Mixture Models for the Analysis of Gene Expression: Integration of Multiple Experiments and Cluster Validation

Ivan Gesteira Costa Filho

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

Gutachter:
Prof. Dr. Martin Vingron
Prof. Dr. Joachim Selbig

1. Referent: Prof. Dr. Martin Vingron
2. Referent: Prof. Dr. Joachim Selbig
Tag der Promotion: 29 Mai 2008

Contents

List of Figures

List of Tables

Preface

Acknowledgments

First of all, I would like to thank Dr. Alexander Schliep for his supervision and support during my Ph.D. studies. He introduced me to the field of mixture models, motivated me to do state-of-the-art bioinformatics research, and was a valuable source of research ideas. Some of such ideas that I could turn into reality are the main parts of this thesis.

I am grateful to Dr. Stefan Roepcke for introducing me to the Lymphoid cells research area and for great collaborative work. And more importantly, for keeping me aware of the other side (biology!). I would also acknowledge Prof. Fritz Melchers for his inspiring "private lectures" on B (and T) cells development, and encouragements on my work on Lymphoid development. I thank Dr. Alexander Schoenhut for his work on topology learning and Viterbi decomposition for HMMs and several discussions on the analysis of gene expression time courses. I am also in debt to Dr. Roland Krause for his valuable opinions, as well as for his work on the analysis of Drosophila development data. I acknowledge the aid of Lennart Optiz in the in-situ image pre-processing. I thank Dr. Marcilio de Souto for his advices and great help with the thesis corrections. Furthermore, we carried out a collaboration work on clustering of cancer gene expression data, together with several Brazilian colleagues: Prof. Teresa Ludemir, Dr. Francisco de Carvalho, Dr. Ricardo Prudncio, Daniel Arajo e Rodrigo Soares. I am also grateful to Prof. Joachim Selbig for participating in my Ph.D. thesis committee and for his valuable remarks.

It was a great experience to be a member of the Algorithms group. The group was a perfect environment for scientific discussions and for the collaborative development of "open source" software libraries (e.g., ghmm, GQL, GATO and pymix). The latter was particularly important for development of this thesis. In special, I would like to thank Christopher Hafemeister for his great work in the software of mixture of dependence trees; Benjamin Georgi for many interesting discussions; Jane Grunau for his prompt support on the GHMM library; and Rubens Schilling for both his work on GQL and his help with image processing analysis.

I owe most of my knowledge on computational biology to the Department of Computational Molecular Biology (CMB) and its members. The interaction with the department members through seminars, meetings and retreats was of great importance to my development as a computational biologist. In particular, I would like to thank Dr. Steffen Grossman for discussions on Gene Ontology, cluster validation and for sharing the office; Helge Roider for discussion on gene regulation and his help with the promoter analysis of Lymphoid development; Hughes Richards for several discussions; Willy for prompt answers on technical problems; and Utz Pape for his aid with this thesis corrections. Finally, I am very thankful to Prof. Martin Vingron for being my supervisor and giving me the opportunity to be a member of the department.

This work would not have been possible without the support, friendship and love from special friends and family members: Carlos Venezuela, Fernandão, Gui, Hugo, Jorgito, Julia, Lauro Mostrinho, Lili, Lucciano, Marcilio, Paulete, Philipão, Reginho, Sandra, Scylla and Stefan. Thanks! I would also like to thank Gabi for all love and companionship on earlier years of this work. Last but not least, I am very grateful to my parents: my father, for being my role model of wisdom, which pursue lead to my curiosity for knowledge, which consequently brought me to science; my mother for inciting and fostering my need for knowledge and for their unconditional love. I dedicate this thesis for them.

Publications

Parts of this thesis have been previously published. Chapter 3 includes results of a paper in the Annual Conference of the German Classification Society 2005 [51]. Also, parts of the results in Chapter 4 were published in the journals IEEE Transactions of Bioinformatics and Computational Biology [185] and Bioinformatics [53]. Chapter 5 includes results presented at the PLoS Track of the International Conference on Intelligent Systems for Molecular Biology 2006 and published in the journal BMC Immunology [50] and results accepted for publication in the International Conference on Intelligent Systems for Molecular Biology 2008 [49]. Chapter 6 contains results presented at the NIPS Workshop on New Problems and Methods in Computational Biology 2005, published in the ECML Workshop of Data and Text Mining for Integrative Biology 2006 [52], and in the journal BMC Bioinformatics [48]. Some collaborative work during my Ph.D studies, which were not described in this thesis, also lead to publications on the topics of cluster validation [47, 59, 60] and semi-supervised learning [189].

Chapter 1

Introduction

We are concerned with the use of computational and statistical methods for the analysis of gene expression data. In this chapter, we describe some basic concepts in gene expression and biotechnological methods used to measure gene expression in large-scale experiments. Additionally, we give a brief overview of the main tasks and challenges in the analysis of the resulting data. Finally, we outline the main scientific contributions of this thesis and summarize its contents.

1.1 Gene Expression: Transcription, Translation and Control

First, we briefly review the process of gene expression. A detailed description can be found in many textbooks, see for example [4]. The genetic information of organisms is stored in deoxyribonucleic acid (DNA) molecules. These molecules are composed of two polynucleotide chains (or strands) forming the double helix structure (Figure 1.1). The nucleotides, which are the building blocks of a DNA molecule, are characterized by the base attached to a sugar phosphate. The classical four types are: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). One particularity of the double stranded DNA is the complementary base pairing, i.e., a particular base on a strand only binds to a complementary base on the opposite strand. More precisely, “A” binds only to “T”, and “C” to “G” (Figure 1.1). The reaction in which a single stranded DNA molecule binds to a complementary strand is called hybridization, a reaction exploited by many molecular biology techniques.

In eukaryotes, i.e., organisms which have cellular nucleus, several linear DNA molecules, called chromosomes, are present in the cell nucleus. Each of these chromosomes is formed by billions of base pairs. Genes are regions of the chromosome that code one or more proteins[1]. They represent the basic units responsible for storing and passing on hereditary characteristics.

Gene expression is the process by which the genetic information contained in the genes

[1] Some genes will code functional RNA structures, which are not translated into proteins.

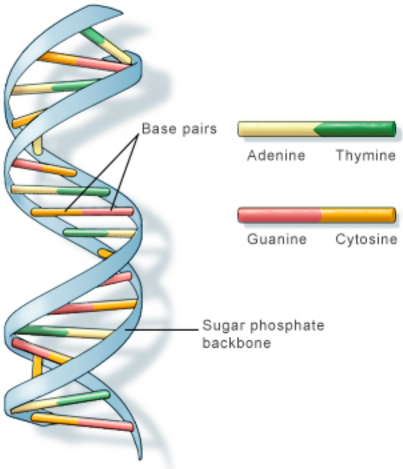


Figure 1.1: *Example of a double stranded DNA molecule. Figure reproduced from the US National Library of Medicine.*

is translated into ribonucleic acid (RNA) molecules and, later, into protein molecules [4]. This process is divided into two main steps: transcription and translation. In the transcription step, regions of DNA, which code genes, are transformed into RNA molecules by the RNA polymerase (Figure 1.2, Step 1). RNA molecules are different from DNA in several aspects: (1) they are only single stranded; (2) they have Uracil (U) instead of Thymine (T); and (3) they have a quicker degradation time.

Next, in the translation process (Figure 1.2, Step 2), RNA molecules leave the nucleus and are "read" by the ribosome in order to synthesize proteins. Triplets of RNA bases are mapped via the genetic code into one of the twenty amino acids. These amino acids are the building blocks of the proteins. Proteins, the final products of genes, are vital to the cell functioning, since they constitute the structural components of the cells and catalyse biochemical reactions.

While most cells of an eukaryotic organism encode the same genetic information, they express genes at distinct levels. The expression of a particular set of genes is either a response to distinct environmental conditions or is part of the specific repertoire of a given cell type. Understanding the mechanisms controlling gene expression is a central question in molecular biology. This control can happen at several levels of the gene expression process. The first level and the one of main concern in this work is the transcriptional control. At this level, proteins, called transcription factors, bind the upstream (or regulatory) regions of genes. These factors act as initiators (or repressors) of transcription by facilitating (or blocking) the access of the RNA polymerase to initiate transcription.

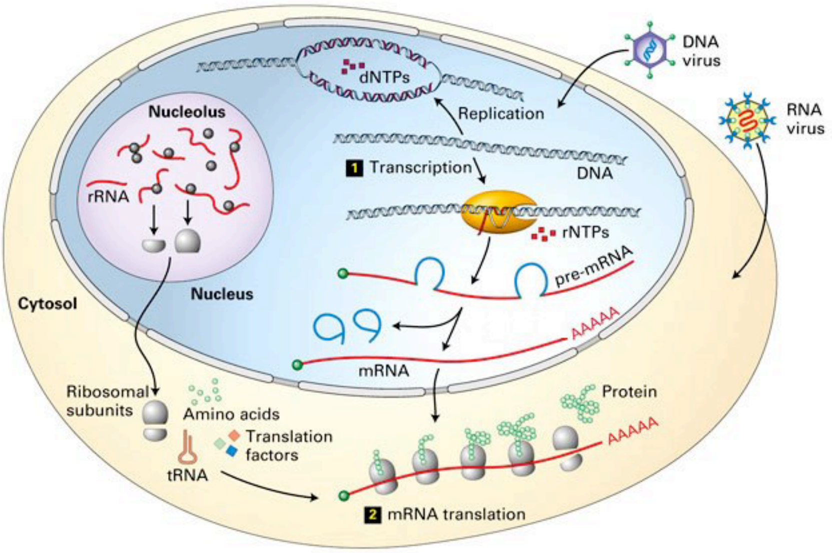


Figure 1.2: *We depict here the main stages of gene expression. Step 1 corresponds to the transcription of DNA to RNA molecules. Step 2 corresponds to the translation of messenger RNA (mRNA) to protein molecules. Figure reproduced from [136].*

1.2 Measuring Gene Expression with Microarrays

Microarray technology allows the simultaneous measurement of the concentrations of RNA molecules (or transcripts). More precisely, this technology allows the measurements of the expression patterns of genes—also known as expression profiles. For example, by comparing the expression profiles of disease and normal cells [5], responses of cells to environmental conditions [82], or during biological process such as cell cycle [201] and development [214], the researchers can explore the dynamics of gene expression, to form hypotheses about regulatory and functional roles of genes, and to obtain molecular signatures of cell types and all this on a genome-wide scale.

Microarray Technology. The main idea behind DNA microarrays is to exploit the fact that two complementary single stranded DNA molecules hybridize [111]. For each gene of interest, a short sequence complementary to its sequence is select. These sequences are called probes and have lengths ranging from 20 to 60 bases. The probes should be selected in such a way that there is a low chance of hybridizing with sequence others than the target gene sequence.

Then, with the aid of robotics or nano manufacture technologies, thousands of copies of a particular probe are placed in a tiny area of a hard surface — the array. Thousands of such probe spots can be placed side by side forming a grid on the array. Each spot contains probes designed to hybridize with RNA from a specific gene. In the end, one can have as many as 10^5 spots arranged in a 2×2 cm array.

In the next step, the RNA molecules of the cell population of interest are separated and transcribed to single stranded complementary DNA (cDNA) molecules. This step is needed as RNA molecules are unstable and would quickly degrade. Afterwards, the cDNA molecules are marked with fluorescent or radioactive labels. The cDNA molecules are, then, poured onto the slide. After some time, the slide is washed, removing the cDNA molecules that did not hybridize with the probes.

Next, the slide is scanned, resulting in an image with all the spots intensities (see Figure 1.3 middle). Such an image is further processed using computational methods. The aim is to calculate the intensity at each spot, which is proportional to the number of transcripts of a gene that a probe is complementary to (the whole process is illustrated in Figure 1.3).

There are several distinct microarrays technologies such as cDNA microarrays [183] and Affymetrix Gene Chips (also known as Oligonucleotide arrays) [134]. They differ mostly by how the chips are manufactured and on methodologies for probe selection. The particular characteristics of such technologies are important for decisions concerning experimental design, experimental costs, measurements reliability and data pre-processing aspects. See for example [119] for a complete description of microarray technologies.

One important aspect of microarrays is the use (or not) of reference RNA samples. In double channel microarrays, such as cDNAs microarrays, two cell samples are poured in the same microarray: cells of interest (e.g., disease cells, treated cells) and reference cells (e.g., healthy cells, untreated cells). Each of these cell populations is dyed with a distinct marker, for instance, a red Cy3 dye versus a green Cy5 dye. Double channel microarrays return a relative quantification of the RNA expression in relation to the reference cell, usually measured by taking the logarithm of the ratio between the red and green signals (see Figure 1.3 (a) for an example of a two-channel microarray).

In single channel arrays, only one RNA sample is poured in the array, and no reference sample is used. Single channel arrays return estimates on the number of copies of a particular transcript in a given sample. With Affymetrix microarrays, an example of single channel array, 20 to 40 distinct probes, which are complementary to the sequence of an unique gene, are placed in distinct spots on the array in order to obtain reliable estimates of RNA quantities. Additionally, a mismatch spot (MM) containing a sequence, where a base in the middle of the original probe (PM) sequence is exchanged, is placed next to each PM spot. These reduce the effect of cross-hybridization, increase the signal to noise ratio and improve the accuracy of the RNA quantification (see Figure 1.3 (b) for an example of an single-channel microarray).

