likelihood function are still reserved as local modes in the posterior distribution. To enhance the ability for the algorithm to find the global maximum solution, which gives answer to the biological question of interest, we choose to use Gibbs sampling [17] for the parameterizations of the Bayesian model that describes the microarray data. The Gibbs sampling procedure produces samples that pictures the posterior distribution as a whole, and consequently the global maximum is decided by Monte Carlo integration of the samples. An alternative way to solve the problem based on the likelihood landscape is by using EM method. However, EM is a method that "climbs" in the posterior distribution and can get stuck in local maxima. To find the global, EM is usually performed for several times and the solution with the greatest posterior probability is chosen. In our experience, the Gibbs sampling procedure turns out to be more efficient to find the global maximum for the query-driven biclustering problem of microarray data, taken into account the massive amount of modes and the fact that it is never easy to decide how many runs of EM can guarantee the discovery of the regional global maximum.

1.6 Organization of the thesis

This thesis is organized as follows. In Chapter 2, we overview the working mechanisms of various microarray technologies, and the pitfalls that lie in these technologies. A review of the main trend quality control measures and normalization techniques that help to minimize the systematic noise in microarray data is then followed. We summarize the chapter by enumerating the characteristics of normalized microarray data of which still require full awareness in designing data analysis tools for such data.

Since the emergence of microarray technology, clustering techniques has been recognized as a useful tool for the analysis of microarray data. Standard clustering methods, such as hierarchical clustering, *K*-means, and self-organizaton map (SOM), were applied directly to microarray data and dominated the early papers for microarray data analysis [28, 84]. With the growing acquaintance with the characteristics of microarray data, it became clear that tailored clustering algorithms are desired to better such analysis. Throughout the years, numerous techniques have been developed for clustering microarray data. Furthermore, This still remains a active field of research. In Chapter 3, we review serveral popular clustering techniques, and introduce further the need for biclustering algorithms. We also account several exisiting biclustering algorithm. The chapter is concluded by a checklist for evaluating the cluster quality.

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

Chapter 5 explicate the application of our methodology to the problem of biclustering experiments. We discuss the application in two scenarios. The first one 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 assume that tumor examples a subtle traits are available. We discuss how to extract information from these examples and use it as prior knowledge for the biclustering discovery. We demonstrate the effectiveness of our algorithm on two data sets of leukemia patients.

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

Finally, in Chapter 7, we conclude the achievement of our work, and propose some future challenges for further research for this topic.

Chapter 2

Microarray: a gene expression profiling technology

A microarray is a chip (i.e. array) on whose surface single-stranded DNAs (called probes) are listed in grid. When exposed to cell culture obtained for a certain biological study, a microarray is able to capture a snapshot of the transcription levels (i.e., the mRNA levels) of hundreds of thousands of genes (nowadays, even a whole genome) under the experimental condition. By performing microarray experiments under different conditions, biologists are able to simultaneously monitor the behavior of the genes on the transcriptional level. The transcriptional behavior of a gene is thus described by its expression profile, which is the expression levels of the gene under different experiment conditions.

Besides gene expression profiling, other applications of microarrays include revealing genome-wide location of DNA-bound proteins [72], genome-wide analysis of DNA sequence copy number variation (the specific microarray technology is called comparative genomic hybridization) [68], and monitoring alternative splicing of pre-mRNA on a genome scale [101], among the others. However, we will limit our discussion to the application of microarray in gene expression analysis to keep in accordance with 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 labelled, 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) labelled 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 which removes the systematic noise before any further analysis can be carried out.

2.1 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 20 to 80 bases in length) or cDNAs, are attached to glass or nylon slides. On the contrary, single stranded DNAs are synthesized directly to 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.

2.1.1 cDNA microarrays

The probes on a cDNA microarray are single-stranded cDNAs fragments genes. These cDNAs are typically of 100 to 5000 bases long. While the cDNAs are prepared (reverse transcribed from mRNAs) by individual labs in the early days of cDNA microarray technology, nowadays pre-synthesized cDNA clones are commercially available and are usually derived from reference banks of expressed sequence tags (ESTs), each of which is documented and 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 mi-

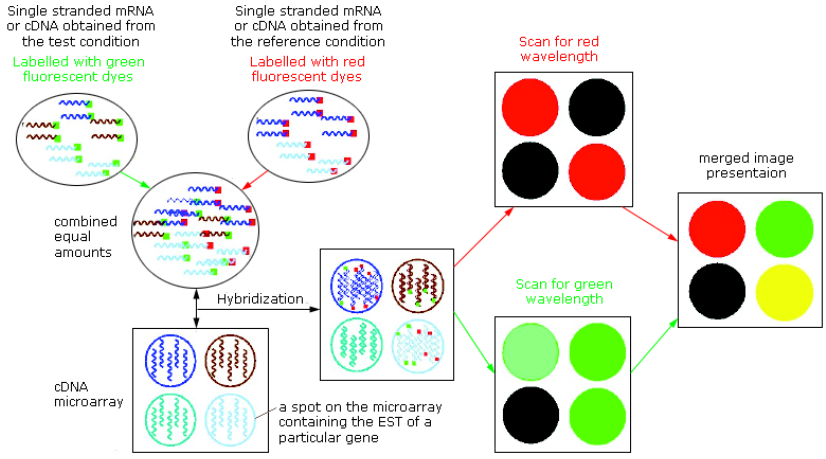


Figure 2.1: Measuring gene expression values by cDNA microarray. Note that the detected signal intensities of the fluorescence dyes (typically Cy3 and Cy5) are often converted by software into a red-and-green-dye presentation for the readout microarray data. Here, for the convenience of illustration, we pretend that red and green dyes are attached to the mRNA or cDNA samples.

croarray. One of the condition is referred to as the reference condition, and the set 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 mRNA/cDNA samples obtained in experimental condition is called the test sample. The reference sample is labelled with fluorescence Cy3, and the test sample is labelled with fluorescence Cy5, or vice versa. After the hybridization, the chip is scanned for both 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 labelled samples effectively removes the array-to-array variabilities in cDNA microarray technology [98]. Figure 2.1 provides an overview for the whole measuring procedure using cDNA microarray.

2.1.2 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 GeneChips.

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 contain 11 to 16 *probe pairs*, which identifies different regions

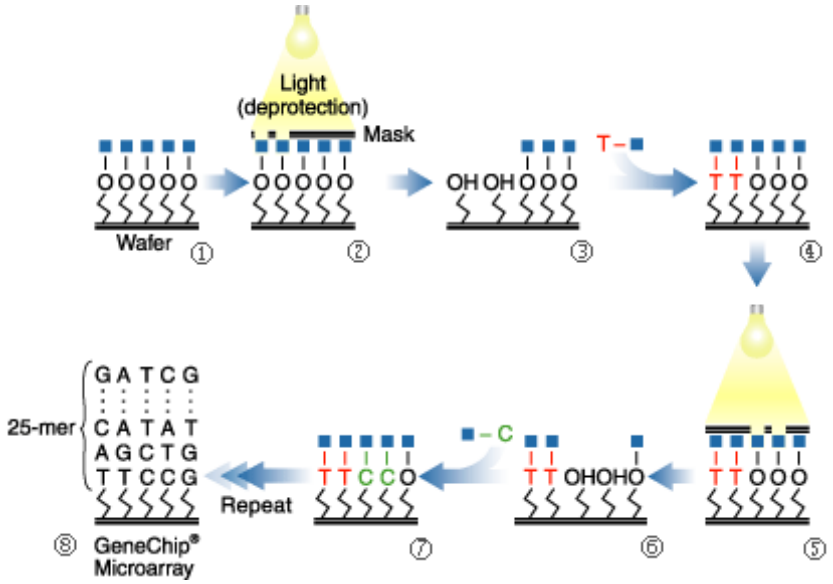


Figure 2.2: The manufacturing of Affymetrix GeneChip (picture source from Affymetrix http://www.affymetrix.com/technology/manufacturing/index.affx). (1) Linker molecules which can be activated by ultraviolet light are attached to the surface of a chip. (2)(5) A photo-protected mask with windows open 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 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 guanosine, and is continued until the probes reach their full length, usually of 25 bases long.

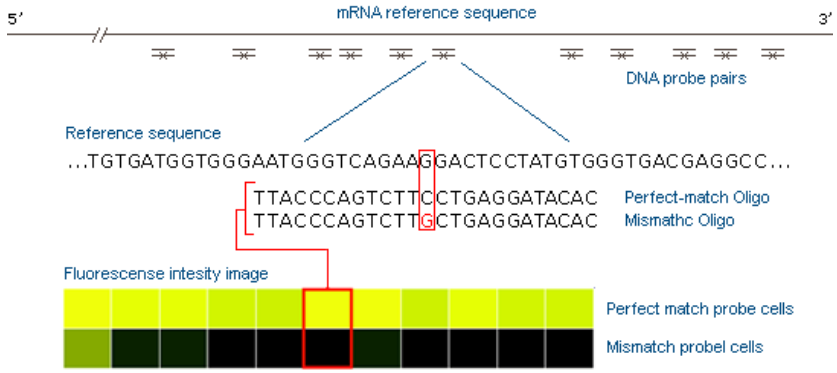


Figure 2.3: Probe design strategy of Affymetrix GeneChip.

of the target gene. The choice for the sequence 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 from the PM probe by only a single base in the center of the oligonucleotide. All the probes (i.e., oligonucleotides) are of 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 copy of mRNA is required, and the gene expression is measured as absolute value instead of ratios.

2.1.3 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 synthesizing and purifying of the biomoledules 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 GeneChips are expensive in price. On the other hand, spotted arrays are usually 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.

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

Speaking of fluorescence dyes, for cDNA microarrays (and other two-channel microarray technologies in general), the difference between the labelling 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 unappropriate 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.

2.3 Preprocessing of microarray data

Due to the vast amount 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. Thus, the goal of the preprocessing procedure is to check and remove as much as possible the systematic noise (i.e., due to such as array effect, plate effect and pin effect for cDNA microarrays, and probe effect for the Affymetrix GeneChips) 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.

2.3.1 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 quantitively 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 spacial 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 spacial 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 cases log intensity

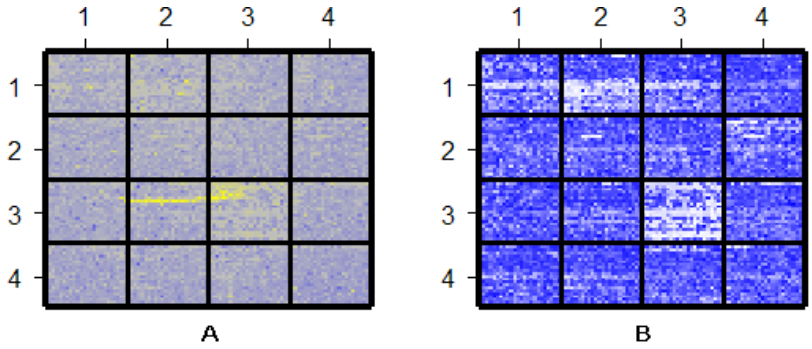


Figure 2.4: Log intensity plots of a cDNA microarray (i.e., array 81 in the "swirl" data from BioConductor package "marray"). (A) Log intensity ratios between the two channels for each spot on the array, a color toward yellow indicates a higher value while a color toward blue indicates otherwise. The figure indicates that there might be a contamination on the surface of the microarray. (B) Average log intensities over 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).

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 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.

2.3.2 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 [99].

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 [99]. 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.

However, the substraction of these measured background signals provided by either of the platforms has been under debate. There is evidence showing that

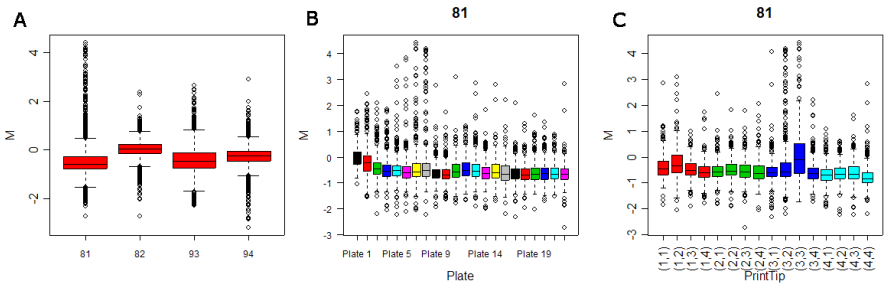


Figure 2.5: Box-plots of the "swirl" data (from BioConductor package "marray"). (A) Box plot of the log intensity ratios between the two channels (y-axis) of the spots, calculated for each array (x-axis) in the data. (B) Box plot of the log intensity ratios between the two channels (y-axis) of the spots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

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 [99, 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 [66]. 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 [59].

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 [47].

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 degrade 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 a exponential signal component. The normal distribution is truncated at zero in order to avoid negative background signals. The model is fit by the expression levels obtained by the PM probes. See [47] for more details.

2.3.3 Normalization

The main step to calibrate microarray data so as to remove as much as possible systematic noise on each array is carried out in normalization procedure. 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 [69] in the sense that up-regulated genes have expression ratios between one and infinity, while down-regulated 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 [18]. 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 [70], rank invariant methods [93], ratio statistics [18], and analysis of variance models [53]), in terms of correcting systematic biases mainly caused by dye effects [70, 100]. It is observed that the log ratios between the red and green channels of the data is 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 [100], 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) [21] to the MA-plot, and subtract 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 a zero-mean, and thus remove the spacial 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 GeneChips, the task of normalization is to remove the array effects and the probe effects. Both linear and non-linear 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 GeneChips data is quantile normalization [15], 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 are 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. 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 [94] to a linear model describing the log intensities as an 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) [47].

Besides the platform dependent normaliztion methods mentioned above, the variation stabilization normalization (VSN) [46, 26], a normaliztion method that works both for cDNA microarrays and Affymetrix GeneChips is receiving increasing attention. It combines the background correction, the non-linear transformation, and the normalization procedures together. The motivation of the method is to solve 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 [73]. 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 [73] 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 [26]. Therefore, by using VSN, the variance of the intensities becomes approximately independent fo the mean [46], and data in the whole range can be reserved 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 [74] and the *arsinh* function [46]. However, the basic form of these models (i.e., transformations) is the same,

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

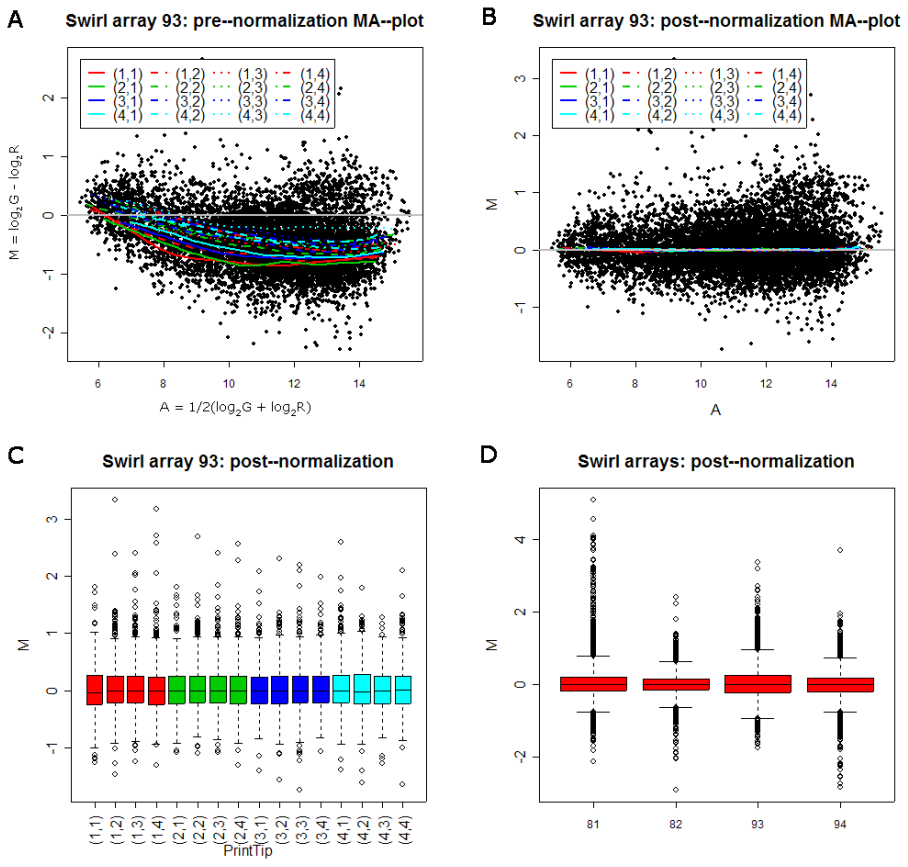


Figure 2.6: Lowess normalization of the "swirl" data (provided by BioConductor package "marray") (A) MA-plot showing the intensity dependent effects on the log ratios obtained for Array 93. The x-axis present the intensity of the genes by the A values, which are the average log intensities of the red (R) and green (G) channels. The y-axis gives the M values of the genes, which are the log-ratios of the two channel intensities. Different line types illustrates 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.

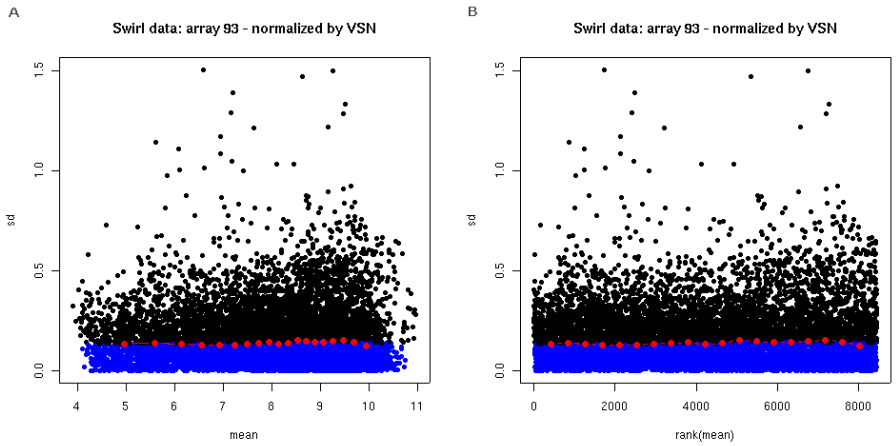


Figure 2.7: Result of VSN on the swirl data for Chip 93. The red dots depict the running median estimator. If there is no variance-mean dependence, the line formed by the red dots should be approximately horizontal. Each dot in the plots represent a gene. In each plot, the y-axis represent the standard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

where y is the measured intensity, and a is specified individually by different realizations of the method [46, 26, 27]. In addition, these different realizations are all motivated by the model that describes the raw microarray data (y) by two terms [73], 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 independetly 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 intesities,

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

where s_{η}^2 is the variance of e^{η}, and s_{ϵ}^2 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 proportion of various available approaches 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 are related to the specific biological question under study. Efforts in investigating the optimal normalization method are well deserved for each study.

2.4 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 expression profile of the gene. While the number of rows of the matrix can contain hundreds of thousands of genes, the dimension of the columns is much smaller. Current cost for a microarray chip limits the number of experiments in a study. Most labs can afford studies done under around 10 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 noises in 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 2.2) 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 [92] 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 assumably 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 microarry study. A conventional way is to remove the gene expression profiles from the data set that do not satisfy some simple criteria [28]. 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

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, 95, 85, 83]. 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 experiment 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 (i.e. coexpression) 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 [28], *k*-means [40] and self-organizing maps (SOM) [55]) applied to gene expression profiles were mostly developed outside biological research. Although encouraging results have been produced [84, 87, 86], some of their characteristics often complicate their use for clustering expression data [81]. 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 into a cluster. In general, a considerable number of genes included in the microarray experiment do not contribute to the biological process studied, 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. Finally, the computational and memory complexity of some of these algorithms often limit the number of expression profiles that can be analyzed at once.

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

A second generation of clustering algorithms have 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 [103, 39, 63], the self-organizing tree algorithm [44], quality-based algorithms [45, 82], simulated annealing [61], the cluster affinity search technique [12], and biclustering algorithms [19, 80, 41, 10]. 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 [39, 61, 103]).

In particular, probabilistic models have become a popular choice for the analysis of microarray data [103, 63, 78] 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 variabilities in the biological system.

In this chapter, we first give an general reminder of the use of classical clustering methods, namely hierarchical clustering, *k*-means clustering, and SOM, on microarray data. We then particularly focus the discussion of clustering algorithms for microarray data analysis on the model-based algorithms. Finally, we provide an overview on the current status of researches on biclustering algorithms for microarray data. To conclude this chapter, a list of popular methods to assess the quality of the clusters for microarray data is provided.

3.1 Some preliminaries

Before we dive into the detailed discussion of various clustering algorithms, there are yet a couple of common issues shared by different clustering techniques.

3.1.1 Clustering genes vs. clustering experiments

So far, we have addressed the clustering problem of microarray data only as an analysis to find genes that behave similarly over the experimental conditions. Instead of clustering genes, we can also cluster experimental conditions, where the task is to find groups of experiment conditions (which can be, for example, tumor samples) across which the all the genes behave similarly. This types of clustering can be helpful for problems like histopathological discovery of tumors whose types are primarily unknown.

To put the clustering problems in a mathematical language, when we cluster genes, each expression profile is often represented as a data point (or a vector, pointing from the origin) in a high-dimensional space whose dimension equals to the number of experiments presented in the study. The Euclidean axis of the data point is given by its expression value under every experiment. Similarly, when experiments are to be clustered, it is the experiments that are presented as data points (or vectors) in a Euclidean space whose dimension equals the number of genes in the data set. In both of the cases, the problem then is to find clusters of the data points in these high dimensional spaces. We will based our discussion hereafter mainly on the clustering of gene expression profiles, but the same methodologies can also be applied to the clustering of experiments.

3.1.2 Standardization of gene expression profiles

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 [69].

3.1.3 Distance metrics

Depending on the way to define a cluster, clustering methods can be divided into two types – model-based clustering methods and distance-based clustering

methods. Model-based clustering methods consider the data points in the high-dimensional space under study as a whole. Their aim is to discover the underlying structure of these data points. To be more specific, model-based clustering algorithms assume that the data points in the high-dimensional space as generated by a mixture of probabilistic models with different parameters. Each of these models is thus defined as a cluster.

Distance-based clustering methods (to which most of the classical clustering methods, such as hierarchical clustering, k-means, and SOM belong), on the other hand, explore the pairwise relation between the data points in the high-dimensional space under study. In their case, a cluster contains data points whose pairwise 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 is the cosine 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 it means that two expression profiles to be "coexpressed" [28].

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].

4. *Jackknife correlation:* New techniques of distance measurements are developed to overcome some of the shortcomings of the classical distance metrics. One of the examples is the jackknife correlation [45], which is an improvement for Pearson correlation measure. Although Pearson correlation gives high score to coexpressed genes, it is not robust to outliers.

An extreme case of a false positive is, for example, when two genes are totally unrelated under all but one experimental conditions, where they both have a high peak. These high peaks then pull the vector presentations of the two profiles to the same direction, and therefore the two genes will have a high Pearson correlation. Jackknife correlation increases the robustness to single outliers by computing a collection of all the possible leave-one-(experiment)-out Pearson correlation between two genes and then select the minimum of the collection as the final measure for the correlation.

3.2 Classical clustering methods

3.2.1 Hierarchical clustering

The first introduction of hierarchical clustering to the world of biology was its application to the construction of phylogenetic trees. Early applications of the method to gene expression data analysis [28, 84] have proved its usefulness.

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

Concerning the clustering procedure, 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 [28]). In agglomerative clustering [28], 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 is assembled. 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 [50] or when the number of clusters (say, k) presented in the data is known in advance [25] (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, a full dendrogram is usually desired by biologists. 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 pairwise distance between clusters. Note that the final cluster partition is thus rather arbitrary.

Because the agglomerative clustering is more commonly used in microarray

data analysis, we will discuss the method in more details. For divisive clustering, see [40, 50] for more information, and for its application on gene expression data, see [3].

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 [50]. 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 [86] rather than that of grouping expression profiles.

Distance measure between two clusters

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 [25]. 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 pairwise distance between clusters, Ward's method [96] 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 [25]. Ward's method, although less well-known, often produces the most satisfactory results.

Visualization of the results

A heat map 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 microrray data. The length of a branch in the dendrogram is proportional to the pairwise distance between the clusters. Importantly, the leaves of the dendrogram, and accordingly the rows of the heat map, 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 heat map. However, the time complexity of such an optimal organization of the dendrogram is $O(2^{N-1})$ (because for each of the $N-1$ merging steps there are two possible orders to arrange the concerned clusters). Yet, 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 [28] by weighting genes using combined source of information, and then placing the genes with lower average weight earlier in the final ordering. Further, [9] reported a dynamic programming method that helps to reduce the time and memory complexities for solving the optimal leaf-ordering problem.

3.2.2 *K*-means clustering

K-means clustering [40] is a simple and widely used partitioning method for data analysis. Its helpfulness in discovering groups of coexpressed genes has been demonstrated [87].