Pre-processing and normalization procedures are the initial computational tasks in the analysis of data from microarrays. These procedures are responsible for improving the estimates of the RNA levels measured by microarrays. In the pre-processing step, one tries to correct the probe intensities for errors introduced by experimental artifacts, such as non-specific hybridization, dye efficiency, spatial biases, and so on. See for example [106]

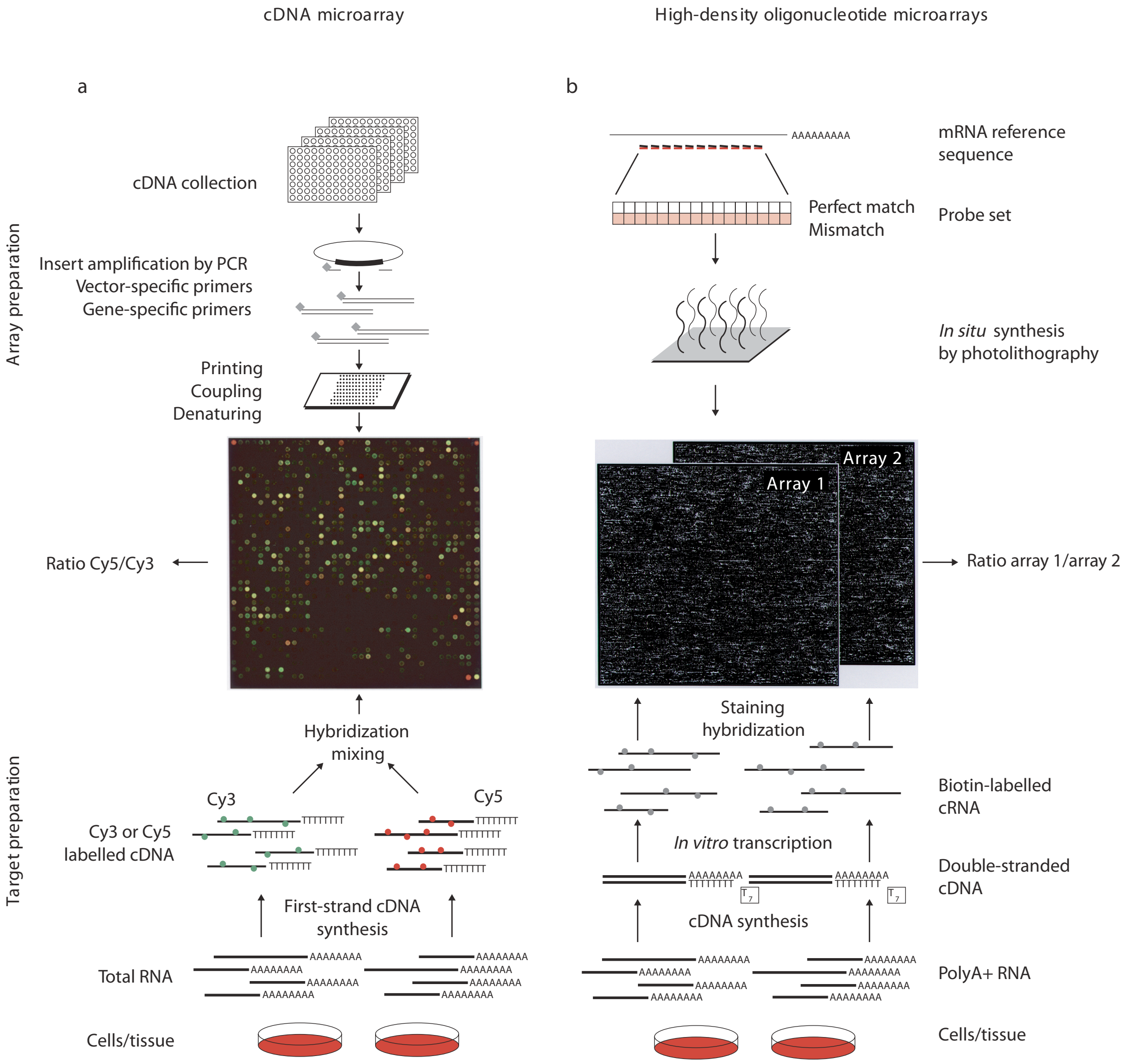


Figure 1.3: *We depict how microarray experiments are performed for cDNA (a) and Oligonucleotide (b) microarrays. In the top, we depict how microarrays are manufactured; and in the bottom, how RNA samples are obtained. In the middle, we can see the images obtained after RNA samples hybridize to the microarrays. For cDNA microarrays (a), each dot represents a probe, and the red (or green) colors are proportional to the counts of RNA hybridized to that probe in the reference (or control) sample. Similarly, the intensity of white dots in Oligonucleotide arrays (b) represents the counts of RNA hybridized to that probe. Figure reproduced from [190].*

for a review of methods on Affymetrix Gene Chips or [230] for protocols for cDNA microarrays. Next, normalization methods are applied with the aim of making expression values obtained in distinct hybridization experiments comparable. See [203] for a review of pre-processing and normalization methods.

Computational Challenges in the Analysis of Gene Expression data. Large-scale data produced by microarrays experiments shows how gene expression changes in distinct biological conditions and tissue types. Manual analysis of such amount of data is not feasible. Due to this limitation, statistical and computational methods are vital for analyzing gene expression data. In fact, data arising from microarrays have several particularities that should be taken into consideration by these methods: they should be able to cope with the high dimensionality of the data, be robust to noise and take advantage of the experimental design associated with the biological experiment.

For example, gene expression levels are often measured in few experimental conditions, i.e., tissues types or time points (less the 100) for thousands of genes (more the 10,000). Furthermore, despite improvements of microarrays experiments and protocols, these technologies still suffer from several sources of noise: either by manufacturing failures, problems in the reading procedure, unspecific probes, variability in biological samples, or variations in the environment conditions in which experiments are performed. A recent study [107] showed that at least 10% of expression measurements differ significantly in replication experiments.

Another important aspect is the experimental design procedure used for acquiring the data. For instance, in microarrays experiments measured over time, such as during cell cycle, the cell populations tend to desynchronize with time. This results in deterioration of the expression measurements of later time points [201]. The explicit use of knowledge of the biological process makes computational and statistical methods more robust to this type of inherent noise.

1.3 Thesis Overview

In this thesis, the main focus is on the problem of finding groups of co-expressed genes, or genes that display the same expression behavior through particular biological conditions, such as cell cycle, or developmental processes. The basic rational underlying this approach is the assumption that co-expressed genes should (1) perform a similar functional task, and (2) be regulated by the same transcription regulation program. Thus, exploiting the guilty by association principle, one can deduce the function of an uncharacterized gene by observing the function of co-expressed genes [71]. Also, by including additional data in the analysis, such as regulatory regions, one can explore and uncover regulatory programs controlling the expression of genes [212].

One traditional approach for finding co-expressed genes is the use of clustering methods, also known as unsupervised learning [64]. Clustering methods are usually based on a similarity metric, which defines how close objects (or gene profiles) are in a given multidimensional space, followed by a method that, for example, searches for groups (or clusters) of objects that lie in compact regions and are far apart from other groups. While cluster analysis is a well-developed research area [109], the characteristics of gene expression data impose challenges not previously addressed by classical clustering methods.

This thesis uses mixture models as a statistical formalism for performing clustering of gene expression data [145]. Mixture models are robust to noise, can model uncertainty about cluster assignments, allow the inclusion of prior knowledge, such as intrinsic dependencies of the experimental design, and offer a flexible framework for integration of additional biological data.

In Chapter 2, we introduce the mixture model formalism and the method used for estimating mixture models; the expectation-maximization (EM) algorithm. Then, in Chapter 3, we describe how mixture models can be used to solve the clustering problem, and how questions as choosing the number of clusters and cluster validation can be answered in the context of mixture models. Additionally, in Chapter 3 we propose a novel external index for validating clustering computed by mixtures. With the exception of the proposal of this external index, Chapters 2 and 3 basically review established research on mixture models, and introduce the methodological framework used in the bioinformatic applications described in later chapters.

Mixture models allow, with a proper choice of component models, to make explicit assumptions about the data. This thesis proposes two novel types of components models for analyzing gene expression profiles. The use of hidden Markov models with linear topologies to analyze gene expression time courses will be the focus of Chapter 4. In Chapter 5, we propose a new type of probabilistic model, dependence trees, to model gene expression profiles during a developmental process. This approach assumes that the sequence of changes from a stem cell to a particular mature cell, as described by a developmental tree, are the most important in modeling gene expression from developmental processes. We also explore in Chapter 5 the benefits of using priors of model parameters to obtain maximum-a-posteriori point estimates, and how this improves the robustness of the method.

Once a given component model is defined, it is straightforward to apply any extension of the expectation-maximization (EM) algorithm. We propose, in Chapter 6, the use of an established semi-supervised learning method [123] to integrate additional biological data and improve clusterings of gene expression time-courses. We evaluated the inclusion of Gene Ontology annotations [9] and location analysis of transcription factor biding derived from Chip-on-chip experiments [128]. Additionally, we propose a novel method, which combines gene expression time-courses with location of gene expression in Drosophila embryos [214], for finding groups of syn-expressed genes. Finally, in Chapter 7, we present final remarks and future work with respect to the specific contribution of this thesis.

Chapter 2

Finite Mixture Models

A finite mixture model is a convex combination of two or more probability density functions. By combining the properties of the individual probability density functions, mixture models are capable of approximating any arbitrary distribution [145]. Consequently, finite mixture models are a powerful and flexible tool for modeling complex data. Mixture models have been used in many applications in statistical analysis and machine learning such as modeling, clustering, classification and latent class and survival analysis. In this chapter, we will introduce the basics about mixture models. Thereby, we define the statistical and computational framework that will be further explored for specific bioinformatics applications in the subsequent chapters. All the content covered in this chapter is a review of established research in the area and can be found, for example, in the textbooks [93, 142, 145].

First, we describe the basic concepts and notations used through this thesis (Section 2.1). Then, we introduce mixture models formally (Section 2.2), show how a mixture model can be efficiently estimated with the expectation-maximization (EM) algorithm (Section 2.3), give an example of mixture models with multivariate Gaussians (Section 2.3.3) and discuss some aspects of model selection and determination of the number of components (Section 2.3.5).

2.1 Basics

A continuous L-dimensional random variable will be denoted as $X = (X_1, ..., X_l, ..., X_L)$, where X_l corresponds to the lth variable. Lower case letters will be used for a particular observation (or realization) $x = (x_1, ..., x_l, ..., x_L)$ of a variable X. Bold face letters, such as **X**, will denote a data of N observations of variable X or, equivalently, a $N \times L$ matrix, where x_{il} is the value of the ith observation for the lth variable in **X**. This notation is based on the one introduced in the textbook [93].

A probability density function (pdf) $p(x)$ is any function defining the probability density of a variable X such that $p(x) \geq 0$ and $\int_{-\infty}^{\infty} p(x) = 1$. By integrating $p(x)$ over an interval, we obtain the probability that variable X assumes values in the interval $[a, b]$, that is

$$\mathbf{P}[a \leq X_i \leq b] = \int_a^b p(x)\, dx.$$

For a given pdf $p(x)$, the expectation of X is defined as,

$$E[X] = \int_{-\infty}^{\infty} xp(x)\, dx. \tag{2.1}$$

In relation to the model parameters, we use the "hat" symbol to indicate an estimator. For example $\hat{\theta}$ is the estimator of parameter θ.

2.2 Mixture Models

Let $X = (X_1, ..., X_j, ..., X_L)$ be a L-dimensional continuous random variable and $x = (x_1, ..., x_L)$ be an observation of X. A probability density function (pdf) of a mixture model is defined by a convex combination of K component pdfs [145],

$$p(x|\Theta) = \sum_{k=1}^{K} \alpha_k p_k(x|\theta_k), \tag{2.2}$$

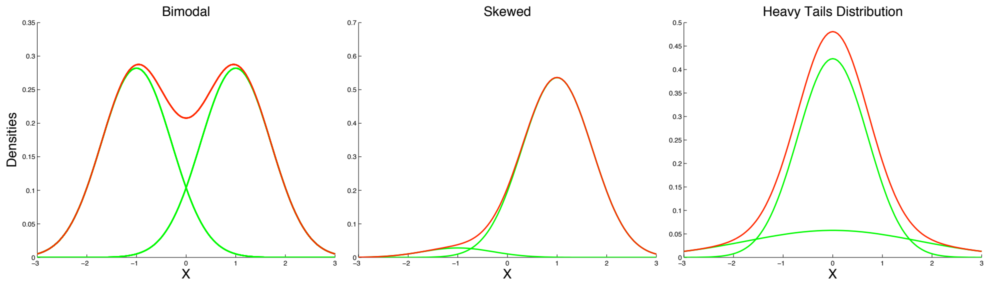
where $p_k(x|\theta_k)$ is the pdf of the kth component, α_k are the mixing proportions (or component priors) and $\Theta = (\alpha_1, ..., \alpha_K, \theta_1, ..., \theta_K)$ is the set of parameters. We assume that

$$\alpha_k \geq 0, \text{ for } k \in \{1, ..., K\}, \text{ and} \tag{2.3}$$

$$\sum_{k=1}^{K} \alpha_k = 1. \tag{2.4}$$

By the property of convexity, given that each $p_k(x|\theta_k)$ defines a probability density function, $p(x|\Theta)$ will also be a probability density function.

The most straightforward interpretation of mixture models is that the random variable X is generated from K distinct random processes. Each of these processes is modeled by the density $p_k(x|\theta_k)$, and α_k represents the proportion of observations from this particular process. For example, the mixture in Figure 2.1 (a) models a bimodal density generated by two independent processes. A mixture can also, by combining simpler densities, model pdfs of arbitrary shapes. For example, with two Gaussian densities as components, we can model a skewed density Figure 2.1 (b), or a heavy tail density Figure 2.1 (c).



(a) (b) (c)

Figure 2.1: *Examples of densities modeled by mixtures of two Gaussians pdfs. Green lines indicate the individual component densities and red lines the mixture densities. In Figure (a), we have a highly overlapping bimodal density, while in Figure (b), we depict an unimodal density skewed to the left, while in Figure (c) a density with heavy tails. These are only a few examples representing the power of mixture models in modeling densities of arbitrary shapes.*

2.3 Mixture Model Estimation

For a given data $\mathbf{X}$ with N observations, the likelihood of the data assuming that x_i are independently distributed is given by

$$p(\mathbf{X}|\Theta) = \mathcal{L}(\Theta|\mathbf{X}) = \prod_{i=1}^{N}\sum_{k=1}^{K}\alpha_k\cdot p_k(x_i|\theta_k). \tag{2.5}$$

The problem of mixture estimation from data $\mathbf{X}$ can be formulated as to find the set of parameters Θ that gives the maximum likelihood estimate (MLE) solution

$$\Theta^* = \arg\max_{\Theta}\mathcal{L}(\Theta|\mathbf{X}). \tag{2.6}$$

The summation inside the product in Eq. 2.5 prevents the possibility of analytical solutions. One alternative is to maximize the complete likelihood in an expectation-maximization (EM) approach [61].

2.3.1 Expectation-maximization Algorithm

The expectation-maximization (EM) algorithm is a general method for finding maximum likelihood estimates when there are missing values or latent variables [61]. In the mixture model context, the missing data is represented by a set of observations $\mathbf{Y}$ of a discrete random variable Y, where $y_i \in \{1, ..., K\}$ indicates which mixture component generated the observation x_i. For now, we will assume that the number K is fixed and known a priori.

The likelihood of the complete data $(\mathbf{X}, \mathbf{Y})$ takes the following multinomial form

$$\begin{aligned} p(\mathbf{X}, \mathbf{Y}|\Theta) = \mathcal{L}(\Theta|\mathbf{X}, \mathbf{Y}) &= p(\mathbf{X}|\mathbf{Y}, \Theta)p(\mathbf{Y}|\Theta) \\ &= \prod_{k=1}^{K}\prod_{i=1}^{N}(\alpha_k \cdot p_k(x_i|\theta_k))^{\mathbf{1}(y_i=k)} \end{aligned} \tag{2.7}$$

where $\mathbf{1}$ is the indicator function, i.e. $\mathbf{1}(y_i = k) = 1$ if $y_i = k$ holds, and $\mathbf{1}(y_i = k) = 0$ otherwise.

The EM algorithm is derived as follows. Let Q be an auxiliary function, the conditional expectation of the complete data $(\mathbf{X}, \mathbf{Y})$, given the observed data $\mathbf{X}$ and a parameterization Θ^{p-1},

$$\begin{aligned} Q(\Theta, \Theta^{p-1}) &= E[\log(p(\mathbf{X}, \mathbf{Y}|\Theta))|\mathbf{X}, \Theta^{p-1}) \\ &= \sum_{\mathbf{Y}\in\mathcal{Y}} p(\mathbf{Y}|\mathbf{X}, \Theta^{p-1}) \log(p(\mathbf{X}, \mathbf{Y}|\Theta)), \end{aligned} \tag{2.8}$$

where $\mathcal{Y}$ is the space of all possible values of $\mathbf{Y}$ and $p(\mathbf{Y}|\mathbf{X}, \Theta^{p-1}) = \prod_{i=1}^{N} p(y_i|x_i, \Theta^{p-1})$. As $\mathcal{Y}$ is the space of all possible values of $\mathbf{Y}$, it follows that

$$\sum_{\mathbf{Y}\in\mathcal{Y}} p(\mathbf{Y}|\mathbf{X}, \Theta^{p-1}) = 1. \tag{2.9}$$

By Bayes rule we can re-write the likelihood function (Eq. 2.5) as

$$p(\mathbf{X}|\Theta) = \frac{p(\mathbf{X}, \mathbf{Y}|\Theta)}{p(\mathbf{Y}|\mathbf{X}, \Theta)}. \tag{2.10}$$

Then, applying the logarithm function to Eq. 2.10 and by Eq.2.9, it follows that

$$\log p(\mathbf{X}|\Theta) = \sum_{\mathbf{Y}\in\mathcal{Y}} p(\mathbf{Y}|\mathbf{X}, \Theta^{p-1}) \log p(\mathbf{X}, \mathbf{Y}|\Theta) - \sum_{\mathbf{Y}\in\mathcal{Y}} p(\mathbf{Y}|\mathbf{X}, \Theta^{p-1}) \log p(\mathbf{Y}|\mathbf{X}, \Theta). \tag{2.11}$$

Next, by replacing the definition of Q (Eq. 2.8) in Eq. 2.11, we can represent the ratio $\log(p(\mathbf{X}|\Theta)/p(\mathbf{X}|\Theta^{p-1}))$ by

$$\begin{aligned} \log p(\mathbf{X}|\Theta) - \log p(\mathbf{X}|\Theta^{p-1}) &= Q(\Theta, \Theta^{p-1}) - Q(\Theta^{p-1}, \Theta^{p-1}) \\ &\quad + \sum_{\mathbf{Y}\in\mathcal{Y}} p(\mathbf{Y}|\mathbf{X}, \Theta^{p-1}) \log \frac{p(\mathbf{Y}|\mathbf{X}, \Theta^{p-1})}{p(\mathbf{Y}|\mathbf{X}, \Theta)} \end{aligned} \tag{2.12}$$

The last term of this equation is equal to the relative entropy between the two densities, and by definition have always positive value [54]. Thus, it follows that

$$\log p(\mathbf{X}|\Theta) - \log p(\mathbf{X}|\Theta^{p-1}) \geq Q(\Theta, \Theta^{p-1}) - Q(\Theta^{p-1}, \Theta^{p-1}). \tag{2.13}$$

Given a parameterization Θ^p such that

$$\Theta^p = \arg\max_{\Theta} Q(\Theta, \Theta^{p-1}), \tag{2.14}$$

and substituting Θ^p in Eq 2.13, we obtain

$$\begin{aligned} \log p(\mathbf{X}|\Theta^p) - \log p(\mathbf{X}|\Theta^{p-1}) &\geq Q(\Theta^p, \Theta^{p-1}) - Q(\Theta^{p-1}, \Theta^{p-1}) \\ &\geq Q(\Theta, \Theta^{p-1}) - Q(\Theta^{p-1}, \Theta^{p-1}) \\ &\geq 0 \end{aligned}$$

and consequently

$$\log p(\mathbf{X}|\Theta^p) \geq \log p(\mathbf{X}|\Theta^{p-1}). \tag{2.15}$$

Intuitively, this means that by maximizing Q (Eq. 2.8) in regard to a parameterization Θ^{p-1}, we obtain a parameterization Θ^p that maximizes the log likelihood (Eq. 2.5). Based on this result, the EM algorithm works by iterating between two steps. In the first (E-step), it finds the expected value of the complete likelihood given the current parameterization Θ^{p-1}. In the second step (M-step), it looks for the set of parameters Θ^p that maximize the expectation from the E-step. At each iteration, the EM increases the log-likelihood converging to a local maximum [61]. These steps are repeated P times or until a convergence criterion is fulfilled.

Before proceeding with the deduction, we need to define the posterior probability of $y_i = k$, given x_i. By Bayes rule this can be defined as follows [145],

$$\begin{aligned} p(y_i = k|x_i, \Theta) &= \frac{p(y_i = k)p(x_i|y_i = k, \theta_k)}{p(x_i|\Theta)} \\ &= \frac{\alpha_k p_k(x_i|\theta_k)}{\sum_{k'=1}^{K} \alpha_{k'} p_{k'}(x_i|\theta_{k'})} \end{aligned} \tag{2.16}$$

For simplicity of notation we denote $p(y_i = k|x_i, \Theta)$ by r_{ik}.

In the case of mixture models, Eq. 2.8 can be re-written, after some mathematical manipulations [27], as follows

$$Q(\Theta, \Theta^{p-1}) = \sum_{k=1}^{K} \sum_{i=1}^{N} r_{ik} \log(\alpha_k \cdot p_k(x_i|\theta_k^{p-1})). \tag{2.17}$$

For the E-Step, we need to find the expected value of $\mathcal{L}(\Theta|\mathbf{X}, \mathbf{Y})$ given x_i and the current parameterization. As $\log(\mathcal{L}(\Theta|\mathbf{X}, \mathbf{Y}))$ is linear in x_i, this step reduces to calculating the expected value $y_i = k$ given x_i and the previous parameterization Θ^{p-1}, that is

$$\begin{aligned} E[y_i = k|x_i, \Theta^{p-1}] &= p(y_i = k|x_i, \Theta^{p-1}) \\ &= r_{ik}. \end{aligned} \tag{2.18}$$

The M-Step can be formally described as

$$\Theta^p = \arg\max_{\Theta} Q(\Theta, \Theta^{p-1}). \tag{2.19}$$

To find the parameter estimates, we need to integrate Eq. 2.8 in relation to its parameters Θ in a maximum likelihood fashion.

For the α_k, the MLE estimate can be obtained as

$$0 = \left[\sum_{k=1}^{K}\sum_{i=1}^{N} r_{ik}\log(\alpha_k \cdot p_k(x_i|\theta_k)) + \lambda(\sum_{k=1}^{K}\alpha_k - 1)\right]\frac{\partial}{\partial\alpha_k} \tag{2.20}$$

$$0 = \sum_{i=1}^{N}\frac{1}{\alpha_k}r_{ik} + \lambda \tag{2.21}$$

where λ is a Lagrange multiplier that guarantees stochasticity (Eq. 2.4). Setting $\lambda = -N$, we have

$$\alpha_k = \frac{\sum_{i=1}^{N} r_{ik}}{N}. \tag{2.22}$$

The estimates of θ_k will be specific to the choice of the component densities. For many families of densities, such as exponential type densities, there are analytical solutions (see Section 2.3.3). Even for cases where the maximum likelihood estimate cannot be found, it is sufficient to find a parameterization Θ^{p-1}, such that

$$Q(\Theta^p, \Theta^{p-1}) > Q(\Theta, \Theta^{p-1}). \tag{2.23}$$

This is the case, for example, when Hidden Markov Models (HMM) are used as the component densities. In this scenario, we can apply the Baum-Welch algorithm [21] for each component of the mixture at the M-Step of the EM algorithm. This procedure estimates a local maximum likelihood estimate of a HMM, and meets Eq. 2.23. This estimation method is known as the generalized expectation-maximization algorithm [27].

2.3.2 Method Initialization

An important point of the EM algorithm is the selection of the initial parameterization Θ^0 of the model. A standard way to obtain Θ^0 is to choose random r_{ik} values uniformly from $[0, 1]$ and estimating the individual models with the M-Step. In order to deal with the effects of the random initialization, all estimations are repeated a number of times (usually 15), and the solution with highest likelihood is selected [143].

2.3.3 Mixture of Multivariate Gaussians

As an example, we show how the estimates of a mixture with multivariate Gaussians can be computed. The probability density function of X is defined as

$$p(x|\theta) = \frac{1}{\sqrt{2\pi|\Sigma_x^{-1}|}} \exp\left(-\frac{1}{2}(x-\mu_x)\Sigma_x^{-1}(x-\mu_x)^T\right) \tag{2.24}$$

where μ_x is a vector of means $(\mu_{x_1}, ..., \mu_{x_L})$, Σ_x is the $L \times L$ covariance matrix, and $\theta = (\mu_x, \Sigma_x)$. By replacing 2.24 in 2.17, we obtain,

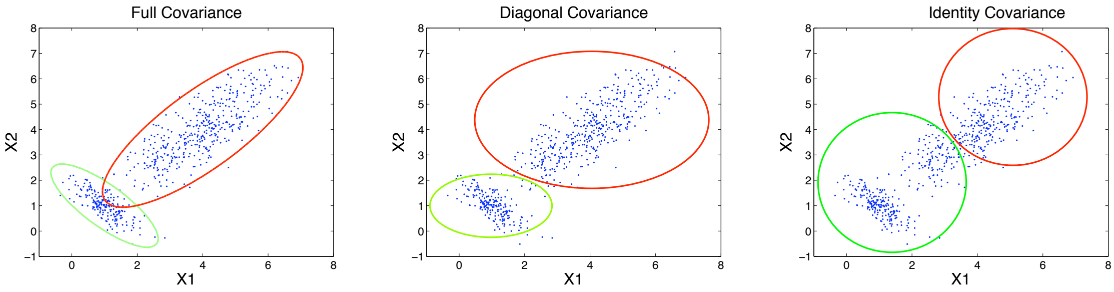
$$\begin{aligned} Q(\Theta, \Theta^{p-1}) &= \sum_{k=1}^{K}\sum_{i=1}^{N} r_{ik}\log(\alpha_k) - \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N} r_{ik}\log(2\pi|\Sigma_{x|k}^{-1}|) \\ &\quad -\frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N} r_{ik}(x-\mu_{x|k})\Sigma_{x|k}^{-1}(x-\mu_{x|k})^T, \end{aligned} \tag{2.25}$$

where $\theta_k = (\mu_{x|k}, \Sigma_{x|k})$ are the parameters of the pdf p_k. Subscripts on parameter the $\mu_{x|k}$ indicate that the parameter μ is an estimate of the variable X and it is conditioned on the mixture model component k. By taking the derivative of Eq. 2.25 in respect to $\theta_k = (\mu_{x|k}, \Sigma_{x|k})$, we obtain the following estimates,

$$\hat{\mu}_{x|k} = \frac{\sum_{i=1}^{N} r_{ik}x_i}{\sum_{i=1}^{N} r_{ik}}, \text{and,} \tag{2.26}$$

$$\hat{\Sigma}_{x|k} = \frac{\sum_{i=1}^{N} r_{ik}(x_i-\mu_{x|k})(x_i-\mu_{x|k})^T}{\sum_{i=1}^{N} r_{ik}}. \tag{2.27}$$

Mixture of multivariate Gaussians are able to model groups of observations in ellipsoidal regions of the Euclidean space with any orientation and size. See Figure 2.2 for an example. In many situations, it may be desirable to use models with simpler assumptions, and consequently fewer parameters. One alternative is to restrict the covariance matrix



(a) (b) (c)

Figure 2.2: *Example of solutions in a two dimensional data found by a mixture of Gaussians with full covariance matrices (a), mixture of Gaussians with diagonal covariance matrices (b), and mixture of Gaussians with identity covariance matrices (c). The ellipsoids correspond to the region with 95% of the component density. With the full covariance matrix, the mixture fits the two groups shapes well. With the diagonal covariance matrix, the components also model similar groups of observations compared to the full covariance matrix. However, for the former, the density cover regions of the space without observations. Gaussians with identity covariance matrices, which can only find spherical and equal size components, cannot model the two groups of observations well.*

(Eq. 2.27) to the diagonal entries

$$\Sigma^d_{x|k} = \text{diag}(\Sigma_{x|k}), \tag{2.28}$$

where $\text{diag}(\Sigma)$ denotes a matrix, which has same values as the diagonal of the matrix Σ and zero for all off diagonal entries. In this case, we obtain pdfs with ellipsoidal shape, but with orientation parallel to the coordinate (see Figure 2.2). Another possibility is to restrict all covariance matrices of the components to be the identical, which leads to all components having the same shape and orientation. The most simplistic assumption is the use of identity covariance matrices such that $\Sigma^*_k = \sigma^2 I$ and $\alpha_k = 1/K$. In this case, all components cover spherical and equal size regions of the space (see Figure 2.2 for a comparison of distinct parameterization in a toy data). See [10] and [37] for a complete listing of possible parameterizations of the covariance matrix of a multivariate Gaussian.

2.3.4 EM and Local Maxima

Ideally, one would like to use the full covariance matrix parameterization, as it model all covariance between variables. However, with such covariance matrices, the EM usually returns local maximizers, characterized by having a component with few observations assigned to it [143]. In other words, the mixture fits perfectly a small part of the data, obtaining a high likelihood, but does not achieve a good fit for other regions of the space. This follows from the fact that the likelihood function is unbounded on boundaries of the parameter space (very low values of α or diagonal entries of Σ.) In particular, when the number of observations (N) in the data is low, or in the presence of outliers, such solutions will be often found by the EM algorithm [143].

To prevent this, there are several techniques available. One simple method [95] is to constrain the diagonal values of the covariance matrices to never be below a given threshold value. Another alternative, which minimizes the effects of outliers, is to use alternative density functions, such as the student [144], or the use of noise components [10].

A more principled approach is to define prior density functions on the mixture parameters and perform a maximum-a-posteriori (MAP) estimation with Monte Carlo Markov Chains (MCMC) [65, 84, 180]. This requires the specification of a proper conjugate prior on the parameters. For example, [65] considers a Wishart density function as a prior on Σ_k and Dirichlet distributions for the component responsibilities α. However, MCMC has a higher computational cost than the EM algorithm. Recently, [80] showed that for multivariate Gaussians, where the posterior mode solution is given with the use of conjugate priors, EM estimation with point MAP estimates achieves comparable results to those obtained with the computationally costly MCMC.

2.3.5 Determining the Number of Components

We cannot rely on maximal likelihood to estimate the number of components, since overfitted solutions, such as one component per observation would arise (see Figure 2.3). We need to balance between fit versus generality. This is commonly done with a penalized likelihood approach, as the Bayesian information criterion (or BIC for short) [191], and further extensions [26, 38, 227]. The problem of finding the number of components can also be tackled in a Full Bayesian setting using Dirichlet Process priors [75]. However, this approach requires the use of the computationally expensive MCMC. Despite its simplicity, BIC performs well in simulation studies [145]. Thus, it will be the methodology used throughout this thesis for selecting the number of components.