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 [91]. We will show in Section 3.3.1 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 [65]. 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. 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 principle component analysis (PCA) to provide prior knowledge on the number and the means of the clusters was proposed in [69].

3.2.3 Self-organizing maps

SOM [55] 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 during 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 an SOM – it is represented by the prototype vector in the input space and the neuron on the output space. In this way, an 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 [86, 90]. 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 [81].

Because of the advantage in visualization, choosing the geometry of nodes for an 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 an 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 [90]. 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 [55].

3.3 Model-based approaches for gene expression data

Model-based clustering [40] 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 [103, 39, 63],comparing with those are introduced in Section 3.2.

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, and at the mean time, to find out the parameters for each of these distributions.

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 of 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 is guaranteed.

In the following, we discuss respectively the work of Yeung *et al.* [103] where a mixture model of normal distributions is used for the cluster analysis of microarray data, and the work of McLachlan *et al.* [63] who use a mixture of factor analysis for clustering experiments.

3.3.1 Mixture model of normal distributions

When multivariate normal distributions are used, each cluster is represented by a hypersphere or a hyperellipse in the data space. The mean of the normal distribution gives the center of the hyperellipse, 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, and the eigenvalues specifying the shape and the volume of the cluster. 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 [103, 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 [97]. 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) [75] for each model.

Yeung *et al.* [103] reported good results of such analysis as described above using their MCLUST software [29] 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 3.2.2 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 [40]. 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.

3.3.2 Mixture model of t distributions and mixture of factor analysis

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 experiements. To solve this problem, [63] applied mixture of factor analysis to the clustering of experiments (see Figure 3.1). 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 hyperellipse (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, [63] added another stage to reduce the dimension of the gene space before applying the mixture of factor analysis to the clustering of the experiements. 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.

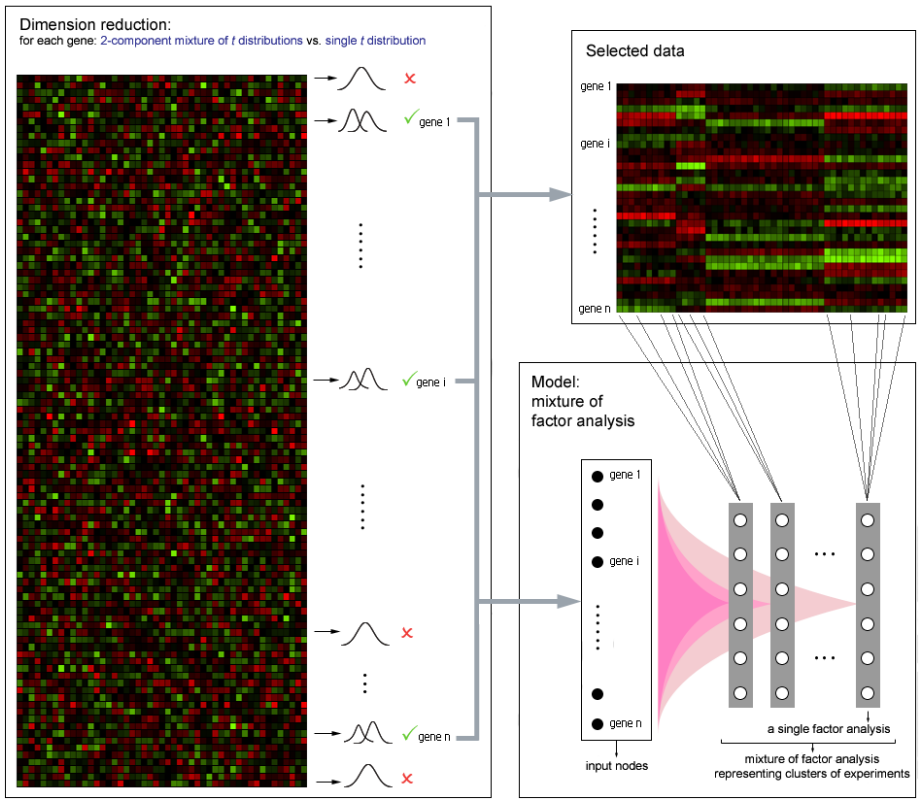


Figure 3.1: McLachlan *et al.* (2002) uses a two-component mixture model of t distributions to examine every gene expression profile against a single t distribution. Expression profiles to which the mixture models fit better (in terms of, for example, likelihood) are selected for further analysis. A mixture of factor analysis is applied on the selected data to cluster the experimental conditions.

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.

3.4 Biclustering algorithms

In addition to discovering the relationship between the genes or the relationships between the the experiments as in the 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 dimension of a microarray data set. As the 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 the works by Alon *et al.* [3] and Getz *et al.* [37], which exploit existing clustering algorithms for the task. In addition, there are also algorithms particularly designed for this purpose. In [57], mixtures of normal distributions, which is called the plaid model by the authors, are used to describe the microarray data and EM is applied for parameter estimation. For another example, the spectral biclustering method [54] applies singular value decomposition for solving the problem.

However, this type of biclustering algorithm has 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) [41]. 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 [41], 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 [49] that identifies gene clusters responsible for classification of experimental conditions.

The third type of biclusteirng algorithm questions 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. This type of biclustering algorithms is pioneered by Cheng and Church (2000) [19], where a heuristic approach is proposed to find patterns as large as possible that have minimum the mean squared residues while allowing variance to be present across the experiments when biclustering genes (or across the genes when biclustering experiments). This dissertation is dedicated to the tackling of the problem by using model based approaches, where the parameterization is done by Gibbs sampling. Another model-based approache is provided by Barash and Friedman (2002) [10], where EM algorithm is used for estimating the parameters of the model.

The work of Segal *et al.* [78] brings the third type of biclustering problem to a higher level. In their 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) [32] in particular, which is an extension of Bayesian networks – and uses (structral) EM [30] for the inference for the parameters of the probabilistic models. In [77], the capability of this framework is illustrated in unveiling the regulation programs of the genes from gene expression data.

Madeira and Oliveira (2004) [62] gives a comprehensive survey on the existing biclustering algorithms. In the following of the chapter, we discuss the gene shaving algorithm [41], the biclustering algorithm of Cheng and Church (2000) [19], and the PRMs for microarray data [78, 77] in more detail.

3.4.1 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 principle 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 carried on 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 measures the variability of each gene about the cluster average, and the between-variance of a cluster examines the variance of a cluster across the experiments. 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.* [41] 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.

3.4.2 Cheng and Church's approach

The biclustering algorithm of Cheng and Church (2000) [19] 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 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 ith row in the bicluster, a_{Ij} is the mean of jth 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 replace the matrix entries in the bicluster by random values. However, the masks are not used for the rows/colums addition procedure.

3.4.3 Probabilistic relational models for microarray data

In the preparation to address the problem in the language of PRMs [32], Segal *et al.* [78] 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 [43] 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 values (or called missing values) of some of the attributes. The learning procedure is carried out by the Structural EM algorithm [30]. 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 values to find the structure that maximizes a Bayesian information score [75]. The EM step uses the learned structure to estimate its CPDs (the M step) and the missing values (the E step).

The efficiency of the method is illustrated in [78] on two yeast data sets as well as synthetic data sets. In [77], it is shown that the method can be tailored for unveiling the regulatory program of the genes.

3.5 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 [87]) is also an appropriate biological val-

idation 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 [91], silhouette coefficients [50], 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 [104] 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 [14], 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. [14] 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 [52] 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. [14]. 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. [14] 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 [39, 87] 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

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

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.

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

Chapter 4

Gibbs sampling on Bayesian hierarchical models for the biclustering problem

Most of 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 about (when clustering patients, for example) what the classifications of the patients are. However, this type of analysis do not provide insights to those specific questions that biologists ask. One of the most intriguing questions for biologists is, for example, "what are the genes that are related to a particular function (or in a specific pathway) of interest to me?" A more ideal clustering algorithm in this regard should allow input (of the algorithm) to identify the target genetic function, and consequently direct the clustering result toward the discovery of genes related to the function.

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

Clustering algorithms based on Bayesian probabilistic models show promise as candidates meeting these requirements. First of all, Bayesian probabilistic models inherit general probabilistic models to reveal the fundamental structure that biologists seek in microarray data [78, 77] (see Section 3.4). Secondly, the hierarchical nature in Bayesian inferencing allow the introducing of prior

knowledge, which helps biologists to impose specific questions to the analysis of gene expression data. Finally, Bayeisan probabilistic models provide reliable and interpretable platforms for the integration of the analysis of gene expression data with other sources of biological information [79, 31].

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 sub set of genes have almost the same expression values. To distinguish these two types of biclustering problems, we refer to the former as "biclustering genes", and the latter as "biclustering experiments". We use Bayesian 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 [56] in DNA sequence analysis, and has become the method-of-choice for the problem [88]. Our idea to apply Bayesian strategy to the biclustering problem of microarray data was inspired by its success in the motif-finding problem.

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 expectation-maximization (EM) algorithm for solving incomplete-data problems. 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.

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 of 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, maximum a posteriori (MAP) estimates are used for the hidden variables and the model parameters.

*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 2.4). 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.

In the above, 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 computational intensive for structural learning. However, for the particular class of structure that addresses the biclusteirng 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 very efficient, and we favor the Gibbs sampling strategy over structural EM.

This chapter is organized as follows. First, in Section 4.1, we review some essential concepts for Gibbs sampling, especially on how and why it works. Then in Section 4.2, 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 4.3.

4.1 Gibbs sampling

Gibbs sampling is a technique to draw samples from a join distribution based on the full conditional distributions of all the associated random variables. Though the idea roots back to the work of Hasting (1970) [42], whose focus was on its Markov chain Monte Carlo (MCMC) nature, the Gibbs sampler was first formally introduced by Geman and Geman (1984) [36] 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) [34]). 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 [17]. We will talk about the difference between Bayesian statistics and classical statistics Section 4.2. In this section, we discuss the working mechanism of Gibbs sampling.

Gibbs sampling allow statisticians to avoid the tedious and sometimes non-trivial mathematical calculations of integrals for obtaining the join 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, ..., X_m, but that the marginal distributions (and thus their joint distribution) are (is) too complex

[†]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.

to directly sample from. Suppose also that the full conditional distributions $p(X_i \mid X_j;\ j \neq i)$ (for $i = 1, \ldots, n$), which can easily be sampled from, are available. 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) [36] 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 $i = 1, \ldots, m$).

4.1.1 The Markov chain property

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

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

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

$p(a \to b)$ 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 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 dimentionality.

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^*, i.e.,

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

if

$$p(j \to k) \cdot \pi^*_j = p(k \to j) \cdot \pi^*_k. \tag{4.9}$$

This condition is called detailed balance. When this condition is met, the stationary distribution π^* of the states (and thus, of the concerned variable) is guaranteed to be reached after a sufficient amount of time (i.e., when t is big enough), independent of the initial distribution of the states $\pi(0)$.

Casella and George (1992) [17] 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.

4.1.2 The Monte Carlo property

Only those samples collected by the Gibbs sampler after the convergence is reached can be used for joint (or marginal) distribution estimation. The Gibbs sampling procedure performed before the convergence is reached is often referred to as the "burn-in procedure", and the procedure during which samples are collected will be called the "sampling procedure" hereafter. The samples collected in the sampling procedure enable us to calculate the expectation of a function $f(X_i)$ over the distribution $p(X_i)$. This is done by the Monte Carlo integration

$$\mathrm{E}_{p(X_i)}[f(X_i)] = \int f(X_i) \cdot p(X_i)\mathrm{d}X_i \approx \frac{1}{T}\sum_{t=1}^{T} f(x_i^{(t)}), \tag{4.10}$$

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

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

However, as illustrated by Gelfand and Smith (1990) [34] (using the Rao-Blackwell theorem), a more accurate estimate of the expected value of X_i is provided by

$$\mathrm{E}_{p(X_i)}[X_i] = \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}_{p(X_i \mid X_j;\, j \neq i)}[X_i]. \tag{4.12}$$

Similarly, the posterior distribution itself can be approximated by

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

With more generality, a better alternative for Equation 4.10 is

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

The estimators obtained by Monte Carlo integration are unbiased MAP estimators.

4.1.3 Checking the convergence

A key issue in using Gibbs sampling is to determine when the procedure has essentially converged. The number of iterations needed for the burn-in procedure varies from 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 with in a few iterations. However, a bad starting point plus a multimodal target distribution with some of its probabilities close to zero can result in a poorly mixed chain so that only a small region of the random variable space is sampled for a long period of time. In this case, the number of burn-in iterations can easily reach a few thousand. In general, an optimal starting point close to the center of the marginal distribution can help in the accelerating the convergence. In addition, using multiple chains starting at independent positions of the random variable space can help to increase the coverage of the samples [35] 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) [71] aims to bound the summary statistics of the target variables within a certain precision. The 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 [38] 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) [35] 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) [23].

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 set of parameters obtained at each iteration and by

$$\bar{\boldsymbol{\omega}} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\omega}^{(t)} \tag{4.15}$$

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

$$\hat{\rho}_h = \frac{\mathrm{Cov}(\boldsymbol{\omega}^{(t)}, \boldsymbol{\omega}^{(t+h)})}{\mathrm{Var}(\boldsymbol{\omega}^{(t)})} = \frac{\sum_{t=1}^{T-h}(\boldsymbol{\omega}^{(t)} - \bar{\boldsymbol{\omega}})(\boldsymbol{\omega}^{(t+h)} - \bar{\boldsymbol{\omega}})}{\sum_{t=1}^{T-h}(\boldsymbol{\omega}^{(t)} - \bar{\boldsymbol{\omega}})^2}. \tag{4.16}$$

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

$$\iota = \sum_{h=1}^{\infty}\hat{\rho}_h \tag{4.17}$$

can be simplified to

$$\iota = (1 + \hat{\rho}_1)/(1 - \hat{\rho}_1). \tag{4.18}$$

A large autocorrelation time indicates that the chain is poorly mixed, 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 lth element in the chain will be used for the posterior summary statistics (see Equation 4.14). Another computational advantage of using the thinning procedure is that it saves the memory complexity of the Gibbs sampling procedure.

4.2 Bayesian hierarchical model for the biclustering problem

4.2.1 Bayesian hierarchical models

While the probability in classical probability theory and classical statistics theory measures the physical property of the chance that a random event would happen (e.g. the "head side" of a coin faces upwards after being tossed), the probability in the Bayesian probabilistic and statistics theory represents rather our *belief* that this certain random event would happen [43]. The purpose of the classical statistics is to estimate this physical probability. But Bayesian statistics concerns more about how our belief changes after we obtain a certain amount of information.

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), the problem for a Bayesian is "what do I learn from the collected information (i.e. data) about the probability that these random events could happen?". 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,

$$p(\mathbf{\Theta} \mid \mathcal{D}, \xi) = \frac{p(\mathbf{\Theta} \mid \xi) \cdot p(\mathcal{D} \mid \mathbf{\Theta}, \xi)}{p(\mathcal{D} \mid \xi)}. \tag{4.19}$$

In Equation 4.19, $\mathcal{D}$ 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 $\mathbf{\Theta}$ stands for the parameters of the distribution of $\mathbf{X}_m$ (i.e., model parameters, which quantify our belief). The first term in the numerator of the equation, $p(\mathbf{\Theta} \mid \xi)$, is called the prior distribution of $\mathbf{\Theta}$. It is parameterized by ξ, which represents our prior knowledge, i.e., our belief before observing the data (about how frequent the random events happens). The second term in the numerator, $p(\mathcal{D} \mid \mathbf{\Theta}, \xi)$, is called the likelihood of $\mathbf{\Theta}$, which is computed based on the probability distribution of $\mathbf{X}_m$. It 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 probability. The evidence can be obtained by

$$p(\mathcal{D}) = \int_{\mathbf{\Theta} \mid \xi} p(\mathcal{D} \mid \mathbf{\Theta}, \xi) \cdot p(\mathbf{\Theta} \mid \xi) \mathrm{d}\mathbf{\Theta}. \tag{4.20}$$

The left-hand part of the Equation 4.19, $p(\mathbf{\Theta} \mid \mathcal{D}, \xi)$, is called the posterior distribution of $\mathbf{\Theta}$. 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.

Interpreting Equation 4.19 from a topological point of view, we see that a simple Bayesian hierarchical model comprises of three levels. At the bottom level of the hierarchy are the concerned random variables $\mathbf{X}_m$ to which the data $\mathcal{D}$ is presented. Tthis 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. In a more complicated case, more levels of hierarchy can be built. The random variables at each level is modeled by a distribution characterized by the parameters at its direct upper level. But for our problem, the simple model of three layers suffices.

When both the data and the prior are given, the inference of the model parameters $\mathbf{\Theta}$ is straight forward by applying Equation 4.19. However, for our biclustering problem, the microarray data that we obtained only composes the "observed data" of the problem, which are addressed by the observed random variables in the model. But the partition of the observed data into the bicluster and the background needs the assistance of some hidden variable, whose value is 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.

4.2.2 Biclustering: an incomplete-data problem

Graphical models, especially directed acyclic graphical models provide a good representation tool to address the incomplete-data problem of biclustering in detail. The theory of graphical models combines probability theory and graph theory, and has 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) [48]. In the following, we briefly introduce the terminology used for graphical models. We then go direct into the discussion of applying graph models to depict the data level of Bayesian hierarchical model of the biclustering problem.

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 concerned random variables. 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 4.1(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 4.1(B), X

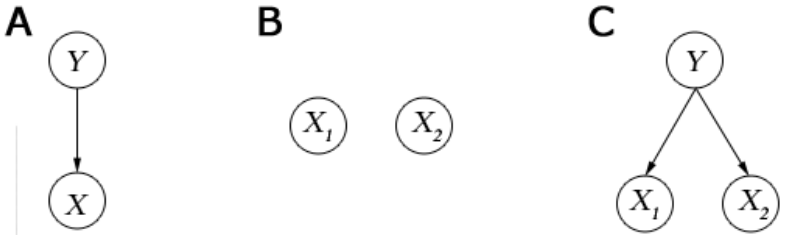


Figure 4.1: Examples of simple graphical model structures.

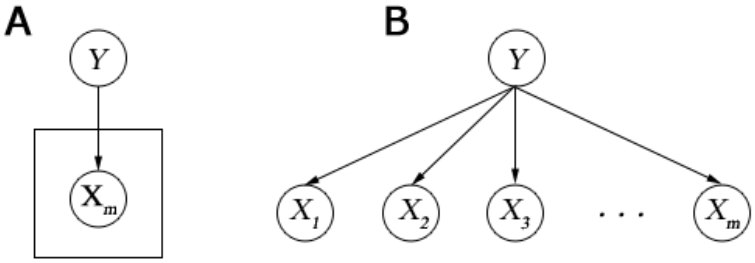


Figure 4.2: Example of plate: A uses a plate to simplify the structure presented in B. Both of the graphs implies that the m variables in $\mathbf{X}_m$ are conditionally independent and identically distributed given Y. This structure is known as a naive Bayes structure.

is independent of Y, mathematically denoted as $X \perp\!\!\!\perp Y$, which means

$$p(X, Y) = p(X) \cdot p(Y). \tag{4.21}$$

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 4.1(C), X_1 and X_2 are conditionally independent of each other given Y, written $X_1 \perp\!\!\!\perp X_2 \,|\, Y$, i.e.

$$p(X_1, X_2 \,|\, Y) = p(X_1 \,|\, Y) \cdot p(X_2 \,|\, Y). \tag{4.22}$$

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

When a graphical model has replicated structure, a plate can be used to simplify the graph, See Figure 4.2.

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 3.3, the task of clustering here

equals to associate each gene expression profile to the best fitting component in the mixture. Thus the data instances refer to the gene expression profiles.

From the graphical model point of view, the mixture model described in Section 3.3 is essentially a naive Bayes model as shown in Figure 4.2. Each data point (i.e., gene expression profile) 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 Y, whose value takes $1, \ldots, k$ indicating to which of the k components (of the mixture) a data point belongs. The values for $\mathbf{X}_m$ are observed (i.e., the microarray data, $\mathcal{D}$), however, the values for Y is not. Therefore, the data for Y is called missing data.

The nodes in the graph only represent the data level of the problem. The edges in Figure 4.2, i.e., the conditional distribution between $\mathbf{X}_m$ and Y, is where the hierarchical model discussed in Section 4.2.1 comes in,

$$p(\mathbf{X}_m \mid Y = y) = f(\boldsymbol{\Theta}_{\mathbf{X}_m \mid y}), \tag{4.23}$$

where $f(\boldsymbol{\Theta}_{\mathbf{X}_m|y})$ is a probability density function for $\mathbf{X}_m$, and $\boldsymbol{\Theta}_{\mathbf{X}_m|y}$ stands for the set of parameters for the density function when $Y = y$. For example, when a normal mixture model is used (see Section 3.3.1),

$$f(\boldsymbol{\Theta}_{\mathbf{X}_m|y}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{X}_m|y}, \boldsymbol{\Sigma}_{\mathbf{X}_m|y}). \tag{4.24}$$

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

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. 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 Figure 4.3 for an illustration). Note that we search for one bicluster at a time. (We will talk about how to find multiple biclusters later in this chapter.)

Therefore, 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 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 is depicted by Figure 4.4. The conditional distribution for $\mathbf{X}^{\text{bcl}}$ is

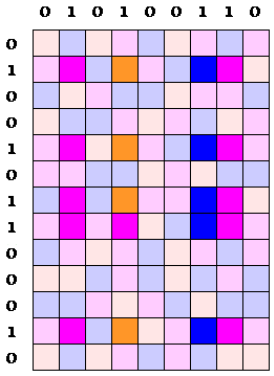


Figure 4.3: Conceptual plot of the biclustering problem: colors in the matrix represent different values. The data of the bicluster is highlighted. Binary labels are used to indicate if the row or the column belongs to the bicluster.

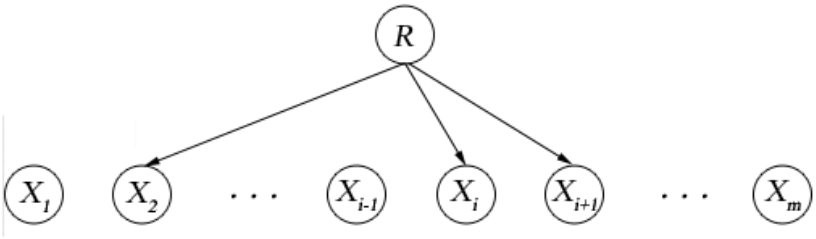


Figure 4.4: Graphical model for the biclustering problem with a fixed structure. Nodes for $\mathbf{X}_m$ that share edges with R (i.e., conditioned on R), e.g., X_2, X_i, and X_{i+1} in this example, correspond to the columns that belong to the bicluster, and are denoted as $\mathbf{X}^{\text{bcl}}$ as a set. The rest of the nodes in $\mathbf{X}_m$ correspond to the columns that belong to the background, and are denoted as $\mathbf{X}^{\text{bgd}}$.

$$p(\mathbf{X}^{\text{bcl}} \mid R = 1) = f(\mathbf{\Theta}^{\text{bcl}}), \tag{4.25}$$
$$p(\mathbf{X}^{\text{bcl}} \mid R = 0) = f(\mathbf{\Theta}^{\text{bgd}}). \tag{4.26}$$

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

$$p(\mathbf{X}^{\text{bgd}}) = f(\mathbf{\Theta}^{\text{bgd}}). \tag{4.27}$$

Again, $f(\cdot)$ denotes a probabilistic distribution in Equations 4.25 to 4.27. The parameters $\mathbf{\Theta}^{\text{bcl}}$ and $\mathbf{\Theta}^{\text{bgd}}$ are modeled by their prior distributions—$p(\mathbf{\Theta}^{\text{bcl}} \mid \boldsymbol{\xi}^{\text{bcl}})$ and $p(\mathbf{\Theta}^{\text{bgd}} \mid \boldsymbol{\xi}^{\text{bgd}})$ 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 the naive Bayes model for the clustering problem, or when the data structure (i.e., which edges are present in the graphical model) is known for the biclustering problem. 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 [24], (1) assuming the model parameters 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 the hidden variables estimated, it finds the model parameter that maximizes the complete likelihood. 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 produre is guaranteed to converge. However, the obtained solution of EM often gets stuck at the local maximum 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 4.1, the Gibbs sampling strategy estimates the joint distribution by sampling from the full conditional distributions of the involved random variables (i.e., nodes), and the MAP estimates of the random variables are obtained by performing Monte Carlo integrations. This strategy helps to enhance the chance to find 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.

The tast of Bayesian inference is therefore extended. Using $\mathcal{M}$ to denote the structure of the graphical model, Equation 4.19 can be rewritten as

$$p(\mathbf{\Theta} \mid \mathcal{D}, \mathcal{M}, \xi) = \frac{p(\mathbf{\Theta} \mid \mathcal{M}, \xi) \cdot p(\mathcal{D} \mid \mathbf{\Theta}, \mathcal{M}, \xi)}{p(\mathcal{D} \mid \mathcal{M}, \xi)}. \tag{4.28}$$

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 4.28,

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

In this case, both structural EM [30] (can be seen as an extension of the EM algorithm) and Gibbs sampling can be used for solving the problem.