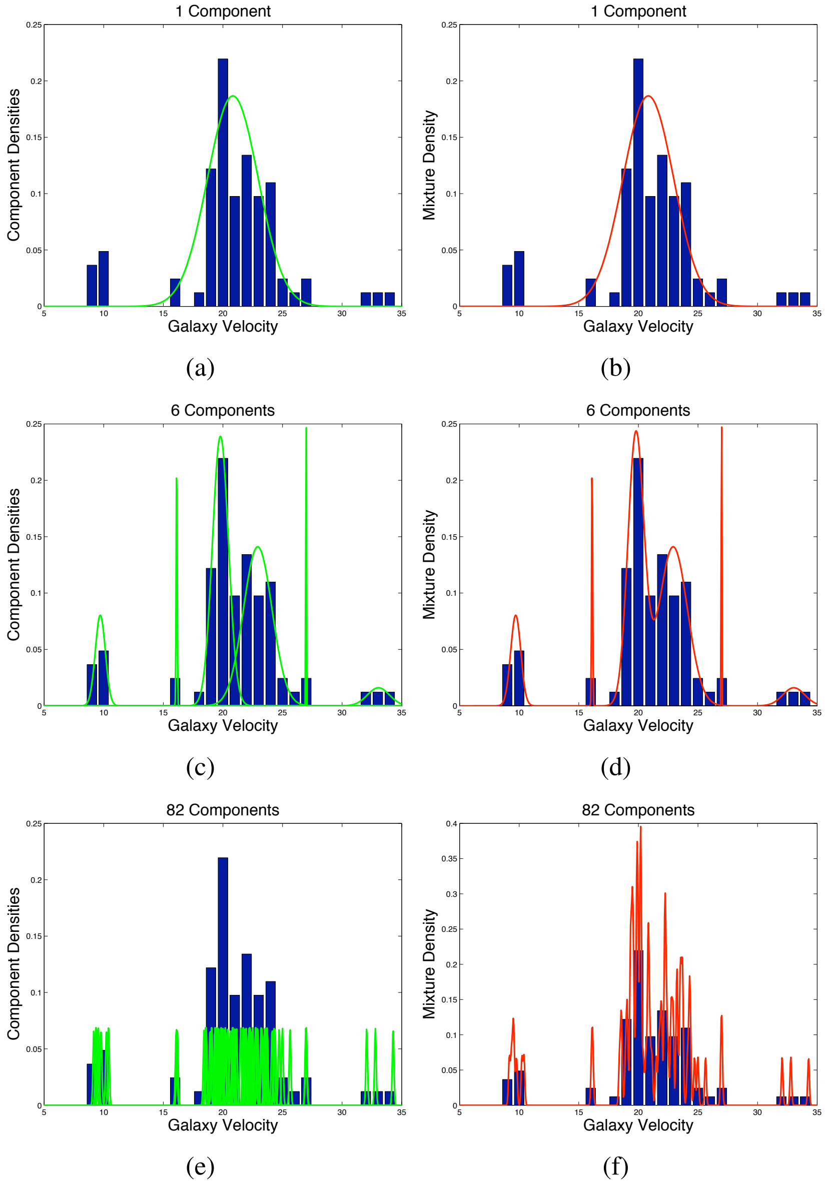
We can tackle the selection of the number of components in a Bayesian framework by comparing two mixture models Θ_K and Θ_{K+1} with Bayes Factors. We calculate the ratio of posterior,

$$B_{K,K+1} = p(X,Y|\Theta_K)/p(X,Y|\Theta_{K+1}), \tag{2.29}$$

where Θ_K and Θ_{K+1} are the parameters of two mixture models with K respectively $K+1$ components. It is possible to compare several models at once, rather than two by two as in frequentist statistical test. When we use the EM-algorithm to estimate maximum likelihood mixture models, approximate Bayes factors can be easily deduced from the Bayesian information criterion (BIC) [191],

$$-2\log p(X,Y|\Theta_K) \approx -2\log \mathcal{L}(\Theta_K|\mathbf{X},\mathbf{Y}) + \psi_K \log N, \tag{2.30}$$

where K is the number of components, $\mathcal{L}(\Theta_K|\mathbf{X},\mathbf{Y})$ is the maximized mixture log-likeli-



(a) (b)

(c) (d)

(e) (f)

Figure 2.3: *Examples of mixture models with 1, 6 and 82 components fitting the Galaxy Velocity data [176]. On the left plots, we have the density of individual components and the histogram of the data, while in the right we have the mixture density and the data histogram. The mixture with one component models roughly the density in the range* $[15, 25]$*, and imposes zero density to other ranges of the density, plots (a) and (b). The mixture with 82 components, the maximum likelihood solution for number of components equal to the number of observations, simply over-fits the data, plots (e) and (f). The solution with 6 components offers a trade off between these two solutions, providing a good fit of the data, modeling well all ranges of the density, plots (c) and (d) . This mixture was presented in [145] as the optimal solution for the Galaxy data.*

hood with K components (Eq. 2.7), ψ_K is the number of free parameters in Θ_K and N is the number of observations in $\mathbf{X}$.

The term $\psi_K \log N$ penalizes more complex models, since the fit of a model tends to improve as the number of parameters increases. The smaller the value of BIC, the better the model. It has been shown that BIC does not underestimate the number of true components asymptotically and performs well in simulation studies [145]. In the case of a multivariate Gaussians, parameterized by (μ_x, Σ_x), the number of free parameters in a model θ_k is equal to $L + L(L-1)/2$. Hence,

$$\psi_K = K * (L + L(L-1)/2). \tag{2.31}$$

This chapter covered the basics aspects on mixture models and their estimation. In the next chapter, we show how mixture models can be used in the context of clustering. Furthermore, for specific applications, as the ones described in Chapter 4 or in Chapter 5, we take advantage of the characteristics of the data at hand, and choose the component models accordingly.

Chapter 3

Mixture Models and Clustering

In this thesis, we focus on the use of mixture models to perform clustering. By clustering we mean finding groups (or clusters) of observations in a finite data set, such that each group represents observations sharing characteristics, which are distinct from the overall data set. Mixture models, being based on statistics, tackle this problem in a formal and principled way. The applications of mixture models for clustering [79] has a number of advantages in contrast to classical clustering methods such as k-means and hierarchical clustering: it quantifies the uncertainty of a given cluster assignment; the estimated models are descriptors of the groups found; and it is possible to answer questions such as the number of clusters in a purely statistical way [145].

The characteristics of mixture models make this approach of great value in the analysis of biological data. In particular, for gene expression analysis [11], the main interest is on finding groups of genes that have similar expression patterns through a set of experimental conditions, and possibly are part of a biological functional module [71, 135]. However, a single gene (and its products) can participate simultaneously in more than one functional module, for example by taking part in distinct protein complexes, each with its particular function [122]. Furthermore, data arising from large-scale experiments, such as microarray measurements of gene expression, contain large amount of noise [135]. In this context, overlapping clusterings, such as the ones given by a mixture models, represent the results of gene expression clustering analysis in a more natural way than "hard" clusterings. Also, the uncertainty of a given cluster assignment returned by the mixture model is a valuable information in the distinction of assignments derived from relevant and noisy observations. Furthermore, the mixture components can model particular assumptions about the data, such as temporal dependencies, therefore producing more reliable estimates. As an evidence, there is a vast list of publications that successfully applied mixture models in finding potentially overlapping groups in gene expression analysis [14, 138, 143, 147, 155, 185–187, 232, 234].

This chapter is organized as follows: Section 3.1 gives a definition of the use of mixture models to perform clustering, and introduces how clusters can be obtained from mixtures. Later, we propose a novel external index to perform validation of mixture models in Section 3.2, which is evaluated with simulated data in Section 3.3.

3.1 Clustering with Mixture Models

Clustering is the task of partitioning a data set of N objects (or observations) from $\mathbf{X}$ into K disjoint groups (or clusters). We represent a data set by a $N \times L$ matrix $\mathbf{X}$, where entry x_{ij} denotes the values of the jth variable of the ith object. The clustering (or partition) can be represented by $\mathbf{Y}$, where $y_i \in \{1, .., K\}$ indicates the group to which a given object x_i belongs to [142]. In mixture model based clustering, we assume that each component in the mixture represents a group of objects. In other words, the density of the kth component can be interpreted as the conditional of x_i on y_i, i.e., $p_k(x_i|\theta_k) = p(x_i|y_i = k, \theta_k)$, and the mixing coefficient as the prior probability of the component, i.e., $\alpha_k = p(y_i = k)$.

For a given data set $\mathbf{X}$ with N observations and a variable Y defining the component assignments, the likelihood of the complete data assuming that the x_i are independently distributed is given by

$$p(\mathbf{X}, \mathbf{Y}|\Theta) = \mathcal{L}(\Theta|\mathbf{X}, \mathbf{Y}) = \prod_{k=1}^{K} \prod_{i=1}^{N} (\alpha_k \cdot p_k(x_i|\theta_k))^{\mathbf{1}(y_i=k)} \tag{3.1}$$

Thus, the problem of clustering observations from $\mathbf{X}$ can be formulated as finding the maximum likelihood estimate (MLE)

$$\Theta^* = \arg\max_{\Theta} \mathcal{L}(\Theta|\mathbf{X}, \mathbf{Y}). \tag{3.2}$$

This problem, as described in Section 2.3, can be solved by maximizing the complete likelihood using the EM algorithm [61].

The posterior probability (Eq. 2.16) of a mixture model reveals the probability of a cluster assignment. The simplest way of decoding a mixture, that is, to infer clusters in the data, is to interpret the mixture components as descriptive models of non-overlapping clusters and assign each object x_i to the cluster k of maximal posterior,

$$y_i = \arg \max_{1 \leq k \leq K} (r_{ik}). \tag{3.3}$$

In model-based clustering as well as k-means, these hard assignments are performed after each E-Step, while for the mixtures this is only necessary after estimation is finished. Indeed, a mixture of Gaussians with the identity covariance described in Section 2.3.3, where $\sigma^2 \rightarrow 0$ and hard assignments are performed, is equivalent to the k-means algorithm [88]. In the next chapters, we will refer to model-based clustering whenever such hard assignments are performed during the EM, and to mixture estimation otherwise.

An inspection of the distribution of the posterior probability of component assignments given an object x_i, i.e., $r_i = (r_{i1}, ..., r_{iK})$, reveals the level of ambiguity in making the cluster assignments. Therefore, we propose here a novel decoding method, entropy thresholding, which takes the ambiguity of assignments into account. As depicted in Figure 3.1,

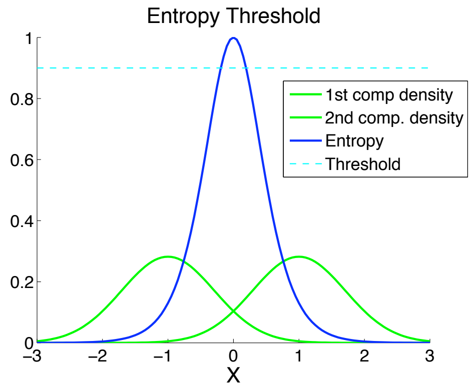


Figure 3.1: *Entropy of posterior assignments for the bimodal density from Figure 2.1. Values in between the two density functions have high entropy, and would be discarded by the entropy threshold method. If we select $\varphi = 0.9$, objects in the range $[-0.2, 0.2]$ will be assigned to the cluster $K + 1$.*

this ambiguity can be quantified by computing the Shannon entropy [54]

$$\mathrm{H}(r_i) = -\sum_{k=1}^{K} r_{ik} \log \frac{1}{r_{ik}}. \tag{3.4}$$

Choosing a threshold φ for the entropy yields a grouping of the data into at most $K + 1$ groups. If $\mathrm{H}(r_i) < \varphi$, we assign x_i to the component with maximal posterior as in Eq. 3.3. Otherwise, x_i is assigned to the $(K + 1)$-st group, which contains all objects which cannot be assigned unambiguously, that is

$$y_i = \begin{cases} \arg\max_{1 \le k \le K}(r_{ik}), & \mathrm{H}(r_i) < \varphi \\ K + 1, & \text{otherwise.} \end{cases} \tag{3.5}$$

3.2 Validation of Mixture Models

The task of obtaining a mixture model does not end with the parameter estimation. Questions on the number of components and the quality of the representation of the data often arise after this step. In classical clustering, there are several methodologies, under the name of cluster validation, proposed for answering these questions. These methodologies, mainly based on re-sampling techniques and fit indices, have been proposed in the vast cluster validation literature [28, 56, 66, 124, 141, 233] and reviewed for example in [89, 109]. Nevertheless, the mixture model framework embraces challenges and characteristics not explored by “classical” cluster validation techniques.

One often over-looked aspect is the use of external indices, which are used to compare the similarity of a cluster solution to a gold standard or to another clustering solution.

Most external indices proposed so far are only able to measure the agreement between two non-overlapping clusterings [109]. Mixtures, however, can be interpreted as a partition with overlap, and encode more information than non-overlapping partitions. Therefore, the overlap, encoded by the posterior distributions of the mixture, should be taken into consideration when for example two mixtures are compared. Additionally, there are cases, where even if the clustering results are non-overlapping partitions, the a priori labels are based on overlapping partitions. This is the case, for example, of functional annotation of genes [9].

Motivated by the previous problem, in the next Section, as one of the contribution of this thesis, we propose a novel external index that can used for the comparison of mixture models and overlapping partitions [51]. Such an index is an extension of a widely employed external index for comparing hard partitions — the corrected Rand index [103].

In Section 3.2.1, we introduce the basics of external indices of non-overlapping partitions, and we define the extension for the overlapping case. Finally, in Section 3.3, we employ simulated data for assessing the characteristics of the external indices in data with overlap.

3.2.1 External Indices

External indices assess the agreement between two partitions defined over the same set of objects, where one partition $\mathbf{Y}$ represents the result of a clustering method, and the other partition $\mathbf{Y}'$ represents class labels[1]. While a number of external indices have been introduced in the literature, the use of corrected Rand (`CR`) is recommended [103]. `CR` has its value corrected for chance agreement, it is not dependent on the cluster size distributions and can compare partitions with distinct number of clusters [149]. See [110] for a comprehensive review of external indices.