Structural EM extended 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, MAP is used for the maximization step, i.e.,

$$\hat{\Theta} = \text{argmax}_{\Theta}\{E[\mathcal{D}^c \mid \Theta, \mathcal{M}, \xi]\}, \tag{4.30}$$

where $\mathcal{D}^c$ denotes the complete data. Then, a Bayesian information criterion (BIC) [75] 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. Note that C_m is thus equivalent to $\mathcal{M}$. The target posterior joint distribution becomes $p(\mathcal{D}_R, \mathbf{C}_m, \mathbf{\Theta} \mid \mathcal{D})$. Gibbs sampling is often thought to be a computational intensive procedure for the 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. We found Gibbs sampling to be an efficient strategy to find the global maximum solution for the problem.

In addition, as we have mentioned several times, Gibbs sampling paints out the entire posterior distribution of interest before a MAP estimate is made. This

property makes Gibbs sampling a suitable candidate for solving the model-based problems in bioinformatics, where the likelihood function or the posterior function usually consists of a large amount of modes due to 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.

4.3 Gibbs sampling for the biclustering problem

We have mentioned in Section 4.2 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 Bayesian context means that Bayesian hierarchical models are also used for R and $\mathbf{C}_m$. All of them are Bernoulli variables by nature (as explained in Section 4.2),

$$P(R) \sim \text{Bernoulli}(\Lambda^{\text{r}}), \tag{4.31}$$

$$P(C_j) \sim \text{Bernoulli}(\Lambda^{\text{c}}) \quad j = 1, \ldots, m. \tag{4.32}$$

The second equation above 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 follows Beta distributions.

$$p(\Lambda^{\text{r}}) \sim \text{Beta}(\zeta^{\text{r}}), \tag{4.33}$$

$$p(\Lambda^{\text{c}}) \sim \text{Beta}(\zeta^{\text{c}}). \tag{4.34}$$

The model level (which is composed of model parameters $\mathbf{\Theta}^{\text{bcl}}$, $\mathbf{\Theta}^{\text{bgd}}$, 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 target of Gibbs sampling. However, the prior level of the hierarchical model (which consists of hyperparameters ξ^{bcl}, ξ^{bgd}, and the ζ's) is treated as input of the algorithm, and is not updated during the Gibbs sampling procedure.

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

$$p(\mathcal{D}_R, \mathbf{C}_m, \mathbf{\Theta}^{\text{bcl}}, \mathbf{\Theta}^{\text{bgd}}, \Lambda^{\text{r}}, \Lambda^{\text{c}}, | \mathcal{D}, \xi^{\text{bcl}}, \xi^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}), \tag{4.35}$$

which means that the full conditional distribution of each of the concerned random variables is needed in order 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 varaibles 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.

4.3.1 The target posterior joint distribution

Let us start with some analysis of the target posterior joint distribution using the Bayes rule. First, separate out the evidence component, we have

$$\begin{aligned} & p(\mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \Lambda^{\mathrm{r}}, \Lambda^{\mathrm{c}}, | \mathcal{D}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}) \\ = \; & \frac{p(\mathcal{D}, \mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \Lambda^{\mathrm{r}}, \Lambda^{\mathrm{c}}, | \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}})}{p(\mathcal{D}, | \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}})}. \end{aligned} \tag{4.36}$$

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

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

Then, according to the Bayes rule (see Equation 4.19), the right-hand term in Equation 4.37 can be decomposed into components of likelihood and priors.

$$\begin{aligned} & p(\mathcal{D}, \mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \Lambda^{\mathrm{r}}, \Lambda^{\mathrm{c}}, | \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}) \\ = \; & p(\mathcal{D}, \mathcal{D}_R | \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \Lambda^{\mathrm{r}}) \cdot p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} | \mathbf{C}_m, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}) \\ & \cdot p(\Lambda^{\mathrm{r}} | \zeta^{\mathrm{r}}) \cdot P(\mathbf{C}_m | \Lambda^{\mathrm{c}}) \cdot p(\Lambda^{\mathrm{c}} | \zeta^{\mathrm{c}}) \end{aligned} \tag{4.38}$$

Adding the conditional independence of the observed data given the hidden data as depicted in Figure 4.4 to the above equation, we have

$$\begin{aligned} & p(\mathcal{D}, \mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \Lambda^{\mathrm{r}}, \Lambda^{\mathrm{c}}, | \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}) \\ = \; & p(\mathcal{D}, | \mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}) \cdot P(\mathcal{D}_R | \Lambda^{\mathrm{r}}) \cdot P(\mathbf{C}_m | \Lambda^{\mathrm{c}}) \\ & \cdot p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} | \mathbf{C}_m, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}) \cdot p(\Lambda^{\mathrm{r}} | \zeta^{\mathrm{r}}) \cdot p(\Lambda^{\mathrm{c}} | \zeta^{\mathrm{c}}) \\ = \; & \prod_{i=1}^{n} \left\{ p(\mathbf{X}_m = \mathcal{D}[i, \cdot] | R = \mathcal{D}_R[i], \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}) \cdot P(R = \mathcal{D}_R[i] | \Lambda^{\mathrm{r}}) \right\} \\ & \cdot p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} | \mathbf{C}_m, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}) \cdot p(\Lambda^{\mathrm{r}} | \zeta^{\mathrm{r}}) \cdot p(\Lambda^{\mathrm{c}} | \zeta^{\mathrm{c}}) \cdot P(\mathbf{C}_m | \Lambda^{\mathrm{c}}), \end{aligned} \tag{4.39}$$

where $\mathcal{D}[i, \cdot]$ denotes the ith 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 and $R[i]$, and that $R[i]$ is i.i.d. given Λ^{r}. From now on, we base the derivation of the full conditional distributions on Equation 4.39.

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

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. To simplify the Gibbs sampling procedure, we show in the following that Λ^{r} and Λ^{c} can be integrated out of the target distribution, and will not be sampled during the Gibbs sampling procedure.

$$\begin{aligned}
&p(\mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \,|\, \mathcal{D}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}) \\
&\quad\propto \int\int p(\mathcal{D}, \mathcal{D}_R, \mathbf{C}_m, \Lambda^{\mathrm{r}}, \Lambda^{\mathrm{c}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \,|\, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}})\, \mathrm{d}\Lambda^{\mathrm{r}}\, \mathrm{d}\Lambda^{\mathrm{c}} \\
&\quad= p(\mathcal{D} \,|\, \mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}) \cdot p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \,|\, \mathbf{C}_m, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}) \\
&\qquad\cdot \int P(\mathcal{D}_R \,|\, \Lambda^{\mathrm{r}}) \cdot p(\Lambda^{\mathrm{r}} \,|\, \zeta^{\mathrm{r}})\, \mathrm{d}\Lambda^{\mathrm{r}} \cdot \int P(\mathbf{C}_m \,|\, \Lambda^{\mathrm{c}}) \cdot p(\Lambda^{\mathrm{c}} \,|\, \zeta^{\mathrm{c}})\, \mathrm{d}\Lambda^{\mathrm{c}} \\
&\quad= p(\mathcal{D} \,|\, \mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}) \cdot P(\mathcal{D}_R \,|\, \zeta^{\mathrm{r}}) \\
&\qquad\cdot p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \,|\, \mathbf{C}_m, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}) \cdot P(\mathbf{C}_m \,|\, \zeta^{\mathrm{c}})
\end{aligned} \tag{4.40}$$

Consequently, the target joint distribution becomes

$$\begin{aligned}
&p(\mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \,|\, \mathcal{D}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}) \\
&\quad\propto\quad p(\mathcal{D}, \mathcal{D}_R, \mathbf{C}_m, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \,|\, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}}).
\end{aligned} \tag{4.41}$$

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

$$\begin{aligned}
P(\mathcal{D}_R \,|\, \zeta^{\mathrm{r}}) &= \int P(\mathcal{D}_R \,|\, \Lambda^{\mathrm{r}}) \cdot p(\Lambda^{\mathrm{r}} \,|\, \zeta^{\mathrm{r}})\, \mathrm{d}\Lambda^{\mathrm{r}} \\
&= \int p(\Lambda^{\mathrm{r}} \,|\, \zeta^{\mathrm{r}}) \cdot \prod_{i=1}^{n} P(R = \mathcal{D}_R[i] \,|\, \Lambda^{\mathrm{r}}) \mathrm{d}\Lambda^{\mathrm{r}} \\
&\propto \int (\Lambda^{\mathrm{r}})^{\zeta_1^{\mathrm{r}}-1} \cdot (1-\Lambda^{\mathrm{r}})^{\zeta_0^{\mathrm{r}}-1} \cdot (\Lambda^{\mathrm{r}})^{v} \cdot (1-\Lambda^{\mathrm{r}})^{(n-v)} \mathrm{d}\Lambda^{\mathrm{r}} \\
&\propto \frac{\Gamma(\zeta_0^{\mathrm{r}} + n - v)\Gamma(\zeta_1^{\mathrm{r}} + v)}{\Gamma(\zeta_0^{\mathrm{r}} + \zeta_1^{\mathrm{r}} + n)},
\end{aligned} \tag{4.42}$$

where v denotes the number of $\mathcal{D}_R[i] = 1$ for $i = 1, \ldots, n$. Similarly, for $\mathbf{C}_m$, we have

$$P(\mathbf{C}_m \,|\, \zeta^{\mathrm{c}}) \propto \int P(\mathbf{C}_m \,|\, \Lambda^{\mathrm{c}}) \cdot p(\Lambda^{\mathrm{c}} \,|\, \zeta^{\mathrm{c}})\, \mathrm{d}\Lambda^{\mathrm{c}} \propto \frac{\Gamma(\zeta_0^{\mathrm{c}} + m - w)\Gamma(\zeta_1^{\mathrm{c}} + w)}{\Gamma(\zeta_0^{\mathrm{c}} + \zeta_1^{\mathrm{c}} + m)}, \tag{4.43}$$

where w is the number of $C_j = 1$ for $j = 1, \ldots, m$. Note 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 became dependent of each other given ζ^{c}.

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

We now consider the full conditional distributions of R. For the ith row in the data, the full conditional probability $R = \mathcal{D}_R[i]$ is modeled by a Bernoulli distribution with parameter Λ_i^{r}, which is the probability that $\mathcal{D}_R[i] = 1$.

$$\begin{aligned}
\Lambda_i^{\text{r}} &= P(\mathcal{D}_R[i] = 1 \,|\, \mathcal{D}, \mathcal{D}_R[\bar{i}], \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}, \xi^{\text{bcl}}, \xi^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}) \\
&\propto p(\mathcal{D}, \mathcal{D}_R[i] = 1, \mathcal{D}_R[\bar{i}], | \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}, \zeta^{\text{r}}, \zeta^{\text{c}}) \\
&= P(\mathcal{D}_R[i] = 1, \mathcal{D}_R[\bar{i}] \,|\, \zeta^{\text{r}}) \cdot p(\mathbf{X}_m = \mathcal{D}[i, \cdot] \,|\, R = 1, \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}) \\
&\quad \cdot \prod_{k=1, k\neq i}^{n} p(\mathbf{X}_m = \mathcal{D}[k, \cdot] \,|\, R = \mathcal{D}_R[k], \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}) \\
&= P(\mathcal{D}_R[i] = 1, \mathcal{D}_R[\bar{i}] \,|\, \zeta^{\text{r}}) \cdot \prod_{\{j \,|\, C_j = 1\}} p(X_j = \mathcal{D}[i, j] \,|\, \boldsymbol{\Theta}^{\text{bcl}}) \\
&\quad \cdot \prod_{\{j \,|\, C_j = 0\}} p(X_j = \mathcal{D}[i, j] \,|\, \boldsymbol{\Theta}^{\text{bgd}}) \\
&\quad \cdot \prod_{k=1, k\neq i}^{n} p(\mathbf{X}_m = \mathcal{D}[k, \cdot] \,|\, R = \mathcal{D}_R[k], \mathbf{C}_m, \boldsymbol{\Theta}^{\text{bcl}}, \boldsymbol{\Theta}^{\text{bgd}}), \qquad (4.44)
\end{aligned}$$

where $\mathcal{D}_R[\bar{i}]$ denotes the data in all the other rows except the ith 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 ith 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

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

Putting the above two equations together to compute the odds between the two, $\gamma_i^{\mathrm{r}} = \frac{\Lambda_i^{\mathrm{r}}}{1-\Lambda_i^{\mathrm{r}}}$, we have

$$\begin{aligned}
\gamma_i^{\mathrm{r}} &= \frac{P(\mathcal{D}_R[i]=1\,|\,\mathcal{D},\,\mathcal{D}_R[\bar{i}],\,\mathbf{C}_m,\,\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}},\,\boldsymbol{\xi}^{\mathrm{bcl}},\,\boldsymbol{\xi}^{\mathrm{bgd}},\,\boldsymbol{\zeta}^{\mathrm{r}},\,\boldsymbol{\zeta}^{\mathrm{c}})}{P(\mathcal{D}_R[i]=0\,|\,\mathcal{D},\,\mathcal{D}_R[\bar{i}],\,\mathbf{C}_m,\,\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}},\,\boldsymbol{\xi}^{\mathrm{bcl}},\,\boldsymbol{\xi}^{\mathrm{bgd}},\,\boldsymbol{\zeta}^{\mathrm{r}},\,\boldsymbol{\zeta}^{\mathrm{c}})} \\
&= \frac{P(\mathcal{D}_R[i]=1,\,\mathcal{D}_R[\bar{i}]\,|\,\boldsymbol{\zeta}^{\mathrm{r}})}{P(\mathcal{D}_R[i]=0,\,\mathcal{D}_R[\bar{i}]\,|\,\boldsymbol{\zeta}^{\mathrm{r}})}\cdot\frac{p(\mathcal{D}\,|\,\mathcal{D}_R[i]=1,\,\mathbf{C}_m,\,\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}})}{p(\mathcal{D}\,|\,\mathcal{D}_R[i]=0,\,\mathbf{C}_m,\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}})} \\
&= \frac{\Gamma(n-v_{\bar{i}}+\zeta_0^{\mathrm{r}})\Gamma(v_{\bar{i}}+1+\zeta_1^{\mathrm{r}})}{\Gamma(n+\zeta_0^{\mathrm{r}}+\zeta_1^{\mathrm{r}})}\cdot\frac{\Gamma(n+\zeta_0^{\mathrm{r}}+\zeta_1^{\mathrm{r}})}{\Gamma(n-v_{\bar{i}}+1+\zeta_0^{\mathrm{r}})\Gamma(v_{\bar{i}}+\zeta_1^{\mathrm{r}})} \\
&\quad\cdot\frac{p(\mathcal{D}\,|\,\mathcal{D}_R[i]=1,\,\mathbf{C}_m,\,\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}})}{p(\mathcal{D}\,|\,\mathcal{D}_R[i]=0,\,\mathbf{C}_m,\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}})} \\
&= \frac{v_{\bar{i}}+\zeta_1^{\mathrm{r}}}{n-v_{\bar{i}}+\zeta_0^{\mathrm{r}}}\cdot\prod_{\{j\,|\,C_j=1\}}\frac{p(X_j=\mathcal{D}[i,j]\,|\,\boldsymbol{\Theta}^{\mathrm{bcl}})}{p(X_j=\mathcal{D}[i,j]\,|\,\boldsymbol{\Theta}^{\mathrm{bgd}})}. \qquad (4.46)
\end{aligned}$$

We use $v_{\bar{i}}$ to denote the number of rows (all but the ith row in the microarray data matrix) that currently belongs to the bicluster. Therefore, with the model structure and parameters fixed, and knowing what other rows in the microarray data matrix belong to the bicluster, the full conditional odds of whether the row under consideration belongs to the bicluster equals to a weighted likelihood ratio. The likelihood is calculated between the case that the data of the row under the biclustering columns are generated by the bicluster and the case that these data are generated by the background. The weight is calculated as the ratio between the number of rows that are currently assigned and the bicluster to the number of rows that currently belongs to the background.

The full conditional distribution of C_j (for $j = 1,\ldots,m$) involves more parameters. Because of the dependence of $\boldsymbol{\Theta}$ on $\mathbf{C}_m$, the untanglement of the distribution requires a 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,

$$\begin{aligned}
\Lambda_j^{\mathrm{c}} &= p(C_j=1\,|\,\mathcal{D},\,\mathcal{D}_R,\,\mathbf{C}_{\bar{j}},\,\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}},\,\boldsymbol{\xi}^{\mathrm{bcl}},\,\boldsymbol{\xi}^{\mathrm{bgd}},\,\boldsymbol{\zeta}^{\mathrm{r}},\,\boldsymbol{\zeta}^{\mathrm{c}}) \\
&\propto p(\mathcal{D},\,\mathcal{D}_R,\,C_j=1,\,\mathbf{C}_{\bar{j}},\,\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}},\,|\,\boldsymbol{\xi}^{\mathrm{bcl}},\,\boldsymbol{\xi}^{\mathrm{bgd}},\,\boldsymbol{\zeta}^{\mathrm{r}},\,\boldsymbol{\zeta}^{\mathrm{c}}) \\
&= p(\mathcal{D},\,|\,\mathcal{D}_R,\,C_j=1,\,\mathbf{C}_{\bar{j}},\,\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}})\cdot P(\mathcal{D}_R\,|\,\boldsymbol{\zeta}^{\mathrm{r}}) \\
&\quad\cdot p(\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}}\,|\,C_j=1,\,\mathbf{C}_{\bar{j}},\,\boldsymbol{\xi}^{\mathrm{bcl}},\,\boldsymbol{\xi}^{\mathrm{bgd}})\cdot P(C_j=1,\mathbf{C}_{\bar{j}}\,|\,\boldsymbol{\zeta}^{\mathrm{c}}). \qquad (4.47)
\end{aligned}$$

The complement of the parameter is,

$$\begin{aligned}
1-\Lambda_j^{\mathrm{c}} &= p(C_j=1\,|\,\mathcal{D},\,\mathcal{D}_R,\,\mathbf{C}_{\bar{j}},\,\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}},\,\boldsymbol{\xi}^{\mathrm{bcl}},\,\boldsymbol{\xi}^{\mathrm{bgd}},\,\boldsymbol{\zeta}^{\mathrm{r}},\,\boldsymbol{\zeta}^{\mathrm{c}}) \\
&\propto p(\mathcal{D},\,|\,\mathcal{D}_R,\,C_j=0,\,\mathbf{C}_{\bar{j}},\,\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}})\cdot P(\mathcal{D}_R\,|\,\boldsymbol{\zeta}^{\mathrm{r}}) \\
&\quad\cdot p(\boldsymbol{\Theta}^{\mathrm{bcl}},\,\boldsymbol{\Theta}^{\mathrm{bgd}}\,|\,C_j=0,\,\mathbf{C}_{\bar{j}},\,\boldsymbol{\xi}^{\mathrm{bcl}},\,\boldsymbol{\xi}^{\mathrm{bgd}})\cdot P(C_j=0,\,\mathbf{C}_{\bar{j}}\,|\,\boldsymbol{\zeta}^{\mathrm{c}}). \qquad (4.48)
\end{aligned}$$

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

$$\begin{aligned}
\gamma_j^{\mathrm{c}} &= \frac{p(C_j = 1 \mid \mathcal{D}, \mathcal{D}_R, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}})}{p(C_j = 1 \mid \mathcal{D}, \mathcal{D}_R, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}}, \zeta^{\mathrm{r}}, \zeta^{\mathrm{c}})} \\
&= \frac{p(\mathcal{D}, \mid \mathcal{D}_R, C_j = 1, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}})}{p(\mathcal{D}, \mid \mathcal{D}_R, C_j = 0, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}})} \\
&\quad \cdot \frac{p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \mid C_j = 1, \mathbf{C}_{\bar{j}}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}})}{p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \mid C_j = 0, \mathbf{C}_{\bar{j}}, \xi^{\mathrm{bcl}}, \xi^{\mathrm{bgd}})} \cdot \frac{w_{\bar{j}} + \zeta_1^{\mathrm{c}}}{m - w_{\bar{j}} + \zeta_0^{\mathrm{c}}}.
\end{aligned} \tag{4.49}$$

In Equations 4.47 to 4.49, 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 belongs to the bicluster. Equation 4.49 tells us that γ_j^{c} is a product of the likelihood ratio of the data (between the case when the jth column of the microarray data is generated by the bicluster and the case when it is generated by the background), a ratio between the priors of $\boldsymbol{\Theta}$, and a ratio between the number of columns that are already in the bicluster and the number of those that are in the background.

4.3.4 The Gibbs sampling scheme for the biclustrering problem

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

1. Initialization: Assign binary random values to $\mathbf{C}_m$ (i.e., to initialize the model structure) and $\mathcal{D}_R$ (i.e., to initialize the hidden data), and initialize 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 4.46.
   (b) Draw a label for $\mathcal{D}_R[i]$ from the Bernoulli distribution.
3. Update the model structgure: 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 4.49.
   (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

4.3.5 From samples to the final pattern

To reduce the memory complexity of the algorithm, we only collect the samples of the hidden data $\mathcal{D}_R$ and of the model structure $\mathbf{C}_m$ to determine the final position of the bicluster. The parameters for the model and the background—if necessary—will be obtained by evaluating the sample statistics of these parameters according to the determined position of the bicluster. According to Equation 4.12, the MAP estimate of $\mathcal{D}_R[i]$ is.

$$\mathrm{E}_{P(\mathcal{D}_R[i]\,|\,\mathcal{D})}\Big[\mathcal{D}_R[i]\Big] = \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}_{P(\mathcal{D}_R[i]\,|\,\mathcal{D}_R[\bar{i}]^{(t)},\,\mathbf{C}_m^{(t)},\,(\boldsymbol{\Theta}^{\mathrm{bcl}})^{(t)},\,(\boldsymbol{\Theta}^{\mathrm{bgd}})^{(t)},\,\mathcal{D})}\Big[\mathcal{D}_R[i]\Big] \quad i = 1,\ldots,n. \tag{4.50}$$

Similarly, the MAP estimate of C_j is,

$$\mathrm{E}_{P(C_j\,|\,\mathcal{D})}[C_j] = \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}_{P(C_j\,|\,\mathcal{D}_R^{(t)},\,\mathbf{C}_{\bar{j}}^{(t)},\,(\boldsymbol{\Theta}^{\mathrm{bcl}})^{(t)},\,(\boldsymbol{\Theta}^{\mathrm{bgd}})^{(t)},\,\mathcal{D})}[C_j] \quad j = 1,\ldots,m. \tag{4.51}$$

The marginal distribution of $\mathcal{D}_R[i]$ and C_j, according to Equation 4.13, are

$$\mathrm{E}[P(\mathcal{D}_R[i]\,|\,\mathcal{D})] = \frac{1}{T}\sum_{t=1}^{T} P(\mathcal{D}_R[i]\,|\,\mathcal{D}_R[\bar{i}]^{(t)},\,\mathbf{C}_m^{(t)},\,(\boldsymbol{\Theta}^{\mathrm{bcl}})^{(t)},\,(\boldsymbol{\Theta}^{\mathrm{bgd}})^{(t)},\,\mathcal{D}) \quad i = 1,\ldots,n, \tag{4.52}$$

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

Note that to simplify the notation, we omitted the priors $\boldsymbol{\xi}^{\mathrm{bcl}}$, $\boldsymbol{\xi}^{\mathrm{bgd}}$, $\boldsymbol{\zeta}^{\mathrm{r}}$, and $\boldsymbol{\zeta}^{\mathrm{c}}$ out of the notation of the probability distributions. In Equations 4.50 to 4.53, t indexes the iterations and T is the total number of iterations in the sampling procedure. To determine the final position of the bicluster, we can either put a threshold on the MAP estimate of the random variables, and select only the rows and columns for whom the MAP estimate are above that threshold for the bicluster. Or, another more meaningful way is to put a threshold on the, say 95% quantile, of the estimated marginal distribution of the random variables. In the examples that we give in Chapter 5 and Chapter 6, we use the latter method as the criterion for the determination of the position of the bicluster.

In our implementation, we use 500 iteration as default for performing the Gibbs sampling procedure (including both the burn-in and the sampling procedure). The autocorrelation (see Equation 4.16) between the samples is monitored for determining the convergence, and additional iterations are added if necessary.

4.3.6 Multiple biclusters

The probabilistic model that we discussed above considers only the presence of a single bicluster in the data set, which is not biologically realistic. Several methods can be used to enable the detection of multiple biclusters. We choose (for both the biclustering of genes and the biclustering of experiments) to mask the experiments selected for the found biclusters and rerun the algorithm on the rest of the data. By masking, we mean that the random variable (subset of $\mathbf{C}_m$ in the case of biclustering genes, and subset of $\mathcal{D}_R$ in the case of biclustering experiements) 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 does allow 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 4.3.6 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 contain 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 4.3.4). If either of the numbers equals zero, we reinitialize the algorithm and perform Gibbs sampling again. However, if after a predefined number of reinitializations (for example, 50 in our implementation) the algorithm still does not succeed to reach convergence, we terminate the algorithm and consider that the data set does not contain a bicluster.

Chapter 5

Biclustering experiments in microarray data

In this chapter we discuss applying Gibbs sampling to the biclustering problem is to bicluster experiments (e.g. tumor samples) on discretized microarray data [80]. The aim is to find experiments whose discrete expression levels are consistent for each gene selected for the bicluster. To explain in accordance with the frame that we give in Chapter 4, we transpose the discretized microarray data matrix 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 is because of not only the fact that the Gibbs biclustering algorithm was inspired by successfulness of applying Gibbs sampling to the motif-finding problem in DNA sequence analysis [56, 60, 89] where the data has a discrete nature so that the mathematical models of the motif-finding problem can be conveniently borrowed to the biclustering problem, but also the consideration 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 [63]. 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 we discretize the microarray data for the biclustering of experiments in more detail.

- *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 pathological discoveries under two circumstances. In the first case, we assume that the biologist (or the doctor) have no idea about the pathological types of the tumors from whom samples are collected for the microarray experiments. The task of biclustering is to discover global patterns that are embedded in the data, which provide expressional fingerprints for different pathological types. However, the expressional patterns of some pathological traits overwhelms in their size and amplitude than the others, and consequently dominate the results of biclustering. To enhance the ability of the biclusering 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 belongs to the pathological type 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 pathological discovery.

5.1 The discretization of microarray data

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 "up-regulated"), "median" (or "not activated"), and "low" (or "down-regulated"). Mathematically, the equal-frequency discretization is a maximum-entropy method, and thus does not (in principle) introduce additional information during the discretization procedure. The procedure of equal-frequency discretization is carried out as follows. For each gene profile, we assign the experiments with the lowest $\frac{1}{3}$ of the expression values to the first bin (corresponding to "low"), similarly, the $\frac{1}{3}$ 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").

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-frequncy principle is reserved.

Note that the equal-frequency discretization takes care of the standardization of the gene expression profiles automatically. 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 decouple the influence of equal-frequency discretization by permutating the order of the experiments in the data set before performing the discretization.

5.2 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 $0 \leq \psi_i \leq 1$, $\sum_{i=1}^{3} \psi = 1$ for $i = \{1, 2, 3\}$. 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 $0 \leq \phi_{i,j} \leq 1$, $\sum_{i=1}^{3} \phi_{i,j} = 1$ for $i = \{1, 2, 3\}$; 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 $\mathbf{X}_m$ to whom the discrete expression data map to. The explicit distribution of $\mathbf{X}_m$ is now,

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

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

As we discussed in Chapter 4, the prior level is one of the indispensable building part of a Bayesian hierarchical model. The priors in our model plays the main role in directing the discovery of biclusters, as will be explained in Section 5.4 and will be further illustrated in Section 5.5 and Section 5.6. 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.

For the flexibility of the model, 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,

$$\Phi_j \sim \text{Dirichlet}(\boldsymbol{\beta}_j), \tag{5.8}$$

$$\boldsymbol{\beta}_j = [\ \beta_{1,j} \quad \beta_{2,j} \quad \beta_{3,j}\ ]^{\mathrm{T}}, \tag{5.9}$$

$$\mathrm{B} = \{\boldsymbol{\beta}_j \,|\, C_j = 1\}. \tag{5.10}$$

Dirichlet distribution is also used for the prior of Ψ,

$$\Psi \sim \text{Dirichlet}(\boldsymbol{\alpha}), \tag{5.11}$$

$$\boldsymbol{\alpha} = [\ \alpha_1 \quad \alpha_2 \quad \alpha_3\ ]^{\mathrm{T}} \tag{5.12}$$

5.3 Full conditional distributions

Substituting our model into Equation 4.41, the target joint distribution of the Gibbs sampling procedure for the biclustering problem of experiments is

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

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 4.3.3. 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 Ψ.

Using conjugate priors for Ψ and $\boldsymbol{\Phi}$ means that the full conditional distributions of the parameters are also in the forms of Dirichlet distributions. Sampling from Dirichlet distributions is not a trivial procedure and consumes a non-negligible amount of computation. 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 4.3.2). Consequently, the target joint distribution becomes $P(\mathcal{D}_R, \mathbf{C}_m \,|\, \mathcal{D}, \boldsymbol{\alpha}, \mathrm{B}, \boldsymbol{\zeta}^{\mathrm{r}}, \boldsymbol{\zeta}^{\mathrm{c}})$.

To facilitate the analyze the full conditional distributions of $\mathcal{D}_R$ and $\mathbf{C}_m$, we need the help of the following notations. 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 rows (i.e., 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 rows that areassigned to the background. Similarly,

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

is the vector of the indices of the columns (i.e., 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 ith row (or the jth column) of the matrix. E.g., $\bar{\mathbf{r}}_{\bar{i}}$ refers to the rows that are assigned to the background excluding the ith 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 ith row and the jth 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 two scalers u and v, we use $\delta_u(v)$ to denote a index vector of length u whose v^{th} entry equals 1, and the rest of its entries equal 0. Given a vector of indices $\mathbf{u}$ and a scaler v, $\mathbf{u} \oplus v$ denotes including v in vector $\mathbf{u}$;

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 $\boldsymbol{u}^{\boldsymbol{v}} = u_1^{v_1} \cdot u_2^{v_2} \cdots u_k^{v_k}$, $\Gamma(\boldsymbol{u}) = \Gamma(u_1) \cdot \Gamma(u_2) \cdots \Gamma(u_k)$, and $\sum \boldsymbol{u} = \sum_{i=1}^{k} u_i$.

Coming back to the full conditional distribution of $\mathcal{D}_R[i]$, the final result in Equation 4.46 is not applicable anymore because of the integration on the model parameters. In stead, we have to rely on the fact that

$$\begin{aligned}
\gamma_i^{\mathrm{r}} =& \frac{P(\mathcal{D},\, \mathcal{D}_R[i] = 1,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m \,|\, \boldsymbol{\alpha},\, \mathrm{B},\, \boldsymbol{\zeta}^{\mathrm{r}},\, \boldsymbol{\zeta}^{\mathrm{c}})}{P(\mathcal{D},\, \mathcal{D}_R[i] = 0,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m \,|\, \boldsymbol{\alpha},\, \mathrm{B},\, \boldsymbol{\zeta}^{\mathrm{r}},\, \boldsymbol{\zeta}^{\mathrm{c}})} \\
=& \frac{P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 1,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})}{P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 0,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})} \cdot \frac{P(\mathcal{D}_R[i] = 1,\, \mathcal{D}_R[\bar{i}] \,|\, \boldsymbol{\zeta}^{\mathrm{r}})}{P(\mathcal{D}_R[i] = 0,\, \mathcal{D}_R[\bar{i}] \,|\, \boldsymbol{\zeta}^{\mathrm{r}})} \\
=& \frac{P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 1,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})}{P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 0,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})} \cdot \frac{v_{\bar{i}} + \zeta_1^{\mathrm{r}}}{n - v_{\bar{i}} + \zeta_0^{\mathrm{r}}}.
\end{aligned} \tag{5.18}$$

Performing the integration respecitvely for $P(\mathcal{D} \,|\, \mathcal{D}_R[i] = 1,\, \mathcal{D}_R[\bar{i}],\, \mathbf{C}_m,\, \boldsymbol{\alpha},\, \mathrm{B})$

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

$$
\begin{aligned}
&P(\mathcal{D} \mid \mathcal{D}_R[i] = 1,\ \mathcal{D}_R[\bar{i}],\ \mathbf{C}_m,\ \boldsymbol{\alpha},\ \mathrm{B}) \\
&= \int_{\boldsymbol{\Phi}} \int_{\Psi} P(\mathcal{D} \mid \mathcal{D}_R[i] = 1,\ \mathcal{D}_R[\bar{i}],\ \mathbf{C}_m,\ \boldsymbol{\Phi},\ \Psi) \cdot p(\boldsymbol{\Phi} \mid \mathrm{B},\ \mathbf{C}_m) \cdot p(\Psi \mid \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
&\propto \int_{\boldsymbol{\Phi}} \int_{\Psi} \Psi^{h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}},\, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \bar{\mathbf{c}}]\}} \cdot \prod_{\{j \mid C_j = 1\}} \Phi_j^{h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \delta_m(j)]\}} \cdot \Psi^{\boldsymbol{\alpha}-1} \cdot \prod_{\{j \mid C_j = 1\}} \Phi_j^{\boldsymbol{\beta}_j - 1}\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
&= \frac{\Gamma\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}},\, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}},\, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)\Big\}} \cdot \prod_{\{j \mid C_j = 1\}} \frac{\Gamma\Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)\Big\}}, \\
&\qquad i = 1 \ldots n,
\end{aligned}
\tag{5.19}
$$

and

$$
\begin{aligned}
&P(\mathcal{D} \mid \mathcal{D}_R[i] = 0,\ \mathcal{D}_R[\bar{i}],\ \mathbf{C}_m, \boldsymbol{\alpha}, \mathrm{B}) \\
&= \int_{\boldsymbol{\Phi}} \int_{\Psi} P(\mathcal{D} \mid \mathcal{D}_R[i] = 0,\ \mathcal{D}_R[\bar{i}],\ \mathbf{C}_m,\ \boldsymbol{\Phi},\ \Psi) \cdot p(\boldsymbol{\Phi} \mid \mathrm{B},\ \mathbf{C}_m) \cdot (\Psi \mid \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
&\propto \int_{\boldsymbol{\Phi}} \int_{\Psi} \Psi^{h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}} \oplus i,\, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \bar{\mathbf{c}}]\}} \cdot \prod_{\{j \mid C_j = 1\}} \Phi_j^{h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \delta_m(j)]\}} \cdot \Psi^{\boldsymbol{\alpha}-1} \cdot \prod_{\{j \mid C_j = 1\}} \Phi_j^{\boldsymbol{\beta}_j - 1}\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
&= \frac{\Gamma\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}} \oplus i,\, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}} \oplus i,\, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)\Big\}} \cdot \prod_{\{j \mid C_j = 1\}} \frac{\Gamma\Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)\Big\}} \\
&\qquad i = 1 \ldots n.
\end{aligned}
\tag{5.20}
$$

The nature of the Γ function reveals that the devision of the latter term in Equation 5.19 by the latter term in Equation 5.20 have simpler forms,

$$
\begin{aligned}
\frac{\Gamma\Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)}{\Gamma\Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \delta_m(j)]\} + \boldsymbol{\beta}_{\cdot,j}\Big)} &= \Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)^{\delta_3\{\mathcal{D}[i,j]\}} \\
\{i \mid i = 1 \ldots n\}, \{j \mid C_j = 1\}&,
\end{aligned}
\tag{5.21}
$$

and

$$
\begin{aligned}
\frac{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i,\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)\Big\}}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r}_{\bar{i}},\, \delta_m(j)]\} + \boldsymbol{\beta}_j\Big)\Big\}} &= \frac{\Gamma(v_{\bar{i}} + 1 + \sum \boldsymbol{\beta}_j)}{\Gamma(v_{\bar{i}} + \sum \boldsymbol{\beta}_j)} = v_{\bar{i}} + \sum \boldsymbol{\beta}_j, \\
i = 1 \ldots n,\ j = 1 \ldots w&.
\end{aligned}
\tag{5.22}
$$

To calculate the devision of the first terms in Equation 5.19 and 5.20, we need the approximation [60] that when the size of U_2 is relatively small compared to U_1 and the composition of U_2 is relatively diverse,

$$
\frac{\Gamma\{h(U_1 \oplus U_2)\}}{\Gamma\{h(U_1)\}} \approx h(U_1)^{h(U_2)}. \tag{5.23}
$$

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 bigger 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,

$$\begin{aligned}
&\frac{\Gamma\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}} \oplus i, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}}, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)} \\
&= \frac{\Gamma\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}}, \bar{\mathbf{c}}]\} + h\{\mathcal{D}[\delta_n(i), \bar{\mathbf{c}}]\} + \boldsymbol{\alpha} + h\{\mathcal{D}[\delta_n(i), \mathbf{c}]\}\Big)}{\Gamma\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}}, \bar{\mathbf{c}}]\} + h\{\mathcal{D}[\delta_n(i), \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)} \\
&\approx \Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)^{h\{\mathcal{D}[\delta_n(i), \mathbf{c}]\}}, \quad i = 1 \ldots n. \qquad (5.24)
\end{aligned}$$

Similarly,

$$\begin{aligned}
&\frac{\Gamma\Big\{\sum\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}} \oplus i, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}}, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)\Big\}}{\Gamma\Big\{\sum\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)\Big\}} \\
&= \sum\Big(h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}} \oplus i, \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}\Big)^{\sum(h\{\mathcal{D}[\delta_n(i), \mathbf{c}]\})}, \quad i = 1 \ldots n. \qquad (5.25)
\end{aligned}$$

Equipped with Equations 5.21 to 5.25, we put Equation 5.19, Equation 5.20 together to evaluate Equation 5.18. We finally arrive at

$$\gamma_i^{\mathrm{r}} = \prod_{\{j \,|\, C_j = 1\}} \left(\frac{\hat{\Phi}_j}{\hat{\Psi}}\right)^{\delta_3(\mathcal{D}[i,j])} \cdot \frac{v_{\bar{i}} + \zeta_1^{\mathrm{r}}}{n - 1 - v_{\bar{i}} + \zeta_0^{\mathrm{r}}}, \quad i = 1 \ldots n, \qquad (5.26)$$

where

$$\hat{\Phi}_j = \frac{h\{\mathcal{D}[r_{\bar{i}}, \delta_m(j)]\} + \boldsymbol{\beta}_j}{v_{\bar{i}} + \sum \boldsymbol{\beta}_j} \qquad \{j \,|\, C_j = 1\} \qquad (5.27)$$

$$\hat{\Psi} = \frac{h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{i}}, \cdot]\} + h\{\mathcal{D}[\mathbf{r}_{\bar{i}}, \bar{\mathbf{c}}]\} + h\{\mathcal{D}[\delta_n(i), \bar{\mathbf{c}}]\} + \boldsymbol{\alpha}}{n \cdot m - v_{\bar{i}} \cdot w}. \qquad (5.28)$$

The final result of γ_i^{r} in Equation 5.26 is exactly in the same form as Equation 4.46. $\hat{\Phi}$ and $\hat{\Psi}$ are the byproduct of our method, which represent respectively the model of the bicluster and the model of the background. Equation 5.27 and 5.28 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 divided data according to $\mathcal{D}_R[\bar{i}]$ and $\mathbf{C}_m$. Intuitively, by fixing all the other labels 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 5.18, for the model structural variables $C_j \in \mathbf{C}_m$, we also have

$$\gamma_j^{\mathrm{c}} = \frac{w_{\bar{j}} + \zeta_1^{\mathrm{c}}}{m - w_{\bar{j}} + \zeta_0^{\mathrm{c}}} \cdot \frac{P(\mathcal{D}, \,|\, \mathcal{D}_R,\, C_j = 1,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\alpha}, \mathrm{B})}{P(\mathcal{D}, \,|\, \mathcal{D}_R,\, C_j = 0,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\alpha}, \mathrm{B})}. \tag{5.29}$$

Comparing the graph 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 $\phi_j|\boldsymbol{\alpha}$. Evaluating $P(\mathcal{D}, \,|\, \mathcal{D}_R,\, C_j = 1,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\alpha}, \mathrm{B})$ and $P(\mathcal{D}, \,|\, \mathcal{D}_R,\, C_j = 0,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\alpha}, \mathrm{B})$ respecitvely, we have

$$\begin{aligned}
&P(\mathcal{D} \,|\, \mathcal{D}_R,\, C_j = 1,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\alpha}, \mathrm{B}) \\
&= \int_{\boldsymbol{\Phi}} \int_{\Psi} P(\mathcal{D} \,|\, \mathcal{D}_R,\, C_j = 1,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\Phi},\, \Psi) \cdot p(\boldsymbol{\Phi} \,|\, C_j = 1, \mathbf{C}_{\bar{j}},\, \mathrm{B}) \cdot p(\Psi \,|\, \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
&\propto \int_{\boldsymbol{\Phi}} \int_{\Psi} \Psi^{h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}}]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}} \oplus j]\}} \cdot \prod_{\{k \,|\, C_k = 1,\, k \neq j\}} \Phi_k^{h\{\mathcal{D}[\mathbf{r},\, \delta_m(k)]\}} \cdot \Phi_j^{h\{\mathcal{D}[\mathbf{r},\, \delta_m(j)]\}} \\
&\quad \cdot \Psi^{\boldsymbol{\alpha}-1} \cdot \prod_{\{k \,|\, C_k = 1,\, k \neq j\}} \Phi_k^{\boldsymbol{\beta}_k - 1} \cdot \Phi_j^{\boldsymbol{\alpha}-1}\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
&= \frac{\Gamma\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}}]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}} \oplus j]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}}]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}} \oplus j]\} + \boldsymbol{\alpha}\Big)\Big\}} \cdot \prod_{\{k \,|\, C_k = 1,\, k \neq j\}} \frac{\Gamma\Big(h\{\mathcal{D}[\mathbf{r},\, \mathbf{c}_k]\} + \boldsymbol{\beta}_k\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r},\, \mathbf{c}_k]\} + \boldsymbol{\beta}_k\Big)\Big\}} \\
&\quad \cdot \frac{\Gamma\Big(h\{\mathcal{D}[\mathbf{r},\, \delta_m(j)]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r},\, \delta_m(j)]\} + \boldsymbol{\alpha}\Big)\Big\}}, \quad j = 1 \ldots m,
\end{aligned} \tag{5.30}$$

and

$$\begin{aligned}
&P(\mathcal{D} \,|\, \mathcal{D}_R,\, C_j = 0,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\alpha},\, \mathrm{B}) \\
&= \int_{\boldsymbol{\Phi}} \int_{\Psi} P(\mathcal{D} \,|\, \mathcal{D}_R,\, C_j = 0,\, \mathbf{C}_{\bar{j}},\, \boldsymbol{\Phi},\, \Psi) \cdot p(\boldsymbol{\Phi} \,|\, C_j = 0,\, \mathbf{C}_{\bar{j}},\, \mathrm{B}) \cdot p(\Psi \,|\, \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
&\propto \int_{\boldsymbol{\Phi}} \int_{\Psi} \Psi^{h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}} \oplus j]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}}]\}} \cdot \prod_{\{k \,|\, C_k = 1,\, k \neq j\}} \Phi_k^{h\{\mathcal{D}[\mathbf{r},\, \mathbf{c}_k]\}} \cdot \Psi^{\boldsymbol{\alpha}-1} \cdot \prod_{\{k \,|\, C_k = 1, k \neq j\}} \Phi_k^{\boldsymbol{\beta}_k}\, \mathrm{d}\boldsymbol{\Phi}\, \mathrm{d}\Psi \\
&= \frac{\Gamma\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}} \oplus j]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}}]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}} \oplus j]\} + h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{j}},\, \mathbf{c}]\} + \boldsymbol{\alpha}\Big)\Big\}} \cdot \prod_{\{k \,|\, C_k = 1, k \neq j\}} \frac{\Gamma\Big(h\{\mathcal{D}[\mathbf{r},\, \mathbf{c}_k]\} + \boldsymbol{\beta}_k\Big)}{\Gamma\Big\{\sum \Big(h\{\mathcal{D}[\mathbf{r},\, \mathbf{c}_k]\} + \boldsymbol{\beta}_k\Big)\Big\}}, \\
&\quad j = 1 \ldots m.
\end{aligned} \tag{5.31}$$

Putting the above two equations together to calculate Equation 5.29, we get

$$
\begin{aligned}
\gamma_j^{\mathrm{c}} &= \frac{\Gamma\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}}]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}} \oplus j]\} + \boldsymbol{\alpha}\Big) \cdot \Gamma\Big(h\{\mathcal{D}[\mathbf{r},\, j]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}} \oplus j]\} + h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{j}},\, \mathbf{c}]\} + \boldsymbol{\alpha}\Big)} \\
&\quad \cdot \frac{\Gamma\Big\{\sum\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}} \oplus j]\} + h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{j}},\, \mathbf{c}]\} + \boldsymbol{\alpha}\Big)\Big\}}{\Gamma\Big\{\sum\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}}]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}} \oplus j]\} + \boldsymbol{\alpha}\Big)\Big\} \cdot \Gamma\Big\{\sum\Big(h\{\mathcal{D}[\mathbf{r},\, j]\} + \boldsymbol{\alpha}\Big)\Big\}} \\
&\quad \cdot \frac{(w_{\bar{j}} + \zeta_1^{\mathrm{c}})}{(m - w_{\bar{j}} - 1 + \zeta_0^{\mathrm{c}})} \\
&= \frac{\Gamma\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}}]\} + h\{\mathcal{D}[\bar{\mathbf{r}},\, \mathbf{c}_{\bar{j}} \oplus j]\} + \boldsymbol{\alpha}\Big) \cdot \Gamma\Big(h\{\mathcal{D}[\mathbf{r},\, j]\} + \boldsymbol{\alpha}\Big)}{\Gamma\Big(h\{\mathcal{D}[\cdot,\, \bar{\mathbf{c}}_{\bar{j}} \oplus j]\} + h\{\mathcal{D}[\bar{\mathbf{r}}_{\bar{j}},\, \mathbf{c}]\} + \boldsymbol{\alpha}\Big)} \\
&\quad \cdot \frac{\Gamma(n \cdot m - v \cdot w_{\bar{j}} + \sum \boldsymbol{\alpha})}{\Gamma(n \cdot m - v \cdot w_{\bar{j}} - v + \sum \boldsymbol{\alpha}) \cdot \Gamma(v + \sum \boldsymbol{\alpha})} \cdot \frac{(w_{\bar{j}} + \zeta_1^{\mathrm{c}})}{(m - w_{\bar{j}} - 1 + \zeta_0^{\mathrm{c}})} \\
&\quad j = 1 \ldots m. \qquad (5.32)
\end{aligned}
$$

Note that the equation cannot be simplified, because the conditions for the approximation in Equation 5.23 to hold is no longer satisfied. Intuitively, the denominator of Equation 5.32 assumes the current prediction of the background and extends the current bicluster by treating the jth condition as one of the biclustering conditions, while the second term in Equation 5.32 adds the jth condition to the currently assigned background. Again, the current bicluster and background refers to the data divided according to $\mathcal{D}_R$ and $\mathbf{C}_{\bar{j}}$.

5.4 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 human expert 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 in the same form the prior. 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 pseudo counts. 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 written as

$$\boldsymbol{u} = l \cdot [v_1, v_2, v_3]^{\mathrm{T}} = l \cdot \boldsymbol{v}, \tag{5.33}$$

where $\sum_{i=1}^{3} v_i = 1$. In this way, a Dirichlet vector is represented by two terms: (1) a frequency term, $\boldsymbol{v}$, which represents the frequencies of observing the three discrete levels; and (2) a term of multiplicand, l, which can be interpreted as the amount of data from which $\boldsymbol{v}$ is observed. Thus, by changing l of a Dirichlet prior, we imposes the strength of our prior knowledge on observing the frequency pattern $\boldsymbol{v}$.

5.5 Biclustering for global pathological pattern discovery

5.5.1 Construction of priors

The 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 to the model of the bicluster, and thus allow the algorithm to discover natural pattern of biclusters that are embedded in the data. Therefore, we use weak priors for the bicluster. By default, we set,

$$\boldsymbol{\beta}_j = \frac{1}{\sqrt{n}} \cdot \frac{h(\mathcal{D})}{n \cdot m}, \quad j = 1 \ldots w. \tag{5.34}$$

Note that $\frac{h(\mathcal{D})}{n \cdot m}$ stands for the frequencies to observe the discrete levels in the entire data.