Let $\mathbf{Y}$ and $\mathbf{Y}'$ be discrete vectors representing the partitions yielded by a clustering method and the class labels. Let $y_i \in \{1,...,K\}$ and $y'_i \in \{1,...,L\}$ be, respectively, observations from $\mathbf{Y}$ and $\mathbf{Y}'$, where $y_i = k$ indicates that object i belongs to cluster k. Note, K and L can be distinct. Thus, the following indicator functions can be defined

$$\mathbf{1}(y_i = y_j) = \begin{cases} 1, & \text{if } y_i = y_j \\ 0, & \text{otherwise} \end{cases} \text{ and,} \tag{3.6}$$

$$\mathbf{1}(y'_i = y'_j) = \begin{cases} 1, & \text{if } y'_i = y'_j \\ 0, & \text{otherwise.} \end{cases} \tag{3.7}$$

From these, we can define the following terms

[1] $\mathbf{Y}$ and $\mathbf{Y}'$ can also be partitions from two distinct clustering methods applied to the same data set.

$$a = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \mathbf{1}(y_i = y_j)\mathbf{1}(y'_i = y'_j), \tag{3.8}$$

$$b = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (1 - \mathbf{1}(y_i = y_j))\mathbf{1}(y'_i = y'_j), \tag{3.9}$$

$$c = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \mathbf{1}(y_i = y_j)(1 - \mathbf{1}(y'_i = y'_j)), \text{ and} \tag{3.10}$$

$$d = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (1 - \mathbf{1}(y_i = y_j))(1 - \mathbf{1}(y'_i = y'_j)). \tag{3.11}$$

The term a measures the number of object pairs that are found in the same cluster in both $\mathbf{Y}$ and $\mathbf{Y}'$. It is the equivalent of the number of true positives commonly used in the machine learning literature. Analogously, b, c and d correspond respectively to the number of false positives, false negatives and true negatives. The total number of object pairs p is equal to $p = a + b + c + d$. From these terms, the corrected Rand is defined as [103],

$$\text{CR} = \frac{(a+d) - ((a+b)(a+c) + (c+d)(b+d))p^{-1}}{p - ((a+b)(a+c) + (c+d)(b+d))p^{-1}}. \tag{3.12}$$

CR takes values from -1 to 1, where 1 represents perfect agreement while values of CR near or below 0 represent agreements occurring by chance. The correction of Rand index, proposed in [103], estimates the expected Rand index value by assuming that the baseline distributions of the partitions are fixed. This is equivalent to calculating the expected Rand index value for random permutations of the objects labels in one partition, while the other is fixed.

Two other interesting external indices, which can be defined by the terms in Eq. 3.8, 3.9, 3.10 and 3.11, are the sensitivity and specificity [199],

$$\text{Sens} = \frac{a}{a+c} \tag{3.13}$$

$$\text{Spec} = \frac{a}{a+b} \tag{3.14}$$

They both take values from 0 to 1, where 1 indicates perfect agreement. The use of these indices is complementary to CR, as they indicate for example a tendency to make more false positives or false negative errors — CR treats both errors equally. In practice, a lower sensitivity (more false positives) is an indicator of joining real clusters; while a lower specificity (more false negatives) indicates a tendency to split real clusters.

Extended Corrected Rand

The main idea of the extended corrected Rand (ECR) is to redefine the indicator functions, as defined in Eq. 3.6 and Eq. 3.7, giving them a probabilistic interpretation [51]. The posterior distribution defines the probability that a given object x_i from $\mathbf{X}$ belongs to the component k, i.e., $y_i = k$, in a mixture model parameterized by Θ, i.e. $p(y_i = k|x_i, \Theta)$. This is exactly the Eq. 2.16, which we refer to as r_{ik} for simplicity. Likewise, we have r'_{il} for indicating the posterior that x_i belongs to component l in $\mathbf{Y}'$. We denote the event that a pair of objects has been generated by the same component in $\mathbf{Y}$, the co-occurrence event, as $x_i \equiv x_j$ given $\mathbf{Y}$. Assuming independence of the clusters from $\mathbf{Y}$, the probability of the co-occurrence of x_i and x_j given $\mathbf{Y}$ for $1 \leq i \leq j \leq N$ can be estimated as

$$p(y_i \equiv y_j \text{ given } Y) = \sum_{k=1}^{K} r_{ik} r_{jk}. \tag{3.15}$$

We use the previous equation to redefine the variables a, b, c and d, used in the definition of CR

$$\begin{aligned}
a &= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} p(y_i \equiv y_j \text{ given } \mathbf{Y}) p(y_i \equiv y_j \text{ given } \mathbf{Y}'), \\
b &= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (1 - p(y_i \equiv y_j \text{ given } \mathbf{Y})) p(y_i \equiv y_j \text{ given } \mathbf{Y}'), \\
c &= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} p(y_i \equiv y_j \text{ given } \mathbf{Y})(1 - p(y_i \equiv y_j \text{ given } \mathbf{Y}'), \text{ and} \\
d &= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (1 - p(y_i \equiv y_j \text{ given } \mathbf{Y}))(1 - p(y_i \equiv y_j \text{ given } \mathbf{Y}')).
\end{aligned} \tag{3.16}$$

From these, the extended corrected Rand (ECR) can be computed by the original formula Eq. 3.12. ECR also takes values from -1 to 1, where 1 represents perfect agreement while values of ECR near or below zero represent agreements occurred by chance. By definition, it works exactly as the corrected Rand when "hard" partitions are given.

3.3 Experiments

To evaluate the extended corrected Rand, we make use of simulated data from mixtures of Gaussians. In the first experiment, we define a very simple scenario with a mixture of two Gaussians components in an univariate space. Hence, we can compare the characteristics

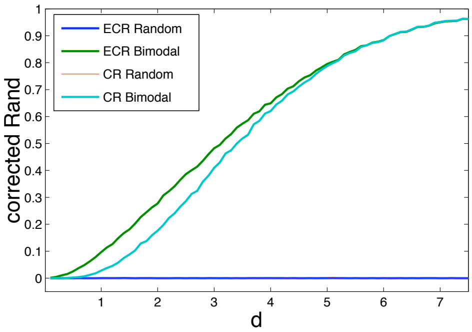


Figure 3.2: *We depict the mean* CR *and* ECR *for the results of the mixture estimation with the normal bimodal density. The larger d, the lower is the overlap between the two components.*

of ECR and CR when distinct degrees of overlap between the components are present. In the second experiment, we sample data from a mixture, where components show a large degree of overlap. From an initial mixture, we vary the number of components and distributions of objects in the components.

The EM algorithm is used to fit the multivariate Gaussian mixtures with full covariance matrices as described in Section 2.3.3. The EM method is initialized as described in Section 2.3.2. In the simulated data experiments, 50 data sets are generated for each proposed mixture.

3.3.1 Simulated Data 1

We perform experiments with a normal mixture with two equiprobable components to evaluate the proposed index characteristics in the presence of distinct degrees of overlap. The components have means $\mu_1 = [0, 0]^T$, $\mu_2 = [d, 0]^T$, and covariance matrices $\Sigma_1 = \Sigma_2 = I$, as suggested in [77]. For obtaining mixtures with distinct degrees of overlap (bimodal data), we vary d in the range $[0.0, 7.5]$. The lower the value d, the higher is the overlap between the two components. For each component we draw 200 objects. The density function given the original mixture parameterization is used to obtain the posterior r'_{il}. We also calculate the values of CR after performing hard assignments of the solutions (Eq. 3.3).

Additionally, we generate random data to function as a null case. This consists of data generated from a single normal component with $\mu = [d/2, 0]^T$ and $\Sigma = I$. A random solution $(\mathbf{Y}')$ with the same number of components and object distributions as the corresponding bimodal data is calculated. For each particular d, we carried out a non parametric equal-means hypothesis test based on bootstrap [70] to compare the mean ECR (or CR) obtained with the bimodal and random data.

Results. As displayed in Figure 3.2, for data with high overlap, ECR has higher values than CR, while for data with low overlap both indices have similar values. With random data, the indices take on mean values near zero and low variance (< 0.001), which indicate that ECR is successful in the correction for randomness.

With respect to the hypothesis test, the equal means hypothesis is rejected with the use of ECR in all $d > 0.0$ with p-value < 0.001. On the other hand, with the use of CR, the null hypothesis (equal means) is only rejected (p-value < 0.001) when low overlap is presented ($d > 0.4$). We can conclude that ECR is able to detect the distinction between the agreement of the random and bimodal data in all cases, while CR fails when a high degree of overlap is present. Furthermore, when overlap is low, both indices behave similarly.

3.3.2 Simulated Data 2

We use a more extensive set of simulated data to evaluate ECR. Based on a mixture defined in [77], which will be called "base mixture", we change and extend its definition to generate data with distinct components densities and number of components. The "base mixture" has four components, three of them with a large overlap and two of them with same mean vectors

$$\begin{aligned}
\mu_1 = & \; [-4,-4], \Sigma_1 = \begin{bmatrix} 6 & -2 \\ -2 & 6 \end{bmatrix}, \\
\mu_2 = & \; [-4,-4], \Sigma_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \\
\mu_3 = & \; [-1,-6], \Sigma_3 = \begin{bmatrix} 0.125 & 0.0 \\ 0.0 & 0.125 \end{bmatrix}, \\
\mu_4 = & \; [2,2], \Sigma_4 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}.
\end{aligned}$$

An example of a data set sampled from this mixture can be seen in Figure 3.3.

As with the bimodal data, we also generate random data from the normal

$$\mu = [0,0], C = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}.$$

For each data set the EM is performed with to 2 to 10 components. As in [149] it is expected that ECR should obtain mean values near zero for random data and low standard error. Additionally, ECR will be maximum at the correct number of components. For comparison, we also compute BIC and CR.

Components Distribution. We use three types of component distributions for the "base mixture": equal density (ED), ($\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0.25$), 10% density (10%) ($\alpha_1 =$

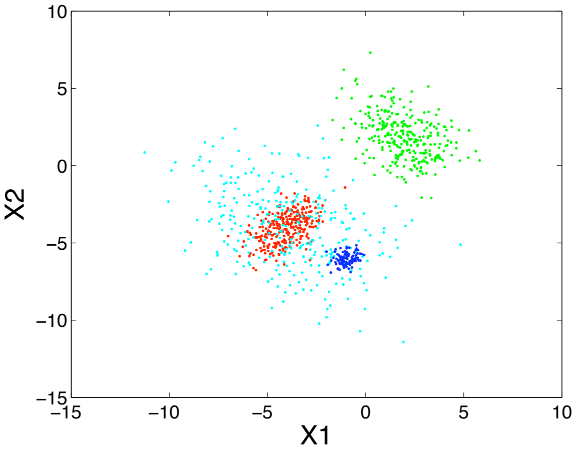


Figure 3.3: *We depict the points of data sampled from the base mixture in the two dimensional space. Objects in light blue corresponds to component 1, in red to component 2, in dark blue to component 3, and in green to component 4. As can be seen, components 1 and 2 have the same mean, but distinct orientations and sizes. Furthermore, component 3 is also inside component 1.*

$\alpha_2 = \alpha_3 = 0.3$ and $\alpha_4 = 0.1$) and 60% density (`60%`) ($\alpha_1 = 0.6$ and $\alpha_2 = \alpha_3 = \alpha_4 = 0.16$).

Number of Components. In addition to the components in the base mixture, we also included the components ($\mu_5 = [-6, -1]^T, \mu_6 = [-12, -12]^T$ and $\Sigma_4 = \Sigma_5$, $\Sigma_3 = \Sigma_6$). We generated data sets with two to six components with 700 observations. For each number of component K, we select the first K components from the mixture. The component distribution used is $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_6 = 0.21$ and $\alpha_4 = \alpha_5 = 0.07$.

Results As can be observed in Table 3.1, `CR` and `ECR` indicate the right number of clusters (four) in all three scenarios of component distributions. Nevertheless, in the setting `10%`, the equal means hypothesis test was not rejected, when comparing the mean of the `CR` with 4 and 5 components. In relation to BIC, it overestimates the number of cluster as five, in the distribution setting `10%`, and as 8 in the distribution setting `60%`.

In relation to the data with distinct number of components, `ECR` indicates the right number in all data sets, as shown in Table 3.2. `CR` overestimates the number of components of the data set with 5 and 6 components indicating 8 components in both cases. BIC can only correctly predict the number of clusters in the data with 2 and 3 components. Note that the degree of overlap varies in the models with distinct number of components, in special for the highly overlapping $c = 2$ and $c = 3$. This makes the mixture estimation task harder, as a result, lower `ECR` (and `CR`) values are obtained in those data sets. Additionally, the estimated mixtures obtain mean `ECR` values near zero and low standard errors (< 0.001) in all situations with the random data (not shown).

Table 3.1: *We depict the mean values for data with distributions* ED *(top),* 10% *(middle) and* 60%*(bottom) against the number of components in the mixtures for corrected Rand, extended corrected Rand and BIC. For all indices, the maximum values (in bold) indicate the predicted number of components.*

no clusters	2	3	**4**	5	6	7	8	9	10
ED CR	0.32	0.68	**0.76**	0.71	0.71	0.69	0.66	0.64	0.65
ECR	0.31	0.64	**0.68**	0.63	0.63	0.60	0.57	0.54	0.55
-BIC	-8.95	-8.20	-8.08	**-7.94**	-8.12	-8.14	-8.00	-8.02	-8.21
10% CR	0.34	0.50	**0.66**	0.66	0.64	0.59	0.57	0.58	0.58
ECR	0.43	0.45	**0.58**	0.55	0.52	0.48	0.45	0.44	0.44
-BIC	-9.03	-8.97	-8.77	**-8.72**	-8.74	-8.76	-8.78	-8.80	-8.80
60% CR	0.36	0.75	**0.81**	0.76	0.74	0.72	0.67	0.66	0.62
ECR	0.35	0.64	**0.71**	0.65	0.61	0.55	0.50	0.44	0.41
-BIC	-8.16	-7.70	-7.45	-7.31	-7.48	-7.35	**-6.92**	-7.09	-7.11

We analyzed how distinct criteria measure the agreement of two mixtures when data is generated from highly overlapping mixtures. The extended Corrected Rand displays better results than the corrected Rand in discriminating the right solutions in all scenarios. Furthermore, ECR behaves similarly to CR when no great overlap is present in the data, and in the correction for randomness. It is important to stress that CR and ECR do not substitute BIC for finding the right number of components, because they require the true labels (or true posteriors). These labels are often not present, and despite the sub-optimal results in this analysis, BIC works reasonably in practice. Nevertheless, if true labels and overlapping assignments are present, ECR is more precise.

In summary, this chapter covers the basic aspects of the use of mixture models to perform clustering. All results discussed here are based on the use of multivariate Gaussians as the components of the mixture. Nevertheless, for specific applications, one can take advantage of the characteristics of the data at hand, and choose the component models accordingly. The EM algorithm offers a flexible framework for such extensions. In practice, for a given model choice, one only needs to redefine the M-Step accordingly.

This thesis focuses on two types of components models for analyzing gene expression profiles. The use of HMMs to analyze gene expression time-courses will be the focus of Chapter 4. While in Chapter 5, we propose a new type of probabilistic model, dependence trees, to model gene expression profiles during a developmental process. Furthermore, once the M-Step for a given model is defined, one can straight-forwardly apply any other extensions of the EM algorithm. We explore, in Chapter 6, the use of semi-supervised extension of the EM to integrate additional data biological and improve clusterings of gene expression time-courses.

Table 3.2: *We present the mean values for data with 2, 3, 4, 5 and 6 components (top to bottom) against the number of components of the estimated mixture for corrected Rand, extended corrected Rand and BIC . For all indices, the maximum values (in bold) indicate the predicted number of components and the line preceding the indices values states the correct number of components.*

no clusters	**2**	3	4	5	6	7	8	9	10
CR	**0.37**	0.30	0.24	0.20	0.17	0.15	0.14	0.13	0.12
ECR	**0.21**	0.15	0.12	0.09	0.07	0.06	0.05	0.04	0.04
-BIC	**-4.74**	-4.75	-4.77	-4.79	-4.81	-4.84	-4.86	-4.88	-4.91
no clusters	2	**3**	4	5	6	7	8	9	10
CR	0.33	**0.52**	0.48	0.43	0.40	0.36	0.33	0.30	0.30
ECR	0.31	**0.39**	0.35	0.30	0.27	0.23	0.21	0.19	0.18
-BIC	-5.74	**-5.49**	-5.50	-5.52	-5.54	-5.56	-5.58	-5.60	-5.63
no. clusters	2	3	**4**	5	6	7	8	9	10
CR	0.34	0.50	**0.66**	0.66	0.64	0.59	0.57	0.58	0.58
ECR	0.43	0.45	**0.58**	0.55	0.52	0.48	0.45	0.44	0.44
-BIC	-9.03	-8.97	-8.77	**-8.72**	-8.74	-8.76	-8.78	-8.80	-8.80
no. clusters	2	3	4	**5**	6	7	8	9	10
CR	0.33	0.46	0.48	0.53	0.54	0.55	**0.56**	0.55	0.56
ECR	0.33	0.42	0.44	**0.47**	0.478	0.46	0.46	0.44	0.44
-BIC	-1.00	-0.99	-0.98	-0.97	-0.97	-0.97	**-0.96**	-0.97	-0.96
no. clusters	2	3	4	5	**6**	7	8	9	10
CR	0.21	0.44	0.48	0.54	0.56	0.56	**0.57**	0.55	0.56
ECR	0.18	0.44	0.48	0.51	**0.52**	0.52	0.52	0.50	0.50
-BIC	-7.34	-6.79	-6.74	-6.71	-6.71	-6.70	-6.70	-6.68	**-6.58**

Chapter 4

Analysis of Gene Expression Time Courses

The analysis of gene expression over the course of time is an important step to understanding function and regulatory roles of genes [11]. For example, during the cell cycle, we can find groups of genes that are only over-expressed in a particular phase. From this information, one can try to infer the gene function by relating it to genes with known function and similar expression profiles in a “guilty by association” approach [71]. Another alternative is to explore the promoter sequence of a group of co-expressed genes and search for common patterns of transcription factor binding sites in the upstream region of those genes [201]. Furthermore, by searching for genes (or groups) with similar expression profiles patterns, but with distinct time of change in gene expression levels, it is possible to explore regulatory roles [78]. For instance, earlier “activated genes” could regulate genes with similar expression patterns with a later “activation” time. A first step towards all these analyses is to find co-expressed genes, or groups of genes that display a similar expression profile over the course of time.

In this context, initial work was based on clustering methods that assume independence between expression values of distinct time points, for example, hierarchical clustering [71, 82], k-means clustering [212] and singular value decomposition [6]. However, in a temporal setting, the expression value in a time point is dependent on values of preceding time points. In contrast to these earlier studies, temporal models were applied to gene expression time courses, such as cubic splines [12, 138] and autoregressive curves [172]. These methods are often robust to noise found in time course experiments, such as noise in a single time point or expression profiles showing a slower rate of expression. Furthermore, these methods can make use of information relevant to time courses such as sampling time and periodicity.

As one of the main contribution of this thesis, we propose in this chapter a hidden Markov model (HMM) with a linear topology that is suitable for modeling time-dependent sequences such as a time courses. These models represent co-regulated genes with a similar prototypical behavior, or the same sequence of expression level changes, in an asynchronous manner. By asynchronous, we mean that the HMM captures time courses with the same events of expression changes at possibly distinct time points. Nevertheless, synchronous groups and their time of expression changes can be latter inferred from a model by the analysis of the most probable state path of a time course in the HMM. Therefore, it is possible to build a partial order on synchronous groups of genes all displaying the same prototypical expression pattern but with distinct time of expression changes.

Two main applications of linear HMMs are presented here. The first is an iterative graphical tool that allows an user to query a set of gene expression time courses for those displaying a specific prototypical behavior. This tool allows the user to build a linear HMM interactively and to explore a given gene expression data set for prototypical patterns of interest. A second, and more intricate application, is to estimate a mixture of linear HMMs to find groups of genes with the same prototypical expression patterns within a given data set.

This chapter is organized as follows. First, we describe related work in Section 4.1. Then, we give a brief definition of HMMs (Section 4.2) and the specific HMM topology employed here. In Section 4.3, we describe the application for querying gene expression data with a linear HMM, while in Section 4.4 we introduce a method for finding groups of genes with a mixture of HMMs. In Section 4.5, we present the evaluation of our method with two gene expression data sets. We present final remarks and future directions in Chapter 7.

4.1 Related Work

There are several computational tasks related to the analysis of gene expression time courses. These can be divided in three levels: (1) pre-processing methods, which are responsible for tasks such as microarray data normalization; (2) exploratory methods, which search for genes differentially expressed or for groups of co-regulated genes; and (3) network-based methods, which try to reconstruct regulatory (or metabolic) networks from genes (or gene products). For methods in (1) and (2), it is a common strategy to take the temporal nature of time courses into account. Next, we will give an overview of the most relevant methods in the exploratory level, which is the category our method belongs to. More complete reviews of methods in all these analyses levels can be found in [8, 11], and methods for normalization and differential expression in [29, 203].

There are few approaches concerning the detection of differentially expressed genes from time course gene expression. For example, [206] extended the significant analysis method in [219] to take temporal dependencies into account. In [208] an Empirical Bayes approach was presented to detect differentially expressed genes when replications of the time course measurements are present.

One interesting variation of the problem described in the previous paragraph is the detection of genes displaying different temporal expression patterns in time courses measured under distinct conditions, e.g., experiments with two (or more) time courses, each one measuring a particular cell applied to a distinct treatment or an environmental condition. Genes differentially expressed in one or more time courses should be biologically relevant to the treatment or condition analyzed. In [13], for instance, data from time courses from yeast cell cycle of a wild type and after FKh1 or FKH2 knockout were studied. There, B-splines were used to model the temporal patterns. Analysis of promoter regions confirmed that the detected differentially expressed genes were related to the knockout factors. A similar problem was approached in [235], where gene expression time courses of mice with distinct oxidative stress and age were investigated. Their method was based on a HMM with linear topology and Gamma distributions as state emissions. Using an Empirical Bayes approach [36], the authors could obtain p-values concerning the significance of these differences.

One particular type of time course data, which has received great attention in the literature, is gene expression time courses of the cell cycle, which is the process through an eukariotic cell duplicates into two genetically identical cells. The cell cycle consists of four phases: G1 phase, where the cell starts to grow and prepare for DNA replication; S phase, where DNA replication occurs; G2 phase, where microtubules are produced; and, finally, M phase, where nuclear and cytoplasmic division occurs [4]. For these experiments, a particular population of cells is arrested at a cell cycle phase (usually G1). After the release of the cells to start the cell cycle, the expression is measured over the course of two to three cell cycles. Examples of such data sets are found in [42, 201] for yeast and in [226] for human HeLa cells. In such data sets, some time courses have a periodic behavior usually displaying a cosine type of gene expression pattern, with an over-expression peak at one particular phase at each cell cycle (see Figure 4.11 for an example of such profiles). One particular effect of the arresting protocol is that the individual cells will have distinct cell cycle periods. Thus they will desynchronize, and the periodic signal deteriorates with time. Also, such data sets have usually more than 30 time points, a number far larger than most other time course experiments [73].

Such cell cycle based experiments pose a number of interesting methodological questions. In [1], a method for aligning pairs of time courses with a dynamic programming algorithm was proposed. Such a method finds pairs of genes, which have a similar expression pattern, but distinct phases. Pairs displaying such time-lags (or shift in expression patterns) are potential candidates for regulatory relationships. A more extensive study was performed in [78], where methods for shift and phase detections were proposed, so that distinct cell cycle experiments could be integrated in one.

One typical application in cell cycle data sets is the detection of transcripts displaying periodic behavior, as well as at which time point these periodic transcripts peak. See [58], for a good review of methods and comparison on popular data sets. Interestingly, [58] found that a simple method based on Fourrier analysis [201] was significantly better than all more sophisticated computational approaches. Also, a recent work has shown that the choice of the background model for performing the statistical tests has a great impact on the detection of periodic expressed genes [81].

In [71], the problem of finding functional modules of genes with the use of clustering methods was first proposed. This study is restricted to the application of simple methods, such as k-means and hierarchical clustering with classical distance measures such as Euclidean distance or the Pearson correlation coefficient. Its main methodological contribution was the proposal of a heat-map style plot for displaying groups of co-regulated genes, also known as red and green plots (see Figure 5.5 for an example of such plots). More recently, [6] proposed the use of singular value decomposition and [232] the use of mixture of multivariate Gaussians. In the latter, a wide comparison on the effect of data normalization strategies and the choice of the covariance matrix on the results was performed in a cell cycle data set. It was shown that full covariances matrices were prone to over-fitting, and that standardization of the time courses improved the results. One alternative to the use of classical distance measures is the use of similarity measures based on mutual information. In contrast to Pearson correlation and Euclidean distance, Mutual information is able to capture non-linear correlations (or dependencies) [33, 204]. Nevertheless, these methods are based on expression data sets with large number of samples, i.e., more than > 100 biological observations, that is rarely the case in gene expression studies.

None of the previous reviewed methodologies take the temporal nature of time courses into consideration for finding co-regulated genes. In particular, classical clustering methods, such as k-means and hierarchical clustering rely on distance functions between expression profiles, such as correlation or Euclidean distance, which neglect the temporal nature of time-courses. This shortcoming was first approached in [172], where auto-correlation models based on a low-order linear regression was proposed. This clustering method works in an agglomerate Bayesian fashion. It relies on several heuristics for the choice of parameters and finding the number of clusters. Also, another shortcoming of this method is the modeling of only low order dependencies, which will fail in modeling correlations in cell cycle data sets, where periodicity occurs after 8 or more time points. Simultaneously [12] and [138] proposed model-based clustering approaches using cubic splines [12] and B-splines [138]. Both methods require the specification of several parameters for the splines and take advantage of time courses with a large number of time points. The use of a model called hidden phase model (HPM) as a prior on a mixture of multivariate Gaussians was proposed in [30]. The HPM model, which can be seen as an extension to an HMM, allows the user to include preference towards component models with a particular type of temporal pattern, e.g., cyclic patterns, increase in expression, or decrease in expression. The method displayed good performance in recovering clusters of periodically expressed genes, even when a low number of time courses were present.

The method proposed in this chapter, mixture of linear HMMs, has a similar motivation as the methods described in the previous paragraph [12, 138, 172]. As in those methods, linear HMMs makes use of temporal dependencies for modeling groups of co-expressed genes. The main distinction of our method to those is the capability of modeling groups of genes with similar expression patterns, but with asynchronous changes in expression. As each of these approaches are based on distinct assumptions for modeling time dependencies, none of them is likely to be the overall best choice. For comparing these methods with our approach, we perform an empirical evaluation using data from yeast cell cycle data set described in Section 4.5.1.

In [234], it was investigated how repeated microarray measurements could be integrated in the cluster analysis. The rationale behind their approach is that measurements with low replicate variability should be more reliable than measurements with high variability. Consequently, they have a higher weight on the clustering procedures. They applied this idea on several model-based clustering methods. The study found that an infinite mixture of multivariate Gaussians obtained the most favorable results. However, the work did not explore any method modeling the temporal dependencies. Even though our proposal do not explore replicate estimates, an extension of our method using the proposal of [234] is straightforward.

Recently, attention has been given to the fact that time courses data sets have usually few time points [73, 150]. Indeed, 80% of the data sets in Gene Expression Omnibus [69] have less the 8 time points. This is of crucial importance, as most of the model-based clustering methods described before [12, 138, 172] will suffer from over-fitting with such data sets. In [150], a fuzzy clustering method for short and unevenly sampled time courses was proposed. This method takes into account first-order dependencies and the sampling of time points. They showed for data set with seven time points that the method was superior to k-means and hierarchical clustering. In [73], the authors proposed a method that performs a greedy search over the set of possible expression patterns. It finds the patterns that are significantly distinct from the others. The authors showed that the method had favorable results in relation to k-means and CAGED [172] for a time course data set with 5 time points. Note that even thought this also poses a problem the mixture of HMMs, which also take advantages of larger time courses, this point could be tackled in our approach with the use of structural learning techniques favoring HMMs with fewer stages.

An interesting problem is the integration of additional biological data in the detection of groups of co-regulated genes. In [101], authors explored the joint analysis of gene expression and sequences from promoter regions. The main idea is to inspect if co-regulated genes also shared similar transcription factor binding sites. They defined a method based on the EM and Gibbs sampling for performing a clustering with both expression and sequence data. The same problem was approached with an EM method in [194] with the use of discriminative position weights matrices as models for hits of transcription factor binding sites. However, none of these approaches used models taking temporal dependencies into account. One exception is the work of [231], which combined location analysis data (also known as Chip-on-chip [128]) with time courses from cell cycle. They could find groups of co-regulated genes displaying time lagged expression pattern in relation to the expression profile of transcription factors. Data from location analysis confirmed regulatory roles between some of these transcription factors and groups of co-expressed genes. In [74], the authors applied an input-output HMM to combine time course gene expression with location analysis data in order to detect groups of genes, which are targets of a given transcription factor and have a similar co-expression profile. They analyzed short time courses after treatment of yeast to stress conditions, and could detect novel putative regulatory roles.

Recently, there has been a great deal of data sets with multiple (usually short) time courses measured over distinct gene knockouts [236], treatments [174], patients [225], or environmental conditions [82]. Such data present new methodological challenges not addressed before. In [113], time courses from multiple sclerosis patients after particular treatments were analyzed. Their Bayesian framework could detect gene responses, which were specific to either the treatment type or to specific responses of a particular patient. In [174], time courses of Arabidopsis after several distinct treatments were analyzed. The work aimed to find genes that display a treatment specific time-lag in their expression profiles in relation to the expression profile of known transcription factors. The groups of time-courses with time-lag were found with the use of a covariance index and an heuristic based on the Gap statistic. In [197], a similar problem was approached with the use of a graphical model. They used a set of known transcription factors and gene target relations to estimate parameters and time lags. Then, their model was used to predict unknown regulatory relationships. One main difficulty of this problem is the fact that the time lag is dependent of the biological condition and transcription factor, which requires the estimation of a large number of free parameters. However, both [174, 197] showed that their methods, by using multiple time courses, were superior is detecting regulatory roles of genes from expression time-lag than methods based on single time course data as [1, 78].