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

$$\boldsymbol{\alpha} = \sqrt{n} \cdot \frac{h(\mathcal{D})}{n \cdot m}. \tag{5.35}$$

Though the above two equations provide good base 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 5.33) 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,

$$\boldsymbol{\alpha} = s^{\alpha} \cdot \sqrt{n} \cdot \frac{h(\mathcal{D})}{n \cdot m}, \tag{5.36}$$

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

5.5.2 Experiments on synthetic data

Data

We use a synthetic data set to show how algorithm works in practice and to illustrate the influence of parameters. We embedded a pattern of 25 rows by 8 columns (see Figure 5.1(d)) into a data set of size 100 by 30 (see Figure 5.1(b)). The pattern was described by eight sharp multinomial distributions, while the background was generated from a multinomial distribution close to a uniform distribution,

$$\mathbf{\Phi}^{\text{true}} = \begin{bmatrix} 0.05 & 0.9 & 0.03 & 0.05 & 0.03 & 0.9 & 0.05 & 0.05 \\ 0.9 & 0.07 & 0.07 & 0.9 & 0.07 & 0.07 & 0.9 & 0.9 \\ 0.05 & 0.03 & 0.9 & 0.05 & 0.9 & 0.03 & 0.05 & 0.05 \end{bmatrix}, \quad (5.38)$$

$$\Psi^{\text{true}} = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix}. \quad (5.39)$$

Discovery of a bicluster

We ran the algorithm for 500 iterations on the data set under the parameter settings $s^{\alpha} = 0.3$ and $s^{\beta} = 5$. We calculated the autocorrelation time (see Equation 4.18) for each of the Bernoulli parameters for 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 5.2 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 a autocorrelation time smaller than 3. Therefore, we consider the Markov chains to have converged well and that enough samples have been collected to carry out the Monte Carlo integration. From the trace plot of the likelihood and from monitoring the Bernoulli parameters of the full conditional distributions, we decide that convergence had been reached by the end of the first 50 iterations, (see Figure 5.3). Thus, we used 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 5.1. Figure 5.1(a) provides a graphic veiw of the posterior probability for each position in the data matrix that it belongs to the bicluster, $P(\mathcal{D}_R[i] = 1, C_j = 1 \,|\, \mathcal{D})$, which is obtained by

$$\begin{aligned} & P(\mathcal{D}_R[i] = 1, C_j = 1 \,|\, \mathcal{D}) \\ &= \sum_{t=1}^{T} P(\mathcal{D}_R[i] = 1 \,|\, \mathcal{D}_R[\bar{i}]^{(t)}, \mathbf{C}_m^{(t)}, \mathcal{D}) \cdot P(C_j = 1 \,|\, \mathcal{D}_R^{(t)}, \mathbf{C}_{\bar{j}}^{(t)}, \mathcal{D}). \end{aligned} \quad (5.40)$$

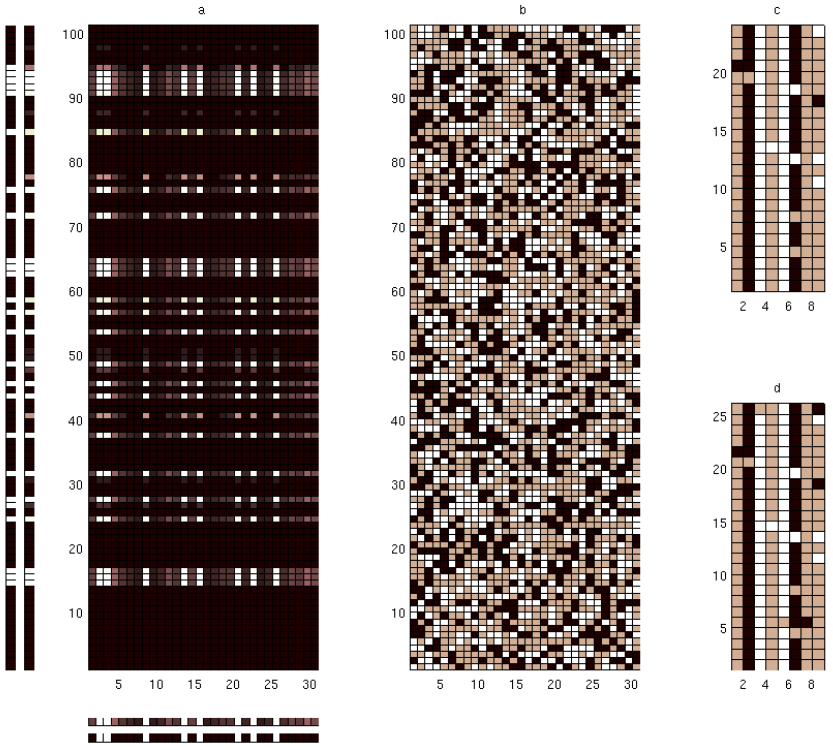


Figure 5.1: Results from the synthetic data set. (a) Main plot: The posterior probability that a position of the data matrix belongs to the bicluster. Inner bars: expected values of the labels. Outer bars: positions of the embedded pattern. (b) The data matrix. (c) Pattern of the bicluster revealed by the Gibbs sampling algorithm. (d) Pattern of the embedded bicluster.

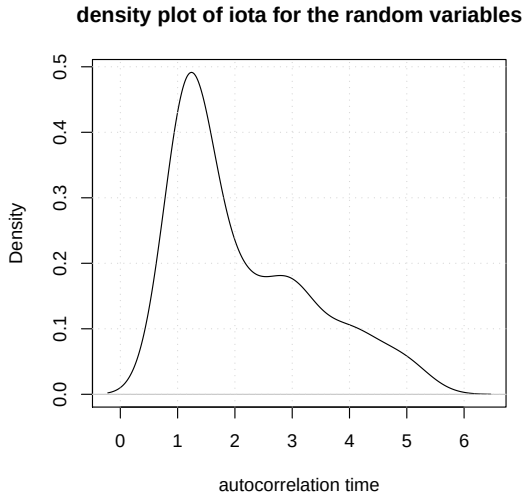


Figure 5.2: Density plot of the autocorrelation time of the Bernoulli paramters for the full conditional distribution of the 130 random variables under concern, i.e. those that are included in $\mathcal{D}_R$ and $\mathbf{C}_m$.

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 5.1(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 4.13. The outer bars mark the embedded positions of the bicluster by a white tag. Figure 5.4 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). Based on Figure 5.4, we use 0.8 as the threshold for the rows, and 0.9 as the threshold for the column, and consider the positions of the target bicluster to be the ones that possess higher expected values than the thresholds at both dimensions. The final pattern of the bicluster revealed by our algorithm is shown in Figure 5.1(c), which resembles the embedded bicluster (seed Figure 5.1(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 a 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 5.1 (a). (More illustrations will be provided in this respect in the following section.) This is a typical characteristic of Gibbs sam-

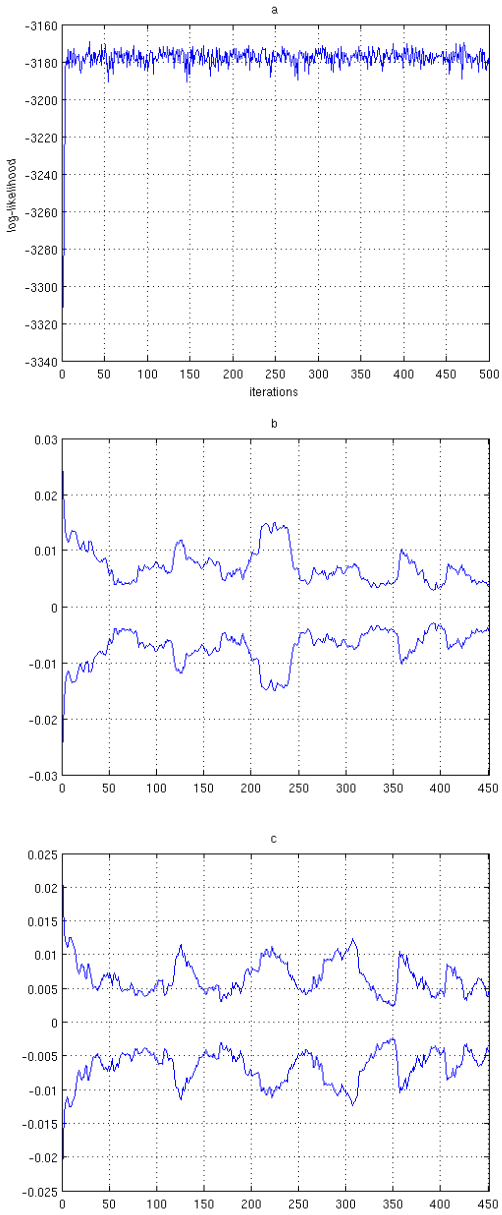


Figure 5.3: (a) Trace of the log likelihood of the synthetic data evaluated at the end of each iteration during the whole Gibbs sampling procedure. (b) and (c) reflects the evolution of the $P(\mathcal{D}_R[i] = 1 \mid \mathcal{D})$ and $P(C_j \mid \mathcal{D})$. For every random variable, we estimated $P(\mathcal{D}_R[i] = 1 \mid \mathcal{D})$ 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 deviations.

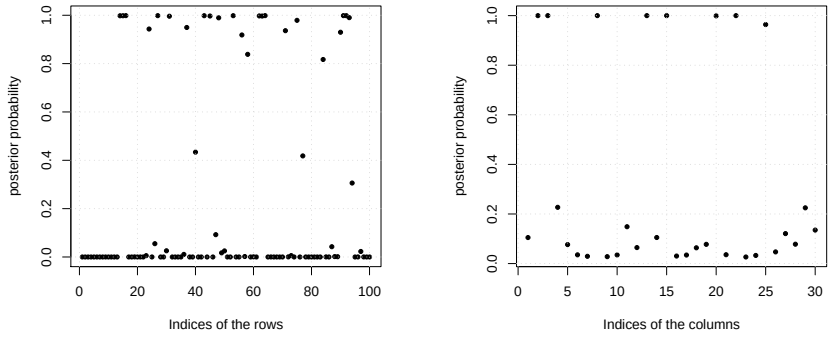


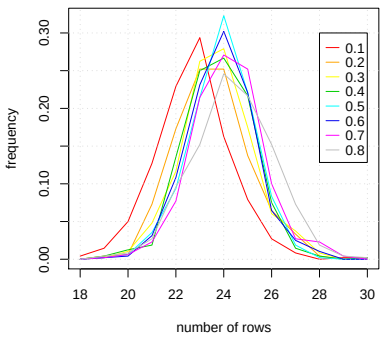
Figure 5.4: Plots of posterior probabilities of the random variables—left: $\mathcal{D}_R$, right: $\mathbf{C}_m$.

pling, 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.

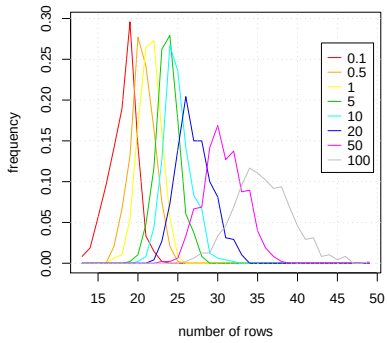
Influence of s^α and s^β

The user input parameters s^α and s^β 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 bilcustering 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 $s^\beta = 5$ remaines fixed.

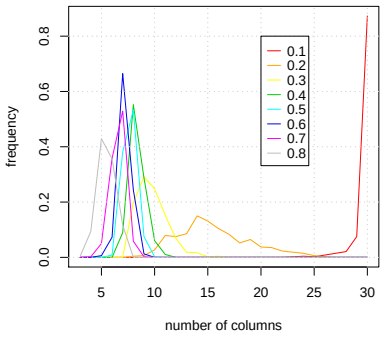
Figure 5.5 illustrates the impact of s^α and s^β on the size of the bicluster found in each iteration. The figure shows that s^α has limited influence on the number of rows discovered in each bicluster (see Figure 5.5(a). However, it is highly related to the decision of the number of columns included in the bicluster. A bigger s^α puts a stringer criterion on the selection of the columns, and as a result, fewer columns are included in the bicluster (see Figure 5.5(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. On the other hand, s^β has little influence on the number of rows selected for the bicluster (see Figure 5.5(d)). Rather, it influences the number of rows 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 5.5(b)). This phenomenon is explained by Equation 5.37, which implies that the resemblance between the bicluster and the background increases with a larger s^β.



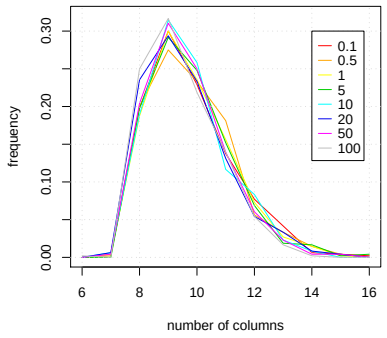
(a) Influence of s^{α} on the number of rows



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

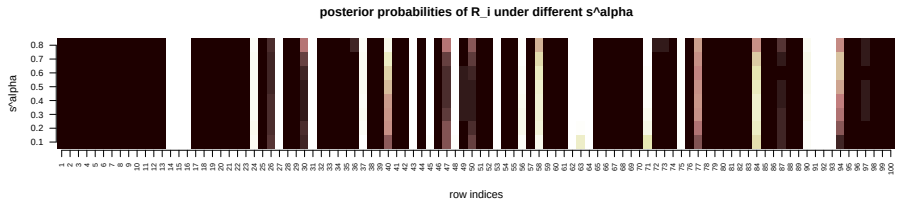


(c) Influence of s^{α} on the number of columns

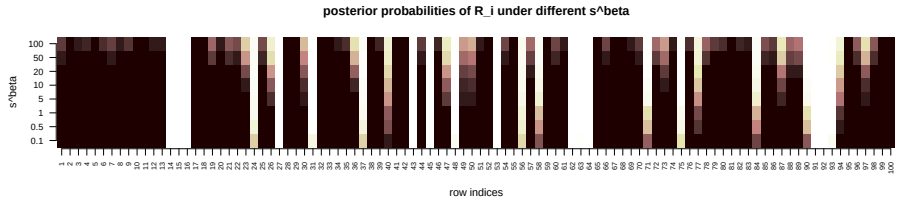


(d) Influence of s^{β} on the number of columns

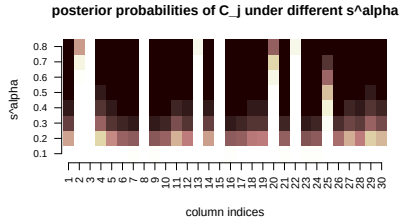
Figure 5.5: Frequency plots of the number of rows (or the number of columns) in the found bicluster of an iteration under different parameter settings. The differently colored lines correspond to different parameter settings.



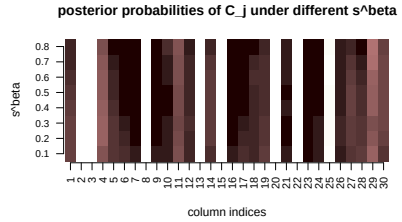
(a) Influence of s^{α} on the posterior of $\mathcal{D}_R$



(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 5.6: Heat map 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 heat map, 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.

Figure 5.6 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 5.6(a)) and that s^β has little influence on $P(\mathbf{C}_m \mid \mathcal{D})$ (see Figure 5.6(d)). However, with a smaller s^α, the posterior probability for a column to belong to the bicluster increases (see Figure 5.6(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 5.6(d)).

Combining the information from Figure 5.5 and Figure 5.6, we can infer that when when s^α is relativey 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. 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.

5.5.3 Case study: biclustering expriments 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 expressional fingerprints of gene 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 and the 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 5.1.

*We favored the preprocessing procedure discribed 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 algorithm stopped after discovering three biclusters one after another. 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 5.7 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 5.8). The third bicluster consisted of 17 (out of 20) MLL patients and 34 genes (see Figure 5.9). 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 expressional finger print 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 permutated 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 ambled 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 the 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 is recovered by at least one of the found biclusters. In general, the expression characters of these genes are in accordance to 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 than 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

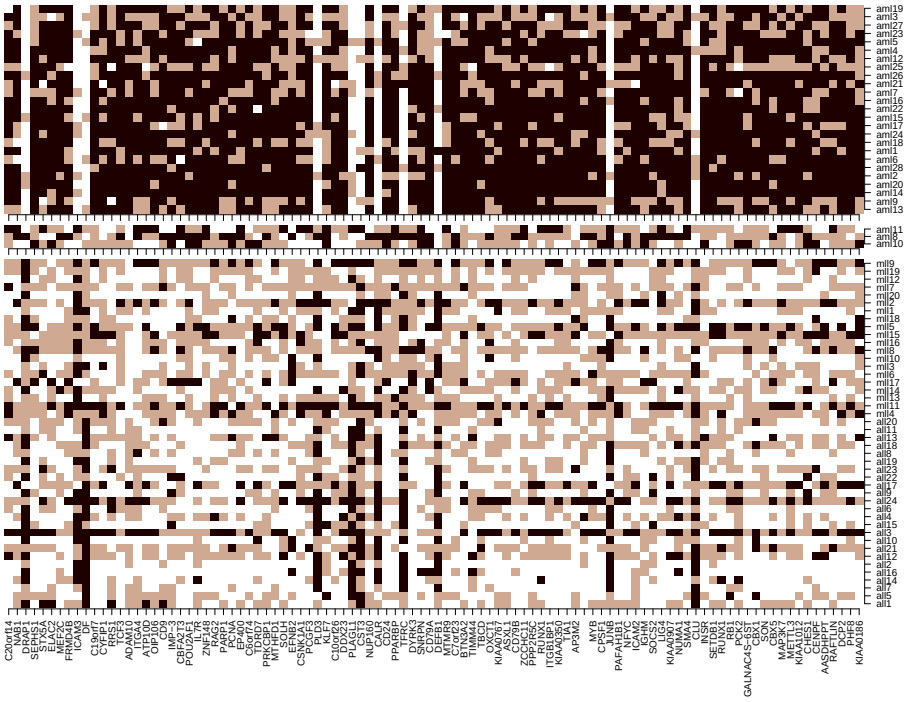


Figure 5.7: The first found bicluster consists of 25 out of 28 AML patients, whose discrete gene expression pattern is illustrated in the top figure as a heatmap, where the rows represent the patients, and the columns represent the genes (100 genes are randomly selected from the 444 genes that are included in the bicluster for the purpose of illustration). Black is used to represent the discrete level "low", brown for "middle", 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.

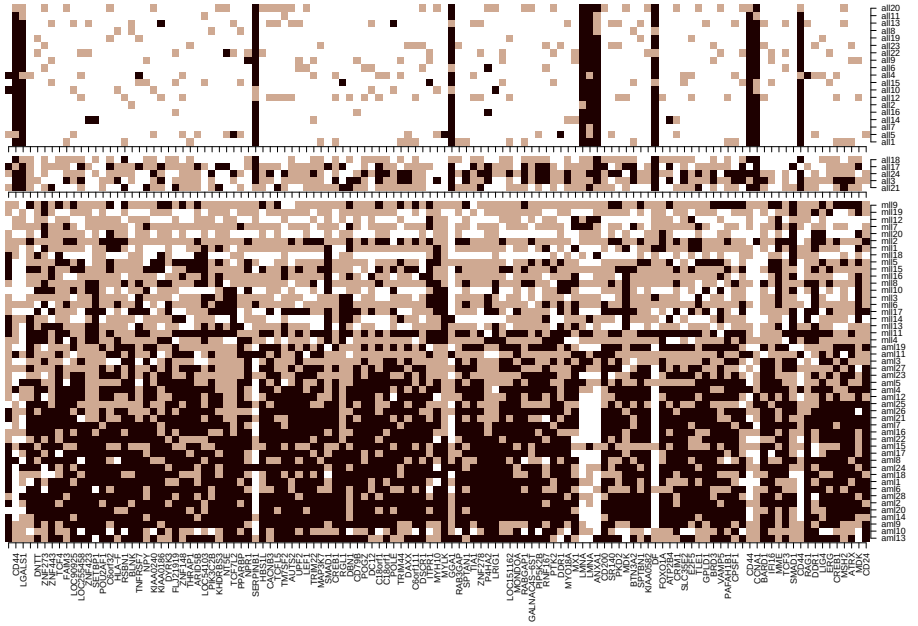


Figure 5.8: The second found bicluster consists of 19 out of 24 ALL patients, whose discrete gene expression pattern is illustrated in the top figure as a heatmap, where the rows represent the patients, and the columns represent the genes. Black is used to represent the discrete level ”low”, brown for ”middle”, and white for ”high”. The heatmap in the middle of the figure shows the discrete expression pattern of the 5 left out 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.

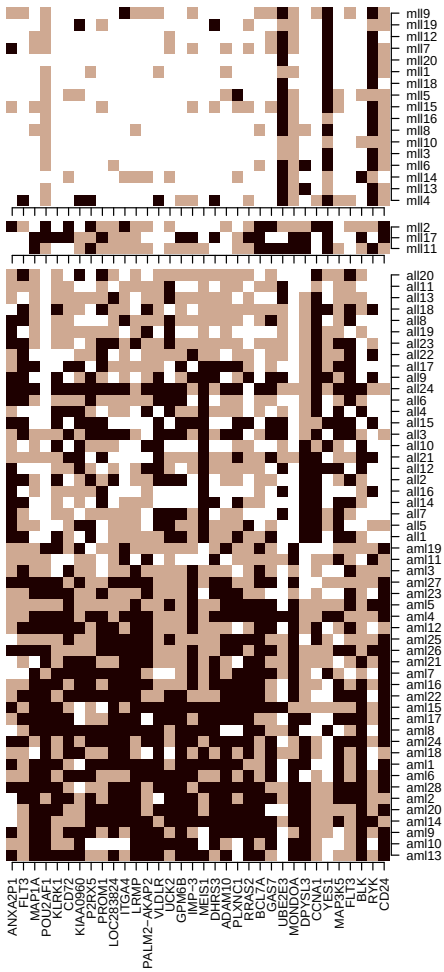


Figure 5.9: The third found bicluster consists of 17 out of 20 MLL patients, whose discrete gene expression pattern is illustrated in the top figure as a heatmap, where the rows represent the patients, and the columns represent the genes. Black is used to represent the discrete level ”low”, brown for ”middle”, 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 not included in the final data set		
LGALS1	Genes encoding certain adhesion molecules	Relatively over-expressed in MLL	–	low	–
ANXA1			–	low	–
ANXA2			–	–	–
CD44			–	low	–
SPN			Gene not included in the final data set		
PROML1	Genes expressed in progenitors	Highly over-expressed in MLL	high	–	–
FLT3			high	–	–
LMO2			Gene not included in the final data set		
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 not included in the final data set		
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 descretized expression level 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 not included in the final data set		
DF	myeloid-specific genes	AML	–	low	high
CTSD			Gene not included in the final data set		
ANPEP			–	–	high

[a]The biclusters discovered by our algorithm are associated with the patient groups that they represent. Shown under this column are the descretized expression level 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 [22] of biological processes based on a hypergeometric distribution (see Chapter 3). Putting a limit of 0.05 on the p-values of the GO terms, the 444 genes included in the bicluster which characterizes AML patients are over-repreresented in 162 GO terms of biological process. Figure 5.10 shows the hierarchical structure of these over-represented GO terms. 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 recombination. 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.

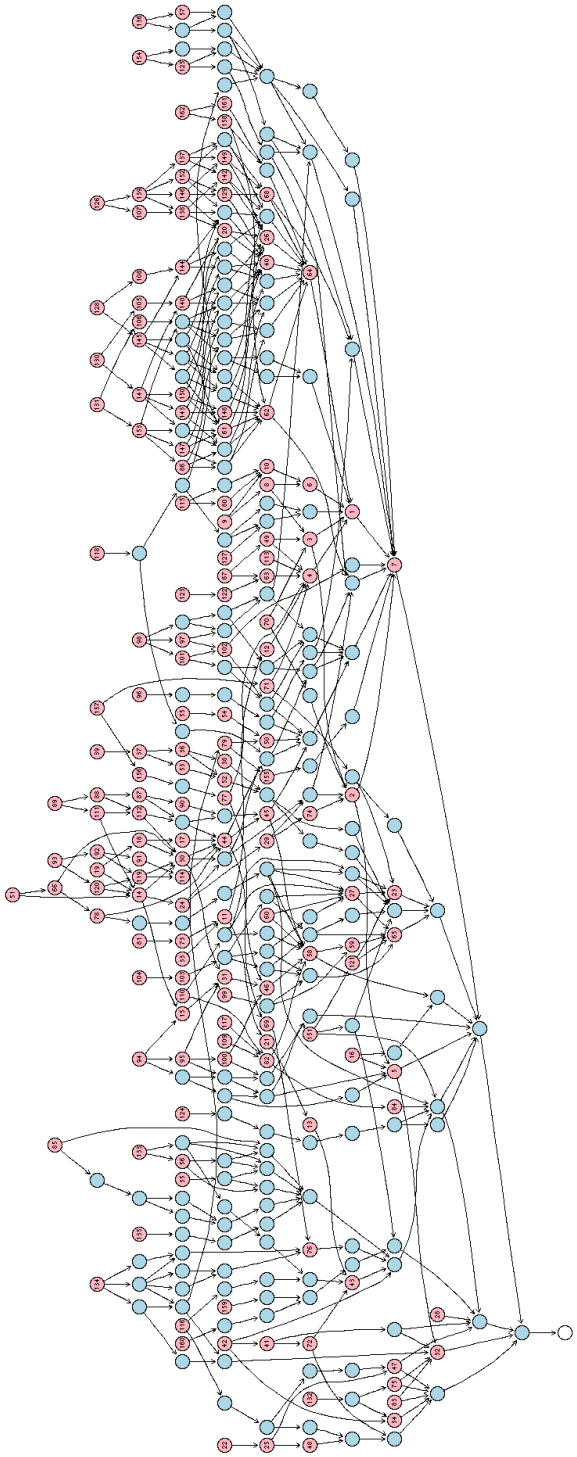


Figure 5.10: GO graph for over-represented GO terms (pink nodes) of biological process for the genes that are included in the bicluster that characterize AML patients.

rank	GO term	p-value	k^{a}	f^{b}
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] Number of LocusLink ID's, corresponding the genes in the bicluster, that are annotated with the GO term of biological process.

[b] 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 5.3: Some Go terms of biological processes that are highly related to leukemia that are associated with genes selected for the bicluster corresponding to AML patients.

rank	GO term	p-value	k^{a}	f^{b}
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]Number of LocusLink ID's, corresponding the genes in the bicluster, that are annotated with the GO term of biological process.

[b]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 5.4: Some Go terms of biological processes that are highly related to leukemia that are associated with genes selected for the bicluster corresponding to AML patients (continued).

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 5.11 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 5.11, 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 leukemia-relevant.

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 pathological meaningful interpretation. However, pathological evidence is relevantly 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 due to 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, that 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.

5.6 Query-driven biclustering for pathological pattern discovery

5.6.1 Construction of priors

As we mentioned, the task of query-driven biclustering of experiments is that 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 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 finger print for the

Figure 5.11: 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	GO term	p-value	k^{a}	f^{b}
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] Number of LocusLink ID's, corresponding the genes in the bicluster, that are annotated with the GO term of biological process.

[b] 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.

[c] Whether the GO term is leukemia-related.

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	GO term	p-value	k^a	f^b
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]Number of LocusLink ID's, corresponding the genes in the bicluster, that are annotated with the GO term of biological process.

[b]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.

[c]Whether the GO term is leukemia-related.

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).

pathology. This type of tools is useful for biologists and doctors to retrieve information from microarray data when only a small amount of (tumor) samples can be confirmed by traditional means to belong to the specific pathological type of interest.

To incorporate information of 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 target pathological type. In the meantime, the soft query means that seed patients (or seed tumor samples) that do not belong 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 $\boldsymbol{\beta}_j$

$$\boldsymbol{\beta}_j = s^{\beta} \cdot \frac{h(D[\mathbf{a}, j]) + 0.001}{n}. \tag{5.41}$$

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

However, for the background model, we keep the parameter setting as described in Section 5.5.1, assuming that the background model is similar to the frequency model observed in the whole data set, i.e.

$$\boldsymbol{\alpha} = \sqrt{n} \cdot \frac{h(\mathcal{D})}{n \cdot m}. \tag{5.42}$$

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 the background.

$$\mathcal{D}_R[i] = \begin{cases} 1, & i \in \mathbf{a} \\ 0, & i = 1 \dots n,\ i \notin \mathbf{a}. \end{cases} \tag{5.43}$$

However, for the column dimension of the data matrix (i.e., for the genes), we initialize by randomly assigning them to either the bicluster or the background.

5.6.2 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 also differs from one to another. The resulting data is shown in Figure 5.12, 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 5.5. We used five different sets of parameters, under each of which biclustering algorithm was performed 10 times. 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.

We now explore the ability of the algorithm to discover the two non-dominant biclusters 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. In order 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 Biclustser 1. This frequency model is concluded as $\mathbf{\Phi}^{\text{true}}$,

$$\Phi_j^{\text{true}} = \frac{h\{\mathcal{D}[\mathbf{r}^{\text{true}}, \delta_{40}(j)]\} + 0.001}{200} \quad j \in \mathbf{c}^{\text{true}}, \tag{5.44}$$

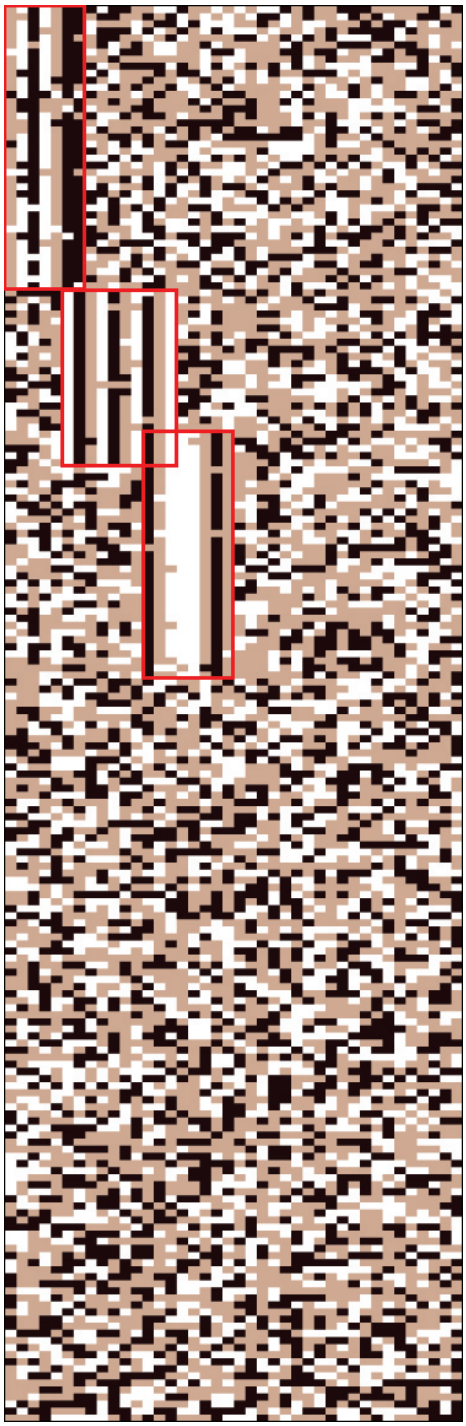


Figure 5.12: Synthetic data set with three biclusters of different size and consistency are embedded. The rows and the columns of the data set are rearranged to manifest the biclusters

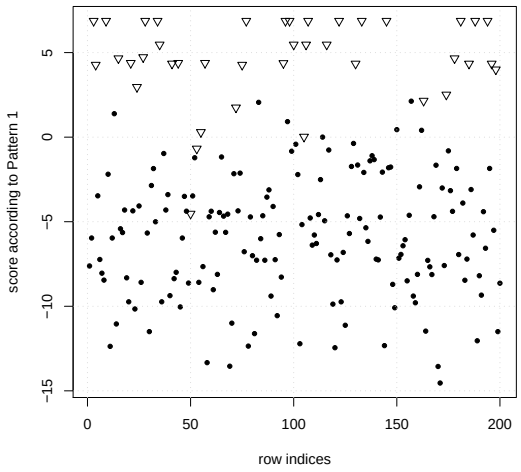


Figure 5.13: Scores of the rows in the whole data set measuring their similarity with the embedded Biclsuter 1. The triangles represent the rows where Bicluster 1 is embedded, and the dots represent the rest of the rows in the data set.

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,

$$d[i] = \prod_{j \in \mathbf{c}^{\text{true}}} \left(\frac{\mathbf{\Phi}^{\text{true}}}{\Psi^{\text{true}}} \right)^{h[\mathcal{D}[i,j]]}, \quad i = 1 \ldots 200. \tag{5.45}$$

The scores of the rows in the whole data set is plotted in Figure 5.13. 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 5.14.

For each set of seeds, we first test the frequency that Bicluster 1 is found back by the algorithm under different parameter settings. We found that the influence of s^α remains the same as explained in Section 5.5.2. That is, a larger s^α results in fewer columns to be selected for the bicluster, and data under the selected columns exhibits highest consistency for those rows that are included in the bicluster. Therefore, we focus our discussion below on the influence of s^β. We fixed $s^\alpha = 0.3$, and performed the algorithm under 8 different values of s^β. For



(a) Seed 1 for Bicluster 1

(b) Seed 2 for Bicluster 1

(c) Seed 3 for Bicluster 1

Figure 5.14: Patterns of 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.

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.

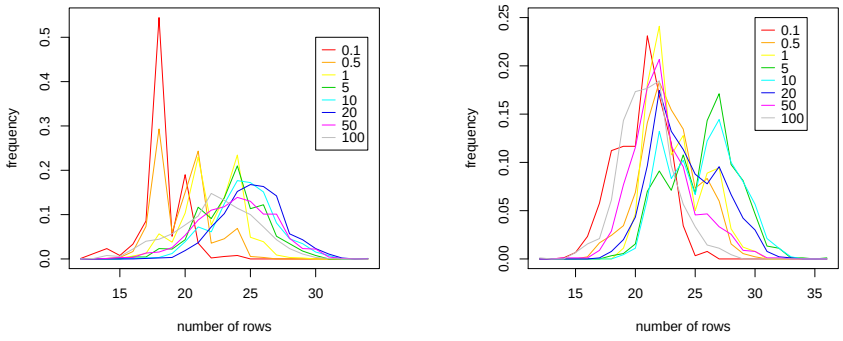
each set of seeds and under the same parameter settings, we ran the algorithm 20 times, and count the number of times that Bicluster 1 is recovered. The result is concluded in Table 5.8.

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 when 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 with the decrement of s^β. The decrement in the consistency of seed rows reduce 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 of the seed rows (to Bicluster 1) are no higher than the maximum score of the rest of the rows (see Figure 5.13)—a smaller s^β, however, helps to recover the embedded bicluster. That is because a big s^β only emphasizes the noisiness 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 large s^β (i.e. ≥ 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 5.15 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 discovered bicluster 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 4.50) helps to fine-tune the final selection of rows.

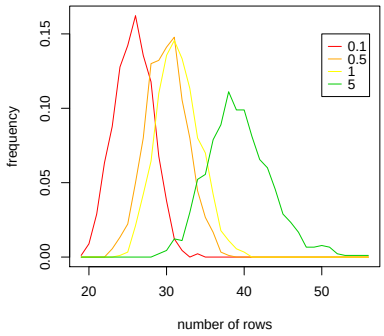
Note that the algorithm is able to exclude seeds that are not consistent with the discovered bicluster. For instance, when Seed 3 is used, those rows as the seeds often have very low posterior probabilities to be included in the final bicluster.

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 Biclustrer 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 increment in the proportion of unique rows in the seed raises the chance to



(a) Seed 1

(b) Seed 2



(c) Seed 3

Figure 5.15: 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}.

# 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 Biclustrer 2 overlaps with Bicluster 3. Each set of seeds consists five rows. The first row in the table shows the number of rows in the seed where Bilucster 2 is uniquely embedded.

find the intended bicluster.

5.6.3 Case study: query-driven biclustering of leukemia patients

The paper of Yeoh *et al.* (2002) [102] 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) [102] contains gene expression profiles measured on 327 patients, of whom 15 are BCR-ABL patients, 27 are E2A-PBX1 patients, 64 belong to the hyperdiploid-over-50-chromosomes group, 20 are in the MLL-rearrangement group, 43 are T-ALL patients, 79 are TEL-AML patients, and 79 are of other pathological categories. We preprocessed the data as described in the original paper [102]. Further, we only retained the genes those whose variation 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 5.1.

We first applied the biclustering algorithm (as described in Section 5.5) to discover global patterns in the data, to get an idea of the manner of the pattern discovery under non-informative prior on the data. Clusters found in earlier rounds are masked in order to discover multiple patterns. The whole procedure—biclustering, masking, biclustering, and so on—is terminated when a biclusters not in correspondence to any of the patient group is found found. This procedure was repeated 20 times in total. Table 5.10 summarizes the frequency of discovering a biclustrer representing a certain patient group. The

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 corresponds 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.

table shows that biclusters resulted from the artifact of the equal-frequency discretization—i.e., those biclusters whose columns depend on their position in the data set—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 5.16 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) [102] and Armstrong *et al.* (2002) [4]. Figure 5.17 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 5.18.

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-dominent patterns embedded in the microarray data can be greatly enhanced.

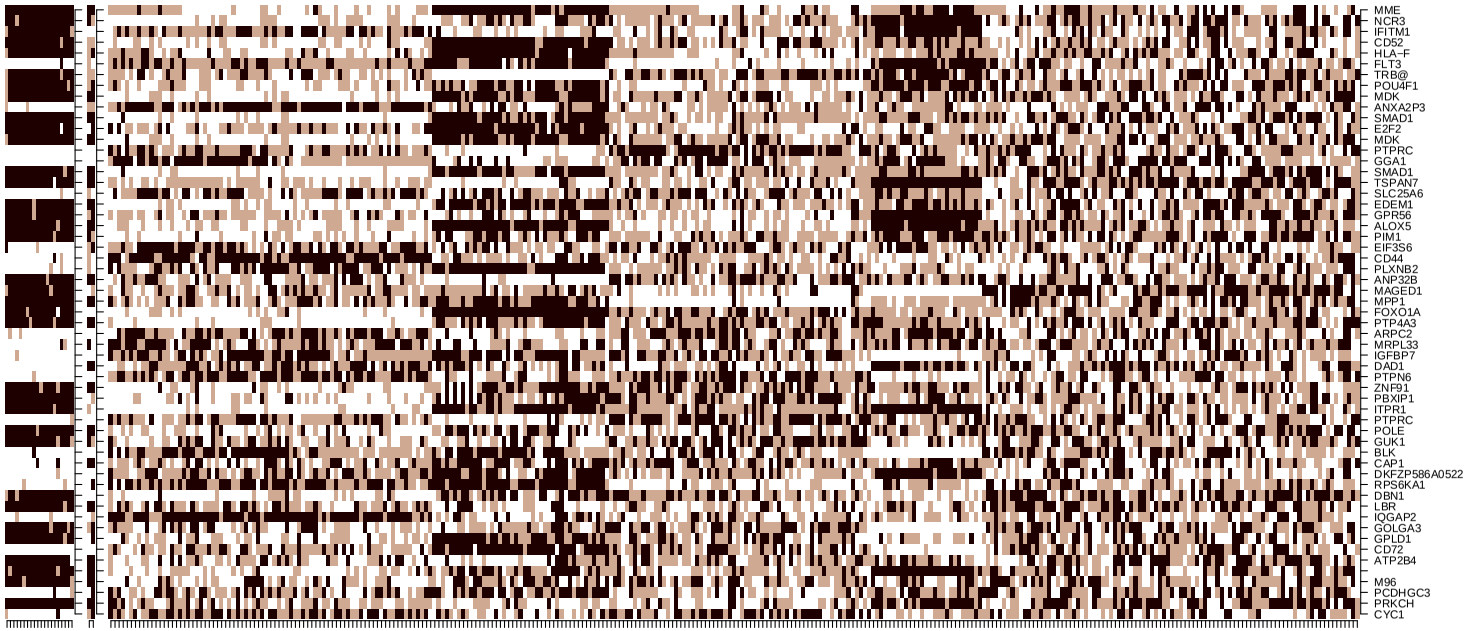


Figure 5.16: The left-most heat map shows the pattern of the bicluster which includes 18 MLL patients and 2 patients from the hyperdiploid > 50 group (represented by the columns) and 57 genes (represented by the rows). Black represents the discrete expression leve ofl ”low”, brown for ”middle”, and white 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.

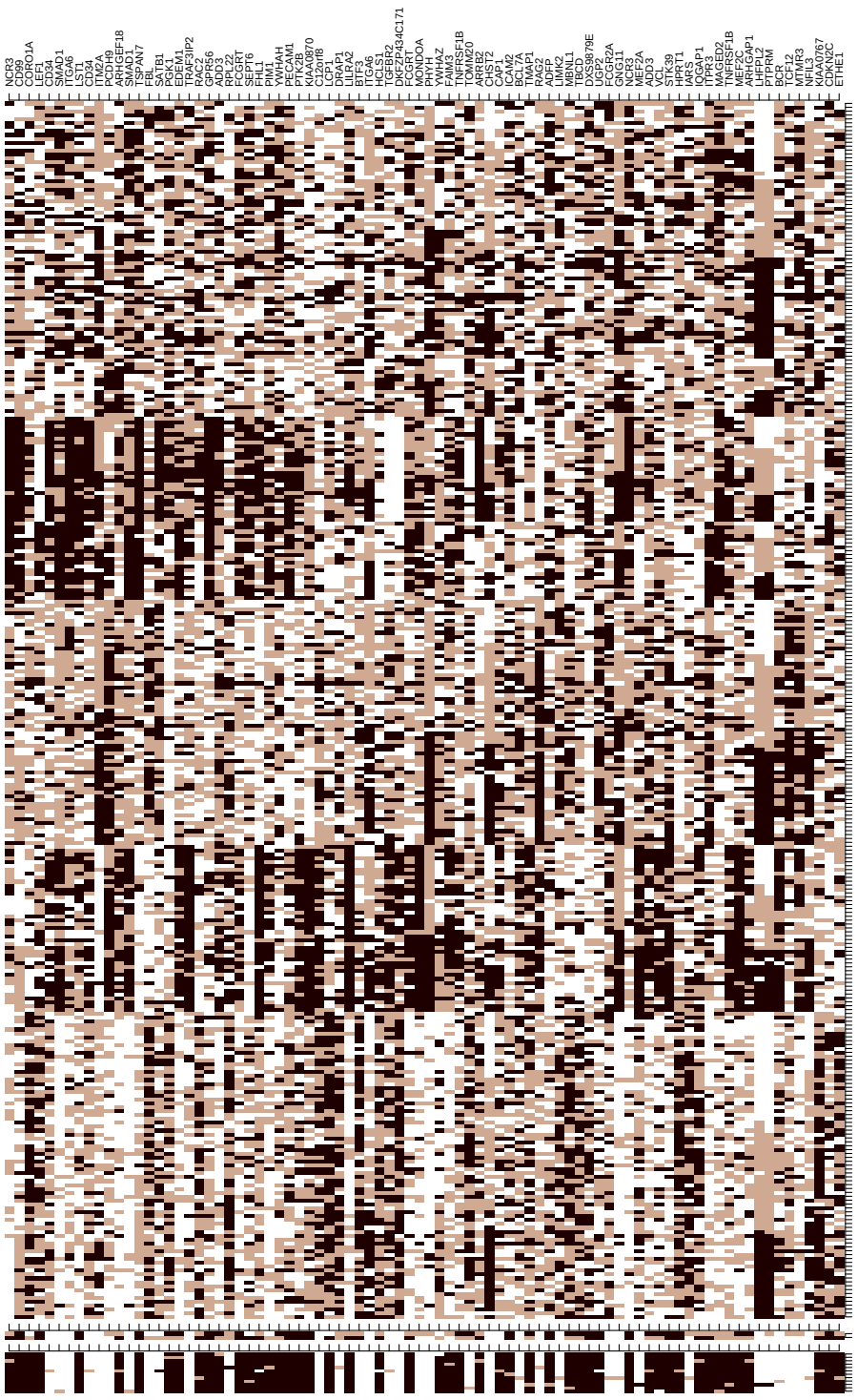


Figure 5.17: The left-most heat map shows the pattern of the bicluster which includes 12 patients of the novel group (represented by the columns) and 84 genes (represented by the rows). Black represents the discrete expression leve ofl "low", brown for "middle", and white for "high". The heatmap in the 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.

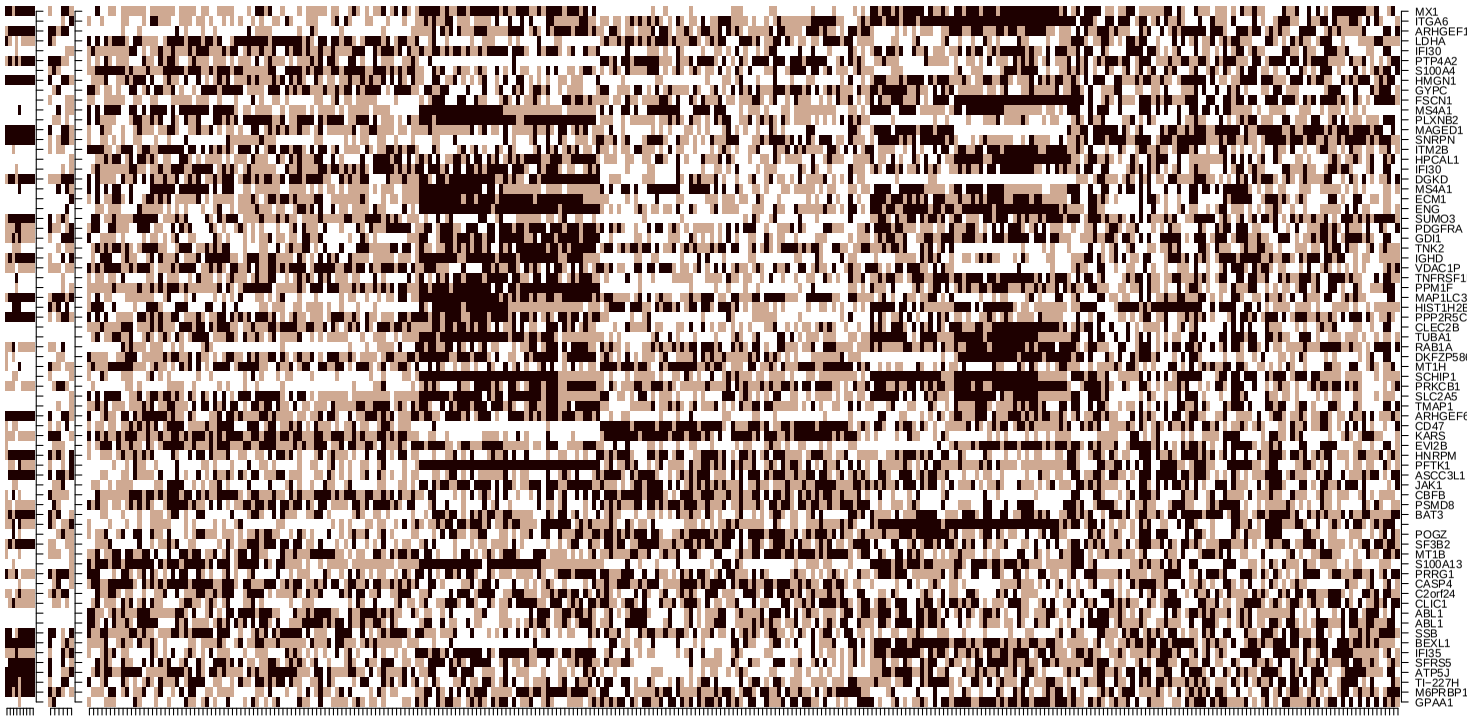


Figure 5.18: The left-most heat map shows the pattern of the bicluster which includes 10 BCR-ABL patients (represented by the columns) and 71 genes (represented by the rows). Black represents the discrete expression leve ofl ”low”, brown for ”middle”, and white for ”high”. The heatmap in the middle of 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 based on microarray data

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, 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 bisects the genes in the micorarray data based on their expression profiles under a selected subset of conditions. The decision of the bisection 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 amount of modes. In our case, the probabilistic mixture is made up of a bicluster component and a background component, which means that each mode in the mixture model can be selected as a bicluster (and consequently the other modes combined as the background).

As described later on in the chapter, same hierarchical model structure is used for both the bicluster and the background. This means that when non-informative prior is used, the Gibbs sampling procedcure tends to bisect the genes corresponding to the two global modes in the maximum likelihood function. Genes included in either of the bisector are often of little interest to biologists. However, we should recognize that for a biologist, one of the main question is always, "what are the genes that are related to a *particular* function (or in a *specific* pathway) of interest to me?"

The strength of 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 specified 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 into the local area of interest of the likelihood landscape, and raise the corresponding area in the posterior distribution.

We consider one possible situation when 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, under 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 imposed prior model is strong enough, the maximum mode of the target joint posterior distribution (see Equation 4.41) 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.
- *Gaussian-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.

We illustrate the ability of our biclustering strategy to assist the discovery of regulatory modules in transcriptional network. 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 where the discussed layout of biclusters are embedded. Finally, we use a case study on a yeast data set illustrate our strategy for transcriptional regulatory module discovery as a whole.

6.1 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 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, we would use different distributions to describe different experiments.

We use $\{X_j \mid j = 1 \dots m\}$ 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 variable $\{\mathbf{Y}_k \mid k = 1 \dots q\}$ (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,

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

Adjusting our Bayesian hierarchical model introduced in Section 4.2 for this change, the nodes for observed data in Figure 4.4 now represents $\mathbf{Y}_1, \dots, \mathbf{Y}_q$. Consequently, the notation for the set of Bernoulli variables to describe whether a node belongs to the bicluster changes to $\mathbf{C}_q = \{C_k \mid k = 1 \dots q\}$. To be explicit, the bicluster model now is,

$$p(\mathbf{Y}_k) = \begin{cases} p(\mathbf{Y}_k \mid R = 1) = f(\mathbf{\Theta}_k^{\text{bcl}}) & C_k = 1,\ R = 1 \\ p(\mathbf{Y}_k \mid R = 0) = f(\mathbf{\Theta}_k^{\text{bgd}}) & C_k = 1,\ R = 0 \\ p(\mathbf{Y}_k) = f(\mathbf{\Theta}_k^{\text{bgd}}) & C_k = 0 \end{cases}, \quad k = 1 \dots q, \tag{6.2}$$

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

To facilitate the notations, 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 indices of those that belong to the background; $\mathbf{c}$ and $\bar{\mathbf{c}}$, however, now denote the indices of 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

$$\mathbf{e} = [j \mid j \in \mathbf{e}_k, \forall k = \{1 \dots q\} \wedge C_k = 1] \tag{6.3}$$

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

$$\mathbf{e} = [j \mid j \in \mathbf{e}_k, \forall k = \{1 \dots q\} \wedge C_k = 0] \tag{6.4}$$

to notate the entire set of indices of the experiments whose corresponding condition are in the background.