4.2 Hidden Markov Models

A hidden Markov model (HMM) is a probabilistic model composed of a Markov chain with M discrete states and emission probability density functions (pdf) assigned to each state. At a given time point, a HMM is at a particular unknown state and it emits a symbol in accordance to the density function assigned to that state. More formally, given a continuous random variable $X = (X_1, ..., X_t, ..., X_L)$ representing the emitted symbols, and a discrete hidden variable $Q = (Q_1, ..., Q_t, ..., Q_L)$. For an given observation $x = (x_1, ..., x_t, ..., x_L)$ from X, we have a corresponding hidden sequence path $q = (q_1, ..., q_t, ..., q_L)$, where $q_t \in \{1, ..., M\}$ represents the state emitting x_t. A HMM allows a computational efficient approximation of the joint densities $p(x, q)$ for observations x and q. There are two main independence assumptions regarding HMMs: (1) the probability to reach a state t depends only on the previous state $(t-1)$ [1]

$$p(q_t|q_1, ..., q_{t-1}) = p(q_t|q_{t-1}), \tag{4.1}$$

and (2) the density function of emitting x_t depends only on the current state t

$$p(x_t|q_1, ..., q_t) = p(x_t|q_t). \tag{4.2}$$

We can represent the probabilities in Eq. 4.1 by a transition matrix $A = \{a_{uv}\}$ for $1 \leq u \leq$

[1] We only consider here, HMMs with first-order dependencies.

M and $1 \leq v \leq M$, where a_{uv} is equal to the probability of going from state u to state v, i.e., $p(q_t = v|q_{t-1} = u)$, given that $\sum_{v=1}^{M} a_{uv} = 1$ and $a_{uv} \geq 0$ for $1 \leq u \leq M$. The initial state probabilities $p(q_1 = u) = \pi_u$ are represented by a vector $\pi = (\pi_1, ..., \pi_u, ..., \pi_M)$. In our problem, which deals with gene expression, the emission variables are continuous, and univariate Gaussian densities are used as emission function on the states

$$p(x_t|q_t = u) = p_u(x_t|\mu_u, \sigma_u^2) = \frac{1}{\sqrt{2\pi\sigma_u^2}} \exp -\frac{(x_t - \mu_u)^2}{2\sigma_u^2}, \tag{4.3}$$

where $p_u(x_t|\mu_u, \sigma_u^2)$ is the probability density function (pdf) associated with the uth state, and μ_u, σ_u^2 are the pdf parameters. Hence, a HMM with M states is parameterized by the vector $\theta = (A, ((\mu_1, ..., \mu_M), (\sigma_1^2, ..., \sigma_M^2)), \pi)$.

Given an observation x and the corresponding sequence of visited states, the joint distribution can be defined as follows

$$\begin{aligned} p(x, q|\theta) &= p(x|q, \theta)p(q|\theta) && (4.4) \\ &= p(q_1)p(x_1|q_1) \prod_{t=2}^{L} p(q_t|q_{t-1})p(x_t|q_t) && (4.5) \\ &= \pi_{q_1} f_{q_1}(x_1|\mu_{q_1}, \sigma_{q_1}^2) \prod_{t=2}^{L} a_{q_{t-1},q_t} f_{q_t}(x_t|\mu_{q_t}, \sigma_{q_t}^2) && (4.6) \end{aligned}$$

For a given HMM defined by the parameters θ, one first natural question, following [169], is how to compute the likelihood of an observation x, or

$$p(x|\theta) = \sum_{q \in \mathcal{Q}} p(x, q|\theta), \tag{4.7}$$

where $\mathcal{Q}$ is the set of all possible state sequences q. A brute force calculation of the previous equation requires the infeasible evaluation of M^L sequences. Nevertheless, a technique based on dynamic programming, called forward-backward algorithm allows us to compute Eq. 4.7 in $O(ML)$ time [20].

The second problem is the maximum likelihood estimation of a HMM from a data set $\mathbf{X}$ with N observations, where x_i is the ith observation from $\mathbf{X}$, that is, finding

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{N} p(x_i|\theta). \tag{4.8}$$

This problem can be solved with the Baum-Welch algorithm [21], which is a specific application of the EM algorithm for HMMs.

The last task is the decoding problem, or for a given observation x and a HMM with

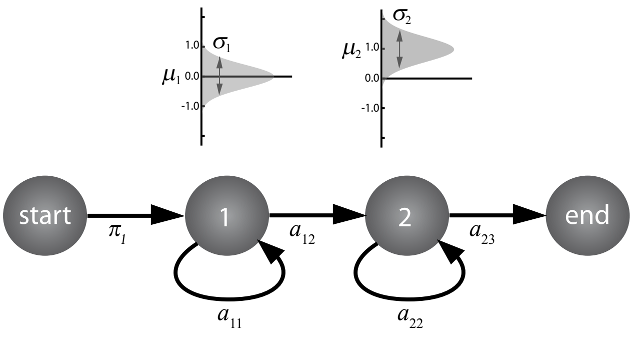


Figure 4.1: *Example of a linear HMM with two emitting states (states 1 and 2), an initial state and an end state. Given the linear topology, all sequences will always visit, respectively, the start state, 1, 2 and end state. The state 1 models expression values near zero, while the state 2 models expression values around one. Thus, this lHMM models time courses up-regulated at some time point. The values of the self-transition parameters a_{11} and a_{22} will define the expected duration that a state will be visited. For example, if we set $a_{11} > a_{22}$, the model will give a higher likelihood for expression profiles with late up-regulation patterns.*

parameters θ, find the sequence of visited states q that has maximum likelihood,

$$q^* = \arg\max_q p(x, q|\theta). \tag{4.9}$$

This problem can be solved with the Viterbi algorithm [222]. For more details, proofs and other extensions we refer to [121, 169].

4.2.1 Linear HMM and Time Courses

We use a particular linear topology for modeling gene expression time courses. In this topology, a given state only has a self-transition and a transition to the next state. We include in the beginning (and end) of each chain a special start state (and end state). Given that the time courses are normalized by taking the logarithm in relation to a control experiment (an usual procedure in gene expression time course analysis [71, 201]), we interpret that an expression value near zero means expression close to background level, above zero over-expression (or up-regulation) and below zero under-expression (or down-regulation). For example, a linear HMM—lHMM for short—with two emitting states: the first ones with a mean emission of zero and the second one with a mean emission of one, we model time courses displaying an up-regulation prototypical behavior (Figure 4.1).

More formally, the random variable $X = (X_1, ..., X_t, ..., X_L, X_{L+1})$ represents the gene expression time courses, where $x = (x_1, ..., x_t, ..., x_L, x_{L+1})$ is an observed time course, x_t is the expression value at time-point t and x_{L+1} is set to the special ending symbol “$\ltimes$”. As described in the Section 4.2, a HMM is parameterized by a transition matrix

A, emission parameters $(\mu_1, ..., \mu_M), (\sigma_1^2, ..., \sigma_M^2)$ and initial probability vector π. As we restrict the topology to a linear chain of states, we can adopt a simple notation for the linear HMMs. For a given state u, we only need to define the self-transition probability a_{uu}, as $a_{u(u+1)} = 1 - a_{uu}$ and all other transitions from state u are set to zero. A more intuitive representation is the expected duration length of a state. For a single state, the length distribution respects a geometrical distribution [67], and has an expected length of

$$d_u = \frac{1}{1 - a_{uu}}. \tag{4.10}$$

This parameter represents how many time steps the observation sequence is expected to spend in state u. Accordingly, we can obtain the transition probability a_{uu} from a given duration,

$$a_{uu} = 1 - \frac{1}{d_u}. \tag{4.11}$$

We also constrain the duration parameters so that

$$\sum_{u=1}^{L} d_u = L, \tag{4.12}$$

for assuring that the expected length of reaching the end state matches the time course length L.

We include an end state $M + 1$ with a self-transition set to one and an emission density fixed to a special ending symbol. The initial probability vector π can be ignored, since we will have $\pi_1 = 1$ and $\pi_u = 0$ for $2 \leq u \leq M$. Given that all observations x have an end symbol at x_{L+1} and $M \leq L + 1$, all sequence paths not visiting each state at least once have zero likelihood. Consequently, such sequence paths are ignored in the parameter estimation performed by the Baum-Welch algorithm. Furthermore, we have the parameters μ_u and σ_u^2 for the emission function. Finally, a given state is parameterized by the triple (d_u, μ_u, σ_u^2), and a lHMM by $\theta^l = ((d_1, \mu_1, \sigma_1^2), ..., (d_M, \mu_M, \sigma_M^2))$.

Such linear HMMs are able to model time courses displaying several prototypical behaviors. For example, in Figures 4.1 and 4.2, we depict a lHMM that models profiles with distinct prototypical expression behaviors (up-regulation, up and down-regulation, and so on). One way to interpret the temporal behavior modeled by an lHMM is the following: each state represents a particular level of expression specified by μ_u, where a certain level of error (encoded by σ_u^2) is allowed[2], and the time course has an expected length of d_u of staying in this particular expression level. This can be interpreted as a "bounding box" specifying the expected expression of time courses, as depicted in Figure 4.3. Even though a lHMM models the expected duration of visiting states, it allows some flexibility concern-

[2] In a normal probability density function the interval $[\mu - \sigma, \mu + \sigma]$ defines the region around the mean containing 68% of the density. In the figures depicting the lHMMs, the lengths of the σ are only used for illustration purposes and do not strictly define the exact proportion of the densities.

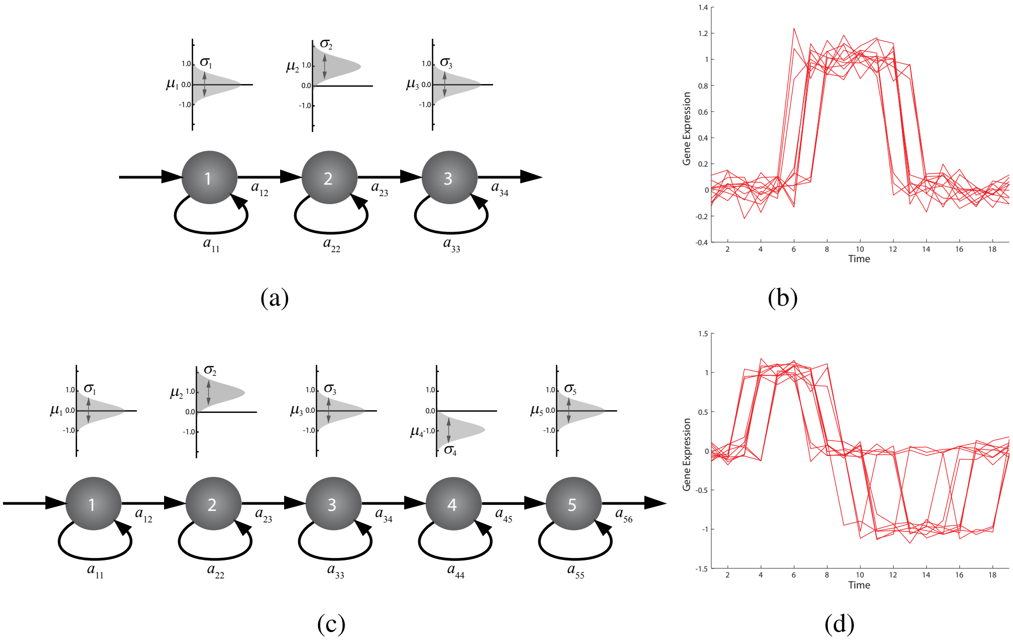


Figure 4.2: *Example of linear HMMs modeling up-regulation (a) and up- and down-regulation patterns (c). On (b) and (d), we display time courses with 19 time points observations from these linear HMMs. The duration of states are all set so that time courses will visit each state for the same number of time points. For simplicity, we do not depict start and end states from models.*

ing the time of changes in expression levels (or transitions between distinct states). This characteristic makes it possible to model asynchronous time courses showing a similar prototypical behavior. An inspection of the Viterbi path q for a given model θ^l and time course x (Eq. 4.9) indicates the most probable sequence of transitions and can be used to recover synchronous groups of time courses. For example, in Figure 4.4, the Viterbi paths indicate the existence of three synchronous groups of up-regulated genes: up-regulation event from the time point 9 to 10 (green), from time point 10 to 11 (red) and from time point 11 to 12 (blue).

Periodic Gene Expression Time Courses. We can extend the linear HMM topology to model some special applications in gene expression time course analysis. For example, in periodic time courses, expression is measured during two or three cell cycles, and interesting genes should show a periodic or cyclic behavior [42, 201]. By including a transition probability from state M to state 1 and adjusting all d_u to match the expected cycle length, we obtain a HMM that models periodicity.