6.2 Gaussian-Wishart model

We use Gaussian distributions to describe the expression data. This choice is based on not only the analytical convenience of normal models, but also the previous success in applying normal mixture models to the clustering problems of microarray data [103, 63]. 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 belongs to the bicluster under each experiment,

$$X_j \sim \mathrm{N}\left(\mu_j^{\mathrm{bcl}}, (\sigma_j^{\mathrm{bcl}})^2\right) \quad j \in \mathbf{e}. \tag{6.5}$$

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, μ_j's follow normal distributions,

$$\mu_j^{\mathrm{bcl}} \sim N\left(\varphi_j^{\mathrm{bcl}}, (\tau_j^{\mathrm{bcl}})^2\right) \quad j \in \mathbf{e}. \tag{6.6}$$

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,

$$(\sigma_j^{\mathrm{bcl}})^2 \sim \text{Inverse-}\chi^2\left(\nu^{\mathrm{bcl}}, (s^{\mathrm{bcl}})^2\right) \quad j \in \mathbf{e}. \tag{6.7}$$

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 6.1. Note that the hierarchical model for the bicluster as explained above (see Equation 6.5, Equation 6.7 and Equation 6.6 for a conclusion), as a whole, is represented by $\mathbf{\Theta}^{\mathrm{bcl}}$ in Equation 6.2.

We use the same hierarchical structure for the background (as illustrated in Figure 6.1),

$$\begin{aligned} X_j &\sim N\left(\mu_j^{\mathrm{bgd}}, (\sigma_j^{\mathrm{bgd}})^2\right) \quad j = 1 \ldots m, && (6.8)\\ \mu_j^{\mathrm{bgd}} &\sim N\left(\varphi_j^{\mathrm{bgd}}, (\tau_j^{\mathrm{bgd}})^2\right) \quad j = 1 \ldots m, && (6.9)\\ (\sigma_j^{\mathrm{bgd}})^2 &\sim \text{Inverse-}\chi^2\left(\nu^{\mathrm{bgd}}, (s^{\mathrm{bgd}})^2\right) \quad j = 1 \ldots m. && (6.10) \end{aligned}$$

Equations 6.9 to 6.10 concludes the hierarchical model for data in the background represented as $\mathbf{\Theta}^{\mathrm{bgd}}$ in Equation 6.2.

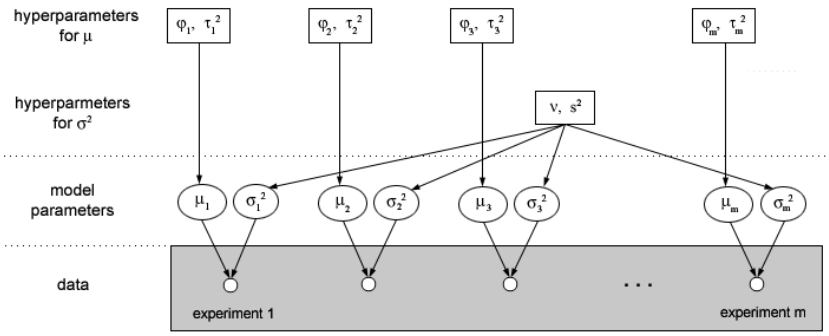


Figure 6.1: The same hierarchal model structure used for describing both the bicluster and the background for the problem of biclustering genes.

From now on, we use

$$\boldsymbol{\mu}^{\text{bcl}} = \{\mu_j^{\text{bcl}} \mid j \in \mathbf{e}\} \tag{6.11}$$

$$\boldsymbol{\mu}^{\text{bgd}} = \{\mu_j^{\text{bgd}} \mid j = 1 \ldots m\}. \tag{6.12}$$

The same system of notation is applied to $\boldsymbol{\sigma}^{\text{bcl}}$, $\boldsymbol{\sigma}^{bgd}$, $\boldsymbol{\varphi}^{\text{bcl}}$, $\boldsymbol{\varphi}^{\text{bgd}}$, $\boldsymbol{\tau}^{\text{bcl}}$ and $\boldsymbol{\tau}^{\text{bgd}}$.

6.3 Full conditional distributions

As explained in Section 4.3.2 (see Equation 4.41) the target joint distribution for this biclustering problem is

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

where

$$\boldsymbol{\Theta}^{\text{bcl}} = \{\boldsymbol{\mu}^{\text{bcl}}, (\boldsymbol{\sigma}^{\text{bcl}})^2\} \tag{6.13}$$

$$\boldsymbol{\Theta}^{\text{bgd}} = \{\boldsymbol{\mu}^{\text{bgd}}, (\boldsymbol{\sigma}^{\text{bgd}})^2\} \tag{6.14}$$

$$\xi^{\text{bcl}} = \{\boldsymbol{\varphi}^{\text{bcl}}, (\boldsymbol{\tau}^{\text{bcl}})^2, \nu^{\text{bcl}}, (s^{\text{bcl}})^2\} \tag{6.15}$$

$$\xi^{\text{bgd}} = \{\boldsymbol{\varphi}^{\text{bgd}}, (\boldsymbol{\tau}^{\text{bgd}})^2, \nu^{\text{bgd}}, (s^{\text{bgd}})^2\}. \tag{6.16}$$

This means that in the Gibbs sampling procedure, we need to iteratively sample from the full conditional distribution of each of the involved random 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 Gaussian-Wishart model is straightforward because of the use of conjugate priors. 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.

$$\begin{aligned}
&p(\mu_j \mid \mathcal{D}[\mathbf{u}, j],\, \sigma_j^2,\, \phi_j,\, \tau_j^2) \\
&\quad\propto\quad p(\mathcal{D}[\mathbf{u}, j] \mid \mu_j,\, \sigma_j^2) \cdot P\Big(\mu_j \mid \phi_j,\, \tau_j^2\Big) \\
&\quad=\quad \exp\left\{-\frac{|\mathbf{u}|}{2\sigma_j^2}\Big[\mu_j - \bar{\mu}_j\Big]^2\right\} \cdot \exp\left\{-\frac{1}{2\tau_j^2}\Big[\mu_j - \phi_j\Big]^2\right\} \\
&\quad=\quad \exp\left\{-\frac{1}{2}\left[\frac{\mu_j^2}{\frac{\sigma^2}{|\mathbf{u}|}} + \frac{\bar{\mu}_j^2}{\frac{\sigma^2}{|\mathbf{u}|}} - 2\frac{\mu_j \cdot \bar{\mu}_j}{\frac{\sigma^2}{|\mathbf{u}|}} + \frac{\mu_j^2}{\tau^2} + \frac{\phi_j^2}{\tau^2} - 2\frac{\mu_j \cdot \phi_j}{\tau^2}\right]\right\} \\
&\quad\propto\quad \exp\left\{-\frac{1}{2}\left[\mu_j^2\left(\frac{1}{\frac{\sigma^2}{|\mathbf{u}|}} + \frac{1}{\tau^2}\right) - 2\mu_j\left(\frac{\bar{\mu}_j}{\frac{\sigma^2}{|\mathbf{u}|}} + \frac{\phi_j}{\tau^2}\right)\right]\right\},
\end{aligned} \tag{6.17}$$

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 $b = \frac{1}{\frac{\sigma_j^2}{|\mathbf{u}|}}$,

$$\begin{aligned}
p(\mu_j \mid \mathcal{D}[\mathbf{u}, j],\, \sigma_j^2,\, \phi_j,\, \tau_j^2) \quad&\propto\quad \exp\left\{-\frac{1}{2}\Big[\mu_j^2\Big(a + b\Big) - 2\mu_j\Big(a \cdot \phi_j + b \cdot \bar{\mu}_j\Big)\Big]\right\} \\
&\propto\quad \exp\left\{-\frac{1}{2}\left[\mu_j^2\Big(a + b\Big) - \frac{2\mu_j \cdot \Big(a \cdot \phi_j + b \cdot \bar{\mu}_j\Big) \cdot (a + b)}{(a + b)} + \right.\right. \\
&\qquad\left.\left. \frac{\Big(a \cdot \phi_j + b \cdot \bar{\mu}_j\Big)^2 \cdot (a + b)}{(a + b)^2}\right]\right\}.
\end{aligned} \tag{6.18}$$

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

$$\begin{aligned}
p(\mu_j \mid \mathcal{D},\, \mathcal{D}_R,\, \mathbf{C}_q,\, \sigma_j^2,\, \phi_j,\, \tau_j^2) &= N(\hat{\mu}_j, \hat{\sigma}_j^2) \\
\hat{\mu}_j = \frac{\frac{\varphi_j}{\tau_j^2} + \frac{\bar{\mu}_j}{\frac{\sigma_j^2}{|\mathbf{u}|}}}{\frac{1}{\tau_j^2} + \frac{1}{\frac{\sigma_j^2}{|\mathbf{u}|}}} \quad &\text{and} \quad \hat{\sigma}_j^2 = \frac{1}{\frac{1}{\tau_j^2} + \frac{1}{\frac{\sigma_j^2}{|\mathbf{u}|}}}.
\end{aligned} \tag{6.19}$$

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

$$p(\sigma_j^2 \mid \mathcal{D}[\mathbf{u}, j],\, \mu_j,\, \nu,\, s^2) = \frac{p(\mathcal{D}[\mathbf{u}, j] \mid \mu_j,\, \sigma_j^2) \cdot p(\sigma_j^2 \mid \nu,\, s^2)}{p(\mathcal{D}[\mathbf{u}, j] | \mu_j,\, \nu,\, s^2)}. \tag{6.20}$$

The numerator of Equation 6.20 is,

$$\begin{aligned} & p(\mathcal{D}[\mathbf{u}, j] \,|\, \mu_j, \sigma_j^2) \cdot p(\sigma_j^2 \,|\, \nu, s^2) \\ = \; & \frac{1}{(\sqrt{2\pi}\sigma_j)^{|\mathbf{u}|}} \exp\left\{-\frac{\sum_{i\in\mathbf{u}}(\mathcal{D}[i,j]-\mu_j)^2}{2\sigma_j^2}\right\} \cdot \frac{2^{-\nu/2}}{\Gamma(\nu/2)} s^\nu \sigma_j^{-2(\frac{\nu}{2}+1)} \exp\left(-\frac{\nu s^2}{2\sigma_j^2}\right) \\ = \; & Z \cdot \sigma_j^{-2(\frac{\nu+|\mathbf{u}|}{2}+1)} \exp\left\{-\frac{\sum_{i\in\mathbf{u}}(\mathcal{D}[i,j]-\mu_j)^2 + \nu s^2}{2\sigma_j^2}\right\}, \end{aligned} \tag{6.21}$$

where $Z = \frac{1}{\sqrt{2\pi}} \cdot \frac{2^{-\nu/2}}{\Gamma(\nu/2)} s^\nu$. The denominator of Equation 6.20 is,

$$\begin{aligned} & p(\mathcal{D}[\mathbf{u}, j] | \mu_j, \nu, s^2) \\ = \; & \int p(\mathcal{D}[\mathbf{u}, j] \,|\, \mu_j, \sigma_j^2) \cdot p(\sigma_j^2 \,|\, \nu, s^2) \, \mathrm{d}\sigma_j^2 \\ = \; & Z \cdot \int \sigma_j^{-2(\frac{\nu+|\mathbf{u}|}{2}+1)} \exp\left\{-\frac{\sum_{i\in\mathbf{u}}(\mathcal{D}[i,j]-\mu_j)^2 + \nu s^2}{2\sigma_j^2}\right\} \mathrm{d}\sigma_j. \end{aligned} \tag{6.22}$$

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

$$\begin{aligned} p(\mathcal{D}[\mathbf{u}, j] | \mu_j, \nu, s^2) \; \propto \; & \int \left(\frac{1}{\sigma_j^2}\right)^{\frac{\nu+|\mathbf{u}|}{2}+1} \exp\left(-\frac{t}{2\sigma_j^2}\right) \mathrm{d}\sigma_j^2 \\ = \; & \int \frac{2}{t}\left(\sigma_j^2\right)^2 \left(\frac{1}{\sigma_j^2}\right)^{\frac{\nu+|\mathbf{u}|}{2}+1} \exp\left(-\frac{t}{2\sigma_j^2}\right) \mathrm{d}\left(\frac{t}{2\sigma_j^2}\right) \\ = \; & \int \left(\frac{t}{2}\right)^{-1} \left(\frac{1}{\sigma_j^2}\right)^{\frac{\nu+|\mathbf{u}|}{2}-1} \exp\left(-\frac{t}{2\sigma_j^2}\right) \mathrm{d}\left(\frac{t}{2\sigma_j^2}\right) \\ = \; & \left(\frac{t}{2}\right)^{-\frac{\nu+|\mathbf{u}|}{2}} \int \left(\frac{t}{2\sigma_j^2}\right)^{\frac{\nu+|\mathbf{u}|}{2}-1} \exp\left(-\frac{t}{2\sigma_j^2}\right) \mathrm{d}\left(\frac{t}{2\sigma_j^2}\right) \\ = \; & \left(\frac{t}{2}\right)^{-\frac{\nu+|\mathbf{u}|}{2}} \Gamma\left(\frac{\nu+|\mathbf{u}|}{2}\right) \end{aligned} \tag{6.23}$$

Therefore,

$$\begin{aligned} & p(\sigma_j^2 \,|\, \mathcal{D}[\mathbf{u}, j], \mu_j, \nu, s^2) \\ = \; & \frac{\left(\frac{\sum_{i\in\mathbf{u}}(\mathcal{D}[i,j]-\mu_j)^2+\nu s^2}{2}\right)^{\frac{\nu+|\mathbf{u}|}{2}}}{\Gamma\left(\frac{\nu+|\mathbf{u}|}{2}\right)} \cdot \sigma_j^{-2(\frac{\nu+|\mathbf{u}|}{2}+1)} \exp\left\{-\frac{\sum_{i\in\mathbf{u}}(\mathcal{D}[i,j]-\mu_j)^2 + \nu s^2}{2\sigma_j^2}\right\}. \end{aligned} \tag{6.24}$$

Using

$$\hat{\nu} = \nu + |\mathbf{u}| \tag{6.25}$$

and

$$\hat{s}^2 = \frac{1}{\nu + |\mathbf{u}|} \sum_{i \in \mathbf{u}} (\mathcal{D}[i, j] - \mu_j)^2 + \frac{\nu}{\nu + |\mathbf{u}|} s^2, \tag{6.26}$$

we have,

$$p(\mathcal{D}[\mathbf{u}, j] | \mu_j, \nu, s^2) = \frac{\left(\frac{\hat{\nu}}{2}\right)^{\frac{\hat{\nu}}{2}}}{\Gamma\left(\frac{\hat{\nu}}{2}\right)} \hat{s}^{\hat{\nu}} \sigma_j^{-2\left(\frac{\hat{\nu}}{2}+1\right)} \exp\left(-\frac{\hat{\nu} \cdot \hat{s}^2}{2\sigma_j^2}\right). \tag{6.27}$$

That is to say,

$$p(\sigma_j^2 \,|\, \mathcal{D}, \mathcal{D}_R, \mathbf{C}_q, \mu_j, \nu, s^2) = \text{Inverse-}\chi^2(\hat{\nu}, \hat{s}^2). \tag{6.28}$$

Thanks to the conditional independence assumptions of our model as described in Section 6.2, in Equations 6.17 to 6.28, 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 $\mathbf{C}_q$,

$$\mathbf{u} = \mathbf{r} \quad \forall\, \mu_j^{\text{bcl}} \text{ and } (\sigma^2)_j^{\text{bcl}},\ j \in \mathbf{e}; \tag{6.29}$$

$$\mathbf{u} = \begin{cases} \bar{\mathbf{r}} & j \in \mathbf{e} \\ [1 \ldots n]^{\mathrm{T}} & j \in \bar{\mathbf{e}} \end{cases} \quad \forall\, \mu_j^{bgd} \text{ and } (\sigma^2)_j^{\text{bgd}},\ j = 1 \ldots m. \tag{6.30}$$

Equation 4.46 is directly applicable for the evaluation of the full conditional distribution of $\mathcal{D}_R[i]$ ($i = 1 \ldots n$), which now becomes,

$$\gamma_i^{\mathrm{r}} = \prod_{j \in \mathbf{e}} \frac{P\left(\mathcal{D}[i, j] \,|\, \mu_j^{\text{bcl}}, (\sigma^2)_i^{\text{bcl}}\right)}{P\left(\mathcal{D}[i, j] \,|\, \mu_j^{\text{bgd}}, (\sigma^2)_j^{\text{bgd}}\right)} \cdot \frac{v_{\bar{\imath}} + \zeta_1^{\mathrm{r}}}{n - 1 - v_{\bar{\imath}} + \zeta_0^{\mathrm{r}}} \quad i = 1, \ldots, n. \tag{6.31}$$

When applying Equation 4.49 to our model here, we have to keep in mind as Equation 6.30 indicates that μ_j^{bgd} and $(\sigma^2)_j^{\text{bgd}}$ 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 4.49 is

evaluated as

$$
\begin{aligned}
\gamma_k^{\mathrm{c}} &= \frac{p(\mathcal{D}, \mid \mathcal{D}_R, C_k = 1, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}})}{p(\mathcal{D}, \mid \mathcal{D}_R, C_j = 0, \mathbf{C}_{\bar{j}}, \boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}})} \\
&\quad \cdot \frac{p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \mid C_k = 1, \mathbf{C}_{\bar{j}}, \boldsymbol{\xi}^{\mathrm{bcl}}, \boldsymbol{\xi}^{\mathrm{bgd}})}{p(\boldsymbol{\Theta}^{\mathrm{bcl}}, \boldsymbol{\Theta}^{\mathrm{bgd}} \mid C_k = 0, \mathbf{C}_{\bar{j}}, \boldsymbol{\xi}^{\mathrm{bcl}}, \boldsymbol{\xi}^{\mathrm{bgd}})} \cdot \frac{w_{\bar{j}} + \zeta_1^{\mathrm{c}}}{m - w_{\bar{j}} + \zeta_0^{\mathrm{c}}} \\
&= \prod_{j \in \mathbf{e}_k} \left\{ \frac{p\left(\mathcal{D}[\mathbf{r}, j] \mid \mu_j^{\mathrm{bcl}}, (\sigma^2)_j^{\mathrm{bcl}}\right) \cdot p\left(\mathcal{D}[\bar{\mathbf{r}}, j] \mid \mu_j^{\mathrm{bgd1}}, (\sigma^2)_j^{\mathrm{bgd1}}\right)}{p\left(\mathcal{D}[\cdot, j] \mid \mu_j^{\mathrm{bgd0}}, (\sigma^2)_j^{\mathrm{bgd0}}\right)} \right. \\
&\quad \cdot \frac{p\left(\mu_j^{\mathrm{bcl}} \mid \phi_j^{\mathrm{bcl}}, (\tau^2)_j^{\mathrm{bcl}}\right) \cdot p\left(\mu_j^{\mathrm{bgd1}} \mid \phi_j^{\mathrm{bgd}}, (\tau^2)_j^{\mathrm{bgd}}\right)}{p\left(\mu_j^{\mathrm{bgd0}} \mid \phi_j^{\mathrm{bgd}}, (\tau^2)_j^{\mathrm{bgd}}\right)} \\
&\quad \left. \cdot \frac{p\left((\sigma^2)_j^{\mathrm{bcl}} \mid \nu^{\mathrm{bcl}}, (s^2)^{\mathrm{bcl}}\right) \cdot p\left((\sigma^2)_j^{\mathrm{bgd1}} \mid \nu^{\mathrm{bgd}}, (s^2)^{\mathrm{bgd}}\right)}{p\left((\sigma^2)_j^{\mathrm{bgd0}} \mid \nu^{\mathrm{bgd}}, (s^2)^{\mathrm{bgd}}\right)} \right\} \\
&\quad \cdot \frac{w_{\bar{j}} + \zeta_1^{\mathrm{c}}}{m - w_{\bar{j}} + \zeta_0^{\mathrm{c}}}
\end{aligned}
\tag{6.32}
$$

The first term in both Equation 6.31 and Equation 6.32 are likelihood ratios (also see the explanations for Equation 4.46 and Equation 4.49). 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.

6.4 Construction of the priors

Note that conjugate priors are used for the hierarchical models (see Section 6.2) 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, and 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

$$\varphi_j^{\text{bcl}} = \frac{\sum_{i \in \mathbf{a}} \mathcal{D}[i, j]}{|\mathbf{a}|}, \tag{6.33}$$

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; and we use a very small value for τ^{bcl}, for example,

$$\tau_j^{\text{bcl}} = 10^{-4},\ k \in \mathbf{e}. \tag{6.34}$$

By setting

$$(s^2)^{\text{bcl}} = \frac{1}{\nu^{\text{bcl}}} \tag{6.35}$$

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 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 μ^{bgd}, we set φ_j^{bgd} to the mean of the expression levels of all the genes under experiment j,

$$\varphi_j^{\text{bcl}} = \frac{\sum_{i \in \bar{\mathbf{a}}} \mathcal{D}[i, j]}{n - |\mathbf{a}|}; \tag{6.36}$$

and we use

$$\tau_j^{\text{bgd}} = 10^{-4},\ k \in \mathbf{e}. \tag{6.37}$$

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

$$(s^2)^{\text{bgd}} = \frac{1}{\nu^{\text{bgd}}}. \tag{6.38}$$

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

$$\zeta_0^{\text{r}} = \zeta_1^{\text{r}} = 0.5 \tag{6.39}$$

$$\zeta_0^{\text{c}} = \zeta_1^{\text{c}} = 0.5. \tag{6.40}$$

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

6.5 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 profile under the working conditions of the regulators. The set of the genes that are co-regulated, the conditions under which the co-regulation occurs and the expression profiles of these genes under the specific set of conditions, as a whole is referred to as a transcriptional module. 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.

Depends on the combination of regulators, same genes can be involved in different transcriptional modules of different size (i.e., biclusters including different addtional genes and different working conditions). A hypothetical example is given in Figure 6.2.

A gene expression dataset 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 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 tested set of conditions. 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 6.2).

6.6 Experiments on synthetic data

To demonstrate the performance of the algorithm, we 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 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 the 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 the resulting data set is illustrated in Figure 6.3 which includes 100 rows from Bicluster 1, 150 rows from Bicluster 2 (note that all the 50 rows at the overlapping region are included as well) and 100 rows from the background data are selected for illustrate. The rows and columns in Figure 6.3 are rearranged so that the embedded biclusters are manifested.

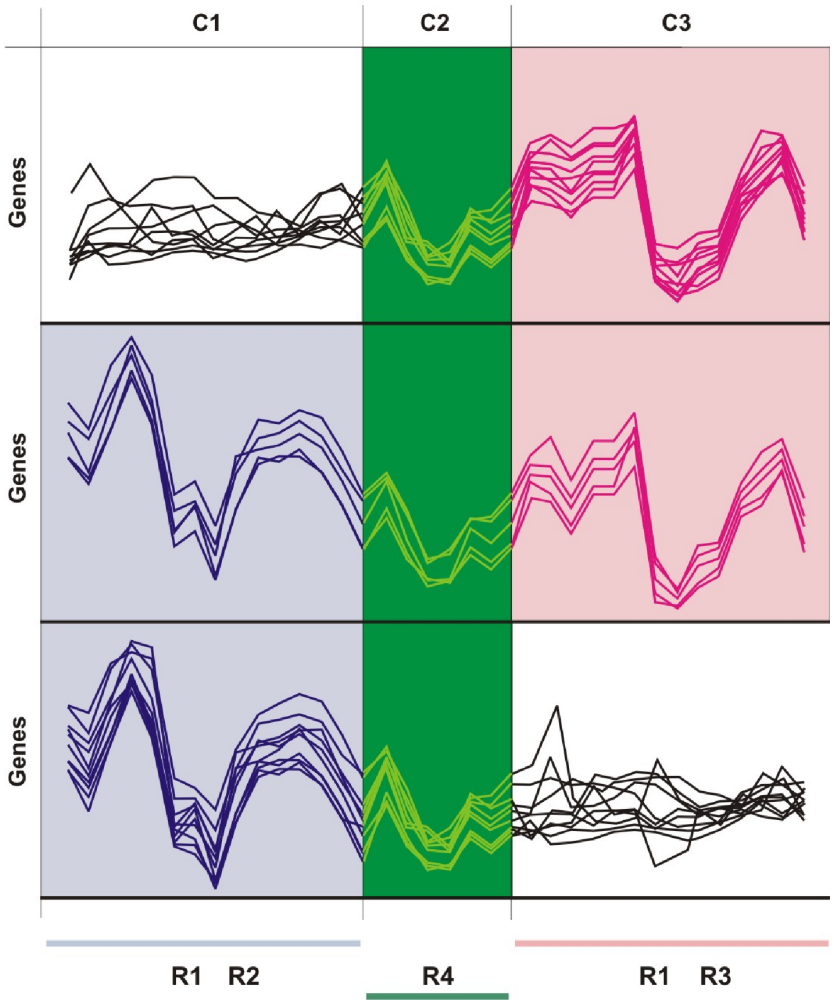


Figure 6.2: C1, C2, and C3 represent three unrelated conditions. Ri represents different regulators active in the respective condition dependent regulatory programs. R1 and R2 are active in C1; R1and R3 are active in C3; and R4 is active in C2. (A) Distinct partially overlapping modules exist. Modules 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.

We want to investigate which part of the three biclusters are recovered under different parameter settings.

In this way, the synthetic data resembles the one that we discussed for Figure 6.2, and yet presents different structures that allow us to explore the influence of the user input parameters ν^{bcl} and ν^{bgd} on the resulting biclusters.

To illustrate the performance of the algorithm under different parameter settings ν^{bcl} and ν^{bgd}, we use four sets of seed rows which are extracted from different part 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 Bicluster 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 our algorithm able to retrieve reproducible patterns for each set of seeds under each of the parameter settings that we tested for this set of synthetic data. We performed the algorithm 10 times on each of Seed 1, 2, 3, and 4, under each parameter setting. For each of Seed 1, 2, 3, and 4, we generate a new set of rand seeds according to the requirement that we described above at each of the 10 rounds. 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 increment in ν^{bgd} raises the stringency in selecting the columns for the bicluster. With a larger ν^{bgd}, fewer columns are selected. Increment 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) are more frequently found with a larger ν^{bcl}. However, comparing biclusters containing the same amount of rows but discovered different ν^{bcl} for the same see, we suppose that a larger ν^{bcl} reinforces the power of the seeds, and consequently relativey 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 is

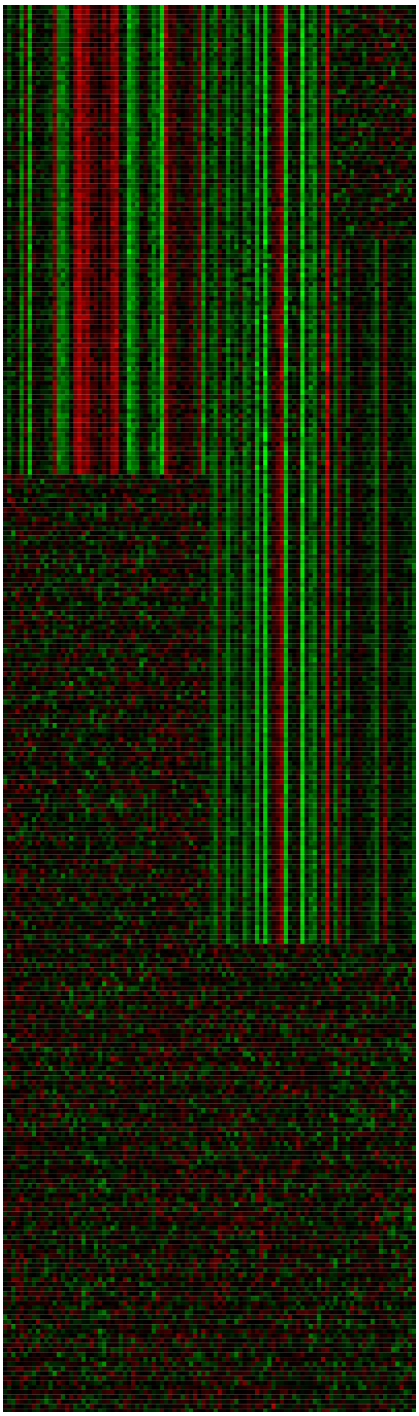


Figure 6.3: Part of the synthetic continuous data set. The values of the data points are reflected by the color scale. The rows and the columns are rearranged to manifest the embedded biclusters.

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 the size such as 4 by 4.

[c]The algorithm failed to converge after 50 (re-initializations), 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).

has a weaker pattern comparing 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. 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 green area in Figure 6.2, 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 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 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 direction of how to manipulate the findings from the input of the algorithm—the seeds, and the parameters ν^{bcl} and ν^{bgd}.

6.7 Transcriptional module discovery on *Saccharomyces cerevisiae*

To discovery a transcriptional regulatory module, we use the method of De Bie *et al*. (2005) [13] to reveal regulatory programs as well as sets of seed genes that are governed these regulatory programs. De Bie *et al*. (2005) [13] combines 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 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 Kellis *et al.* (2003) [51] (motif data), Lee *et al.* (2002) [58] (ChIP chip data), and Spellman *et al.* (1998) [84] (micorarray data). 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 and Table 6.5. 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 6.6.

We applied our algorithm to the combined data set on *Saccharomyces cerevisiae* from Gasch *et al.* (2000) [33] (with stress-response experiments), Spellman *et al.* (1998) [84] and Cho *et al.* (1998) [20] (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 of each other. We expect that for Seed 1 and Seed 2 the the algorithm would identify all the experimental conditions under the data from Spellman *et al.* (1998) [84] and Cho *et al.* (1998) [20], 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, and 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 6.4, Figure 6.5, and Figure 6.6 illustrate the three biclusters we found for Seed1, Seed2, and Seed3. Finally, we validated the bicluster by calculating the functional enrichment of the bicluster using a hypergeometric distribution [87], where the functional categories of the genes are obtained from MIPS [64]. In Table 6.7 to 6.9 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 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 6.10).

Regulators	Gene ID	Names	MIPS	Description
	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
Mbp1			10.03.01	mitotic cell cycle and cell cycle control
Swi4			10.03.02	meiosis
Swi6	YGR221C	TOS2	40.01	cell growth / morphogenesis
Stb1			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
	YLR103C	CDC45	10.01.03	DNA synthesis and replication
			10.03.01	mitotic cell cycle and cell cycle control
	YML027W	YOX1	11	TRANSCRIPTION
	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 6.4: 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 category of the corresponding gene, and the last column describes the functional category lists the regulators of these genes.

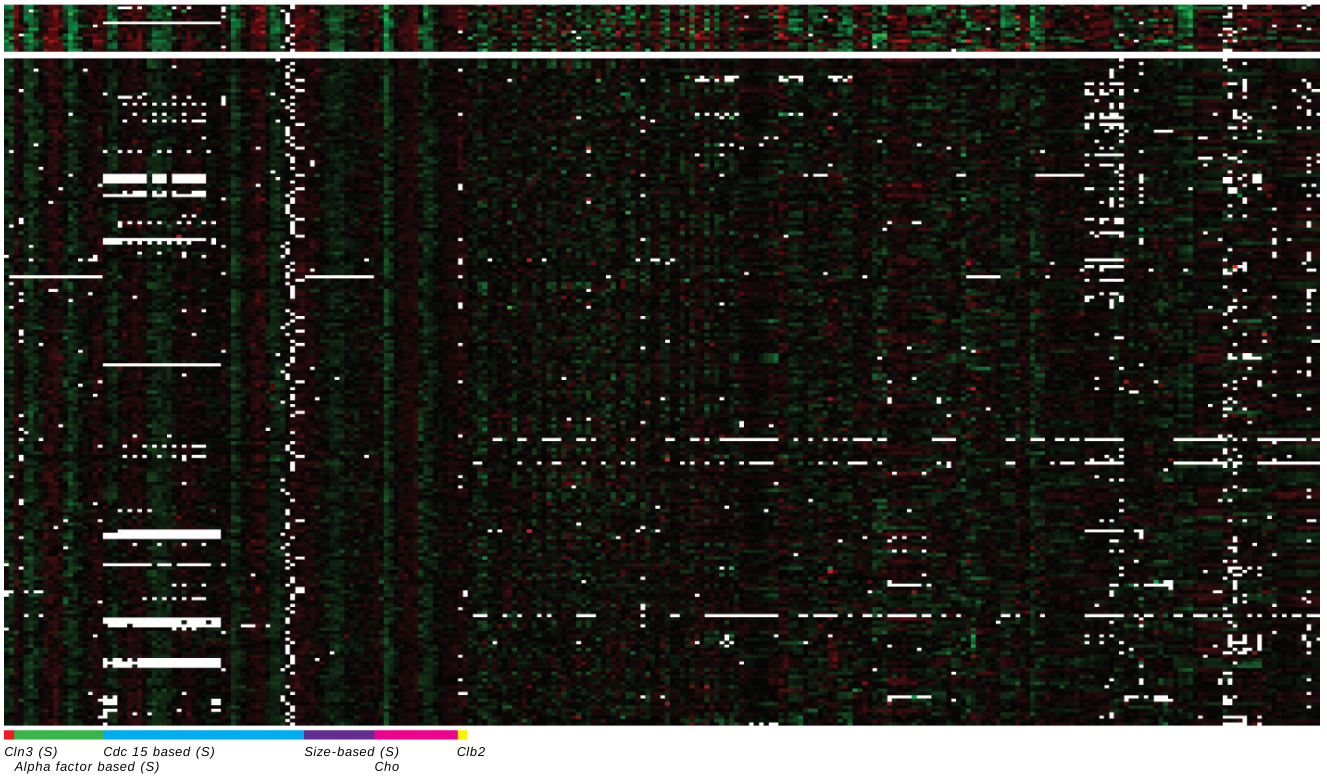


Figure 6.4: Result obtained for Seed 1: The top heat map represents the expression profiles of the seed genes. The bottom heat-map of the expression profiles of the selected genes under all the experimental conditions. Gene expression profiles are represented by rows, and experimental conditions by columns. The conditions are rearranged according to their posterior probabilities to belong to the bicluster in a descending order. The Conditions labeled by color tags are selected for the bicluster. The white spots in the heat map represent missing values in the data.

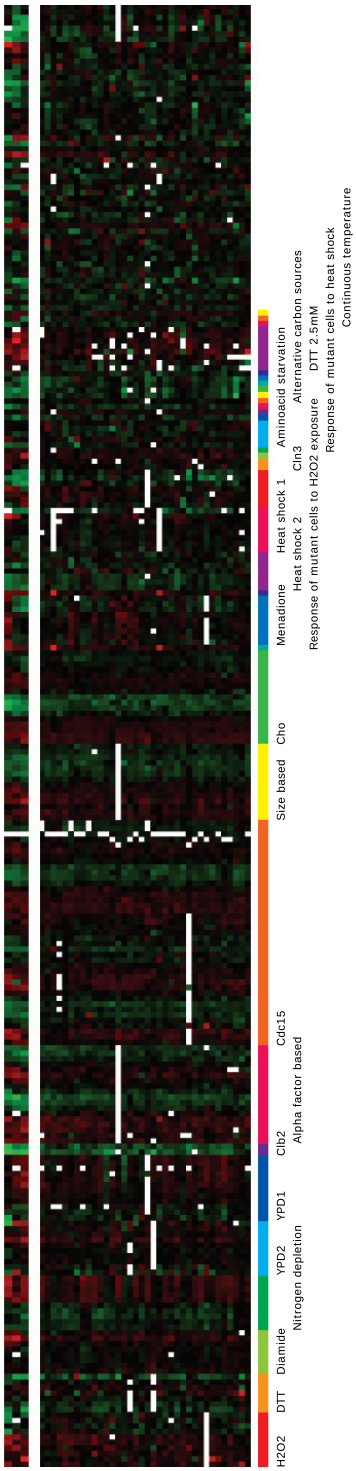


Figure 6.5: Result obtained for Seed 2: The top heat map represents the expression profiles of the seed genes. The bottom heat-map of the expression profiles of the selected genes under all the experimental conditions. Gene expression profiles are represented by rows, and experimental conditions by columns. The conditions are rearranged according to their posterior probabilities to belong to the bicluster in a descending order. The Conditions labeled by color tags are selected for the bicluster. The white spots in the heat map represent missing values in the data.

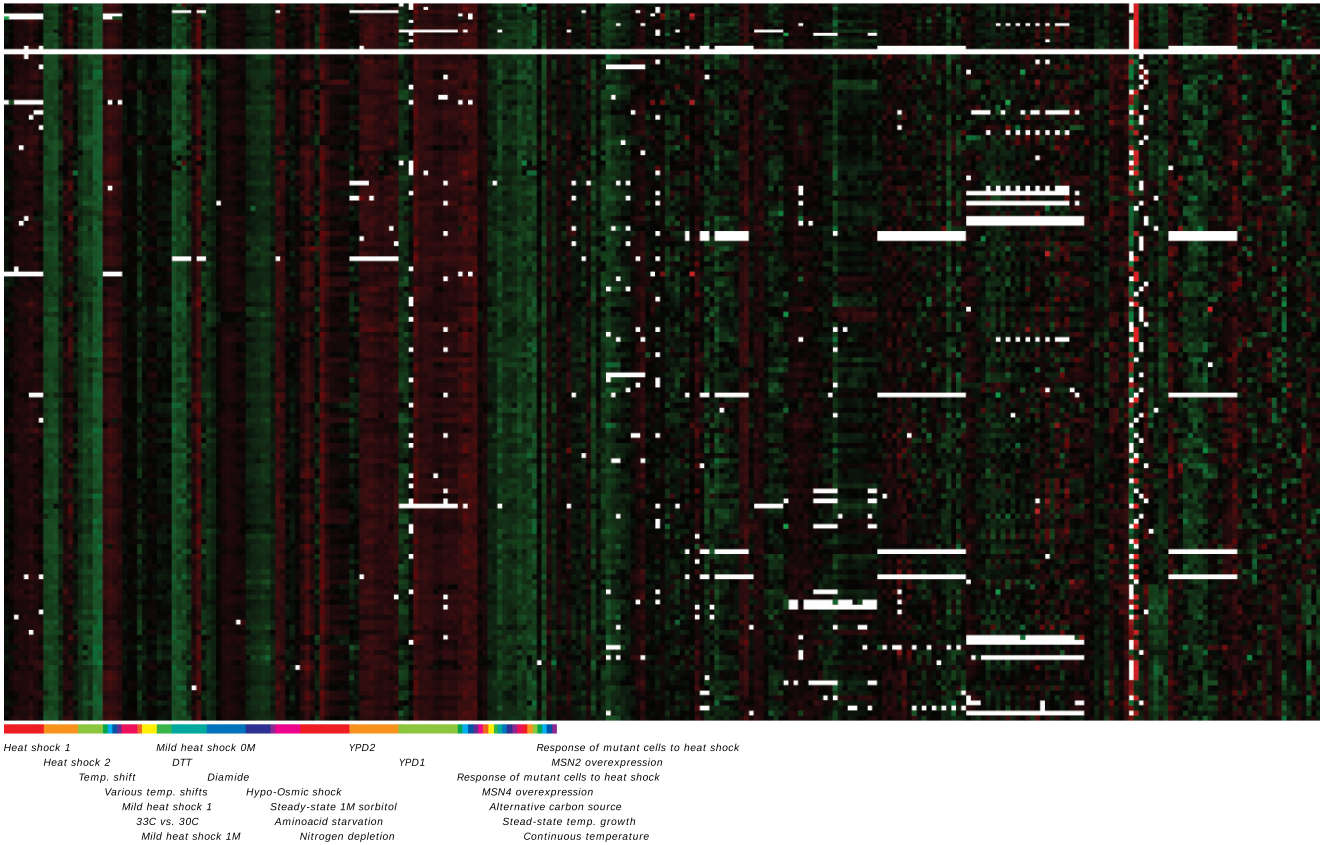


Figure 6.6: Result obtained for Seed 3: The top heat map represents the expression profiles of the seed genes. The bottom heat-map of the expression profiles of the selected genes under all the experimental conditions. Gene expression profiles are represented by rows, and experimental conditions by columns. The conditions are rearranged according to their posterior probabilities to belong to the bicluster in a descending order. The Conditions labeled by color tags are selected for the bicluster. The white spots in the heat map represent missing values in the data.

Table 6.5: 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 6.6: Information about Seed 3.

Regulators	Gene ID	Names	MIPS	Description
Fkl1 Yap5 Rap1	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 14.13.01	ribosome biogenesis 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 34.11.03.07	ribosome biogenesis pheromone response, mating-type determination, sex-specific proteins

Although Seed 3 is obtained by applying the method of [13] to the data set from [84], the cell cycle related experimental conditions are seldom selected to be in the bicluster, while almost all the stress response related conditions from [33] are selected. This result shows that data set from [33] might be a better data set to look at for the study of "ribosome biogenesis" than those from [84] and [20], justified by either some biological explanation or the experimental noise presented 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 6.7: 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 6.8: 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.
		12	PROTEIN SYNTHESIS	119	2.4e-13		
130	1	**12.01**	ribosome biogenesis	112	1.7e-11	YLR333C	most of the
		12.04	translation	7	4.9e-04		stress conditions

Table 6.9: 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 6.10: Function (according to SGD) of the genes that are found for the bicluster applying Seed3, and are not associated to "protein synthesis" according to MIPS.

Chapter 7

Discussions and conclusions

In this thesis, we have developed a general framework based on Bayesian hierarchical models and Gibbs sampling for solving the biclustering problems for microarray data. We have also equipped our methodology with two dedicated models to provide solutions to the problems of both the biclustering experiments and the biclusteirng of genes, which are two distinct problems because of the asymmetry of microarray data.

In this chapter, we conclude 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 researches in this topic.

7.1 Achievements of the work

Our choice of strategy for the biclusering problems provides our algorithms 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 biclusteirng 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 base 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 discovery gene regulatory modules in the study of systems biology. First, useful information from other data sources (in this case, the ChIPchip data and motive information) are extracted (by the method of De Bie *et al*. (2005) [13]) 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 Gaussian-Wishart prior model that is compatible with our Bayesian hierarchical model for the biclustering 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 bicluster toward the finding of gene expressional finger print of subtle traits.

Robust results

Because of the complicated biological highly 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 critic issue for choosing the optimization methods for the probabilistic model. (Structural) EM is one of the popular method for solving the missing data problem that concerns the biclustering problem. However, EM is well known for its frequent encounter with local maxima solutions. Taking into account that it is never obvious in advance of how many runs of an EM procedure can guarantee the discovery of the global mode, we opt for the Gibbs sampling procedure for the parameterization of the Bayesian model. Though it takes longer for the algorithm to converge, Gibbs sampling provides much higher chance in 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 addion, 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. An (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 allow 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 which 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 of strategy, 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, considering their success in detecting differentially expressed genes.

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 knowledges 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 Gaussian-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.

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, and M. C. Costanzo. Saccharomyces genome database, July 2005.

[7] P. Baldi and S. Brunak. *Bioinformatics: the Machine Learning Approach*. Adaptive computation and machine learning. The MIT Press, second edition 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] Z. Bar-Joseph, D. K. Gifford, and S. Jaakkola. Fast optimal leaf ordering for hierachical clustering. *Bioinformatics*, 17(Suppl. 1):S22–S29, 2001.

[10] Y. Barash and N. Friedman. Context-specific bayesian clustering for gene expression data. *J. Comput. Biol.*, 9:169–191, 2002.

[11] 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.

[12] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *J. Comput. Biol.*, 6:281–297, 1999.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] G. Casella and E. I. George. Explaining the Gibbs sampler. *Am. Stat.*, 46(3):167–174, 1992.

[18] 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.

[19] Y. Cheng and G. M. Church. Biclustering of expression data. In *ISMB 2000 proceedings*, pages 93–103, 2000.

[20] 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.

[21] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *J. Amer. Stat. Assoc.*, 32:829–836, 1979.

[22] The Gene Ontoloy Consortium. Gene Ontology: tool for the unification of biology. *Nature Genet.*, 25:25–29, 2000.

[23] M. K. Cowles and B. Carlin. Markov chain Monte Carlo convergence diagnositics a comparative review. *J. Amer. Stat. Assoc.*, 91:883–904, 1996.

[24] 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.

[25] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Willey & Sons, Inc., second edition edition, 2001.

[26] 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.

[27] B. P. Durbin and D. M. Rocke. Variance-stabilizing transformations for two-color microarrays. *Bioinformatics*, 20(5):660–667, 2004.

[28] 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.

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

[30] N. Friedman. The bayesian structural EM algorithm. In *Proceedings of UAI'98*, 1998.

[31] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303:799–805, 2004.

[32] N. Friedman, L. Gettor, D. Koller, and A. Preffer. Learning probabilistic relational models. In *Proceedings of IJCAI'99*, 1999.

[33] 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.

[34] A. E. Gelfand and F. M. Smith. Sampling-based approaches to calculating maginal densities. *J. Amer. Stat. Assoc.*, 85:398–409, 1990.

[35] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–472, 1992.

[36] 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.

[37] 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.

[38] 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.

[39] D. Ghosh and A. M. Chinnaiyan. Mixture modelling of gene expression data from microarray experiments. *Bioinformatics*, 18(2):275–286, 2002.

[40] J. A. Hartigan. *Clustering Algorithms*. Wiley Series i Probability. John Wiley & Sons, Inc., 1975.

[41] 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.

[42] W. K. Hasting. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 87:97–109, 1970.

[43] 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.

[44] J. Herrero, A. Valencia, and J. Dopazo. A hierachical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, 17(2):126–136, 2001.

[45] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, 9:1106–1115, 1999.

[46] 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.

[47] 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.

[48] M. I. Jorda, editor. *Learning in Graphical models*. The MIT Press, 1999.

[49] R. Jörsten and Bin Yu. Simultaneous gene clustering and subset selection for sample classification via MDL. *Bioinformatics*, 19(9):1100–1109, 2003.

[50] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: an Introcuction to Cluster Analysis*. John Wiley & Sons, Inc., 1990.

[51] 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.

[52] 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.

[53] M. K. Kerr, M. Martin, and G. A. Churchill. Analysis of variance for gene expression microarray data. *J. Comput. Biol.*, 7:819–837, 2000.

[54] 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.

[55] T. Kohonen. *Self-Organizing Maps*. Springer Series in Information Sciences. Springer, 1995.

[56] 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.

[57] L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Statistica Sinica*, 12:61–86, 2002.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] J. Quackenbush. Computational analysis of microarray data. *Nature Reviews*, 2:418–427, 2001.

[70] J. Quackenbush. Microarray data normalization and transformation. *Nat. Genet.*, 32:suppl. 496–501, 2002.

[71] 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.

[72] 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.

[73] D. M. Rocke and B. Durbin. A model for measurement error for gene expression arrays. *J. Comput. Biol.*, 8:557–569, 2001.

[74] D. M. Rocke and B. Durbin. Approximate variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics*, 19(8):966–972, 2003.

[75] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[76] 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.

[77] 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.

[78] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller. Rich probabilisitc models for gene expression. *Bioinformatics*, 17(Suppl. 1):S243–S252, 2001.

[79] E. Segal, H. Wang, and D. Koller. Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, 19:i264–i272, 2003.

[80] Q. Sheng, Y. Moreau, and B. De Moor. Biclustering microarray data by Gibbs sampling. *Bioinformatics*, 19(Suppl. 2):II196–II205, 2003.

[81] G. Sherlock. Analysis of large-scale gene expression data. *Curr. Opin. Immunol.*, 12:201–205, 2000.

[82] 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.

[83] 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.

[84] 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 *s*accaromyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9:3273–3297, 1998.

[85] J. D. Storey and R. Tibshirani. Statistical significance for genomewide studies. *Proc. Natl. Acad. Sci. USA*, 100(10):9440–9445, 2003.

[86] 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.

[87] 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.

[88] 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.

[89] 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.

[90] P. Törönen, M. Kolehmainen, G. Wong, and E. Gastrén. Analysis of gene expression data using self-organizing maps. *FEBS Letters*, 451:142–146, 1999.

[91] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles*. Applied mathematics and computaion. Addison-Wesley Publishing Company, 1979.

[92] 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.

[93] 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.

[94] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[95] 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.

[96] J. H. Ward. Hierarchical grouping to optimize an objective function. *Jour. Amer. Stat. Assoc.*, 58:239–244, 1963.

[97] B. L. Wiens. When log-normal and gamma models give different results: a case study. *Am. Stat.*, 53:89–93, 1999.

[98] 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.

[99] Y. H. Yang, M. J. Buckley, and T. P. Speed. Analysis of cDNA microarray images. *Briefings in Bioinformatics*, 2:341–349, 2001.

[100] 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.

[101] 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.

[102] 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.

[103] 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.

[104] K.Y. Yeung, D.R. Haynor, and W.L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318, 2001.

Index

in situ synthesized array, 8
    Affymetrix GeneChip, 8

Bayesian network, 48
BIC, 30

cDNA, 7

EM, 42, 49
EST, 8

graphical models, 47
    edge, 47
    vertex, 47

MA-plot, 16
MAP, 50

normalization, 16
    lowess normalization, 16
    quantile normalization, 18
    RMA, 15
    VSN, 18

oligonucleotide, 8

probe, 7
    probe set, 9
        IM, 15
        MM, 11
        PM, 11

spotted array, 8
    cDNA array, 8

two-channel microarray, 12

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 (Shanghia 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 Katholieke Universiteit Leuven. 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.