Formally, we need to specify the expected cycle length R and the number of cycles measured S. Both values are usually known from the experimental setting used in the gene expression data acquisition. The duration of states should be chosen such that $\sum_{u=1}^{M} d_u = R$. For the last emitting state L, a_{LL} is defined as usual (Eq. 4.11), and the other transitions

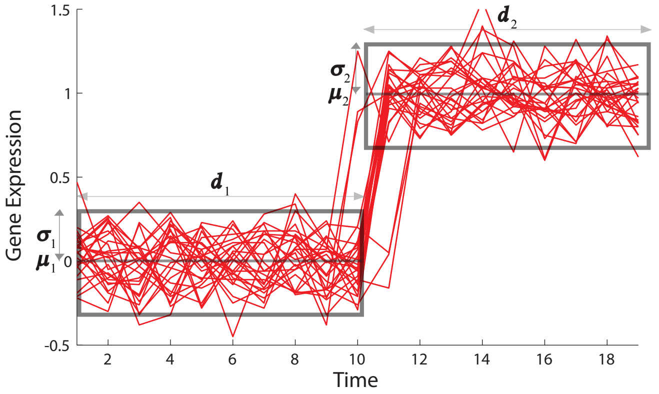


Figure 4.3: *Example of up-regulated time courses with 19 time points sampled from the linear HMM depicted in Figure 4.1. For that topology, we set the parameters of state 1 to* $(d_1, \mu_1, \sigma_1^2) = (10, 0.0, 0.1)$ *and of state 2 to* $(d_2, \mu_2, \sigma_2^2) = (9.0, 1.0, 0.1)$. *The duration value* d_1*=10 corresponds to setting* $a_{11} = 0.9$ *and* $a_{12} = 0.1$. *The gray boxes depict the overall expression pattern modeled by each state. For example, in state 1,* $\mu_1 = 0$ *defines the box location in the* y*-axis (or the mean expression value),* σ_1 *the width of the box in the* y*-axis (or the allowed variation around the mean expression value) and the parameter* d_1 *the length of the box in the* x*-axis (or the duration a time course will stay at a particular expression value). Note that the boxes define only the expected expression behaviors; time courses violating these bounds are allowed as well as long as the overall expression pattern is matched. Also, asynchronicity in the time courses is allowed, as we have time courses where the up-regulation event occurred a few points before or after the expected transition time point (t=10).*

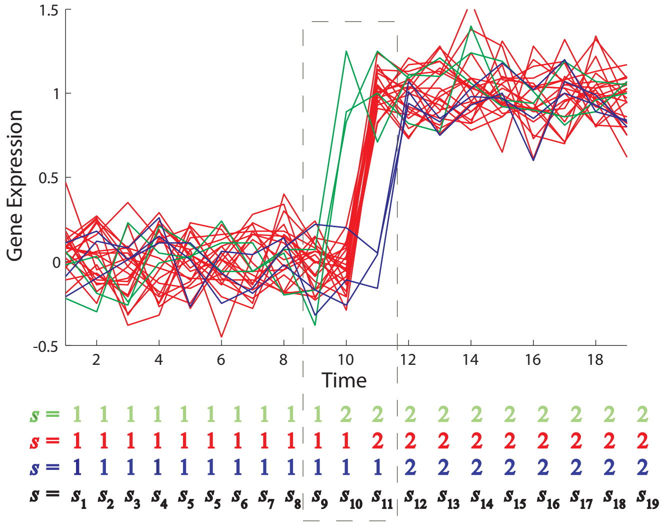


Figure 4.4: *Example of the Viterbi paths of the time courses from Figure 4.3 given the model depicted in Figure 4.1. These time courses have one of the three Viterbi paths: time courses in green have a transition from state 1 to state 2 at time point s_9 to s_{10}, time courses in red at s_{10} to s_{11} and time courses in blue from s_{11} to s_{12}. The Viterbi path sequences are depicted below the graph with the color of the corresponding group of time courses.*

are specified as follows

$$a_{L1} = \frac{(1 - a_{LL})(S - 1)}{S} \tag{4.13}$$

and

$$a_{L(L+1)} = \frac{1 - a_{LL}}{S}. \tag{4.14}$$

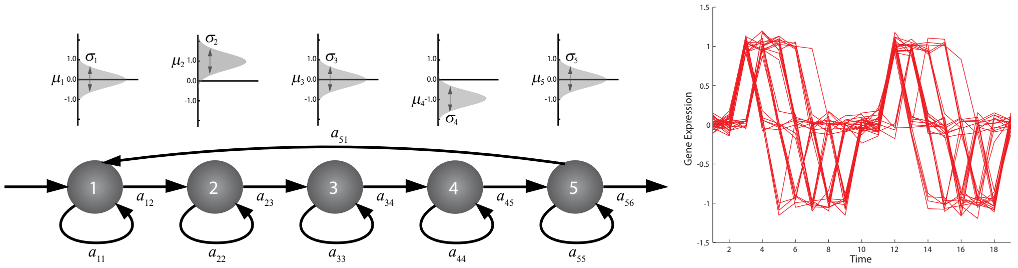
Support of Missing Data. Another useful extension, in the context of gene expression, is the support of missing data. Formally, we define a special symbol (Nan) and extend the space of variable X to $\mathbb{R} \cup \{\text{Nan}\}$. We then redefine the emission density from Eq. 4.3 as follows

$$p(x_t|q_t = u) = (1 - \phi) * f_u(x_t|\mu_u, \sigma_u^2) + \phi * 1(x_t = \text{Nan}), \tag{4.15}$$

where ϕ represents the proportion of missing observations. For given data $\mathbf{X}$, we can derive ϕ by measuring the percentage of missing symbols in $\mathbf{X}$

$$\hat{\phi} = \frac{\#\{x_{iu} = \text{Nan}|x_{iu}, 1 \leq i \leq N, 1 \leq u \leq L\}}{L \cdot N}. \tag{4.16}$$

Linear HMMs and Multivariate Gaussians. There is a close relation between the linear HMMs presented in this chapter and the multivariate Gaussians discussed in Section 2.3.3. In a lHMM with one state per time-point, which implies $M = L$ and $a_{uu} = 0$ for all



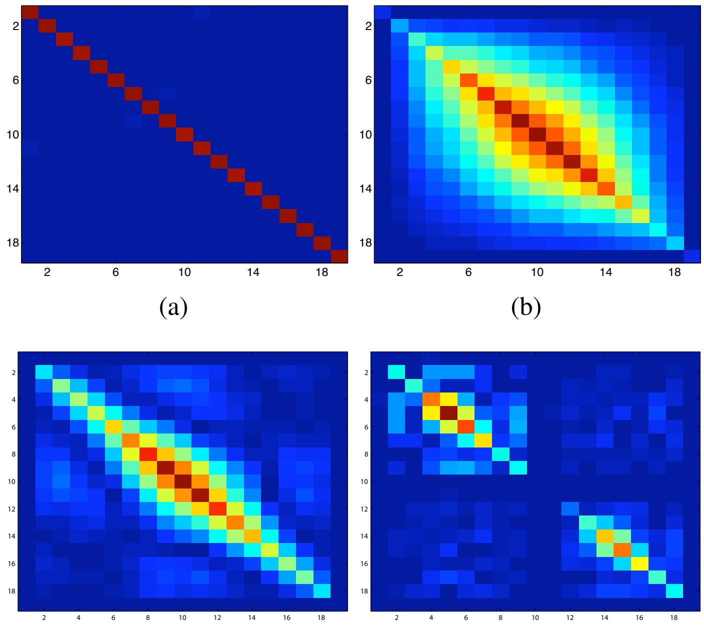
(a) (b)

Figure 4.5: *Example of the extension of the linear HMM defined in Figure 4.2 (a) for modeling periodicity (a). The model requires simply the addition of transition* a_{51} *and re-parameterization of all transition values. We display time courses with* $L = 19$ *sampled from this model (b).*

$1 \leq u \leq L$, the linear model is equivalent to a multivariate Gaussian with mean vector $(\mu_1, \ldots, \mu_L)$ and covariance matrix $diag(\sigma_1, \ldots, \sigma_L)$ (see Section 2.3.3). Typically, the number of states of a lHMM respects $M < L$ and there is no simple analytical way to parameterize a multivariate Gaussian from the parameters of a lHMM in this case. The reason for this is that the probability of visiting a state u is dependent on each time course, which would require the computation of posterior probabilities of being at a given state $(p(s_l = u|x)$ [67]).

One way to evaluate the type of covariance structures modeled by the linear HMMs is to sample data from a linear HMM, and estimate the empirical covariance matrix from it. In order to do so, we sampled 1,000 sequences with length $L = 19$ from the models depicted in Figures 4.1, 4.2 and 4.5, and estimated the covariance matrices as defined in Eq.2.28. We also sampled data from a linear HMM with 19 states, where $\mu_u = 0$ for $1 \leq u \leq 10$, $\mu_u = 1$ for $11 \leq u \leq 19$, $d_u = 1$ and $\sigma_u = 0.01$ for $1 \leq u \leq 19$. This lHMM, which measures time courses up-regulated from time point 10 to 11, corresponds to the case with $L = M$ and it is equivalent to a multivariate Gaussian with diagonal covariance matrix. The covariance matrices of four lHMMs are displayed in Figure 4.6. As expected, the covariance matrix derived from the lHMM with $L = M$ has values near zero (dark blue) in all off-diagonal entries. In other words, there is no dependence between time points. For the lHMM modeling up-regulation—Figure 4.2 (a)—the covariance matrix indicates correlations between consecutive time points (entries near the diagonal). The covariance matrix from the lHMM modeling up- and down-regulation—Figure 4.2(c)—has a more intricate correlation pattern, displaying correlation between consecutive time points, but also a negative correlation with time points 7 steps apart. This is a consequence of the down-regulation event, which happens later in these time courses. In the cyclic lHMM, the block of the matrix σ_{uv} for $1 \leq u \leq 9$ and $1 \leq v \leq 9$ (also σ_{uv} for $11 \leq u \leq 19$ and $11 \leq v \leq 19$) have a courser but similar covariance structure to the lHMM modeling up- and down-regulation (Figure 4.2 (c)), and all other entries have values similar to zero.

If we inspect only time courses following similar Viterbi paths, we get a further under-



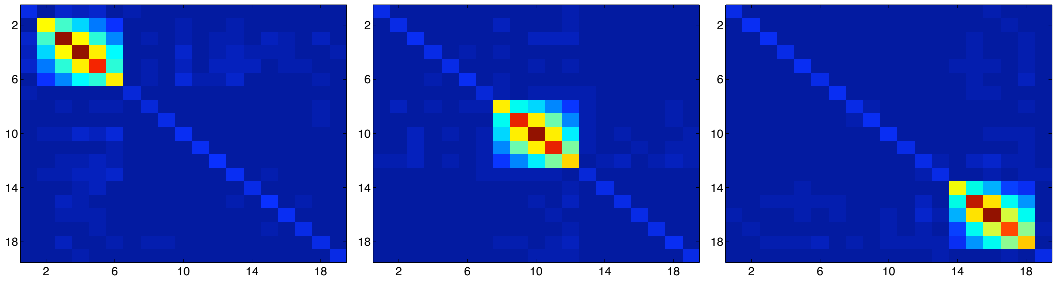
(a) (b)

(c) (d)

Figure 4.6: *We depict heat map plots of the absolute value of the empirical covariance of data sampled from a 19 state lHMM (a), from the up-regulation lHMM depicted in Figure 4.1 (b), up- and down-regulation lHMM depicted in Figure 4.2 bottom (c) and the cyclic lHMM depicted in Figure 4.5 (d). Dark blue entries indicates zero values, while red indicates positive (or negative) values.*

standing of the covariance structure modeled by the lHMM. For example, if we look at the simulated data from the lHMM depicted in Figure 4.1, and divide it into three groups: (1) time courses with the up-regulation event before time point 7; (2) time courses with up-regulation between 7 and 12; and (3) time courses up-regulated after time point 12. The covariance matrices of these groups are depicted in Figure 4.7. The resulting covariance matrices are decompositions of the original matrix—Figure 4.6 (b), where the covariances between consecutive time points are restricted now to earlier (Figure 4.7 (a)), middle (Figure 4.7 (b)) or late (Figure 4.7 (c)) time points.

The linear HMMs allow modeling of dependencies between subsequent time points. Moreover, depending on the topology, longer term dependencies are also captured, e.g., up- and down-regulation lHMM (Figure 4.2 (c)). One very important characteristic is the few number of free parameters necessary for the linear HMMs. A lHMM requires the specification of $(3 \times M - 1)$ free parameters. In contrast, a multivariate Gaussian with diagonal matrix has $(2{\times}L)$ and a multivariate Gaussian with full covariance matrix $\left(2 \times L + \frac{L\times(L-1)}{2}\right)$ free parameters. In other words, lHMM will require a substantially fewer number of parameters than a multivariate Gaussian with full covariance matrix and, whenever $2M < 3L$, a lHMM has fewer parameters than a Gaussian with diagonal covariance matrix. Furthermore, the lHMM captures low order dependencies, whereas the Gaussian with diagonal covariance



(a) (b) (c)

Figure 4.7: *We depict heat map plots of the absolute value of the empirical covariance of data sampled from a 19 state lHMM from the up-regulation lHMM depicted in Figure 4.1, where the up-regulation event occurred before time point 7 (a), between 7 and 12 (b), and time courses after time point 12 (c). Dark blue entries indicates zero values, while red indicates positive values.*

matrix assumes independence between time points. Of course, one could estimate the full covariance matrix, but as discussed in Section 2.3.4, such models could present over-fitting problems when used in mixture models. Thus, due to its characteristics, lHMMs are an ideal candidate for component models in a mixture model for analyzing gene expression time courses, as it requires few free parameters and is able to model temporal dependencies.

4.3 Querying of Time Courses

An initial exploratory analysis of gene expression time course can be performed by querying a particular data set for specific patterns of expression. This can be done by specifying a HMM topology and parameters manually and, later, ranking all time courses in $\mathbf{X}$ by their likelihood for that particular model (Eq. 4.7).

More formally, for a given linear HMM θ^l, a data set $\mathbf{X}$ and a stringency value $v < N$, a query can be described as follows

$$\mathbf{S} = \{x|\text{rank}\left(p(x|\theta^l)\right) \leq v, x \in \mathbf{X}\} \tag{4.17}$$

where $p(x|\theta^l)$ is the likelihood function defined as in Eq. 4.7, the function rank returns the rank of ordering x in relation to the likelihood, and $\mathbf{S}$ is the set of expression profiles, which have highest v likelihoods under the model θ^l.

As the knowledge discovery process in the analysis of biological data is human-centric, a high degree of interactivity is important in an initial analysis. We develop an interactive tool—the Graphical Query Language (GQL)—for querying data sets from gene expression time courses [53]. The tool allows the user to define a linear HMM model, tune its parameters and to define the query stringency (v) interactively. Modifications of the linear HMM parameters in the interface tool are simultaneously showed in the time courses

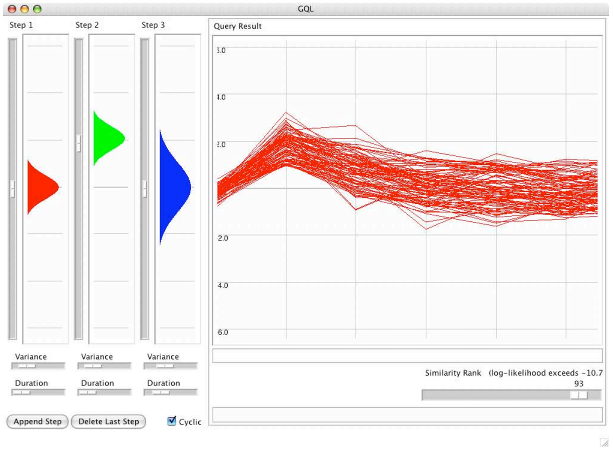


Figure 4.8: *The GQLQuery interface is divided into two main components: the left part is the model editor, where the user can view and change the model's parameters, and in the right part is the query result, where the queried time courses are displayed.*

panel, which plots the time courses on $\mathbf{S}$. The user can explore all possible prototypical patterns of expression presented in the data set. By scrolling the mouse over a desired time course, GQLQuery will depict the Viterbi path in the bar below the time course plots, as well as display information as the gene name. The interface also allows the user to create models for periodic time courses. All query results and generated models can be saved and used for further analysis.

4.4 Mixture of linear HMMs

In order to find groups of co-expressed time courses, we combine K linear HMMs in a mixture model. With such a model, we can find groups of expression patterns displaying a similar prototypical behavior within a gene expression data set. Formally,

$$p(x|\Theta) = \sum_{k=1}^{K} \alpha_k p_k(x|\theta^l_k), \tag{4.18}$$

where α_k is the mixture coefficient respecting $\alpha_k \geq 0$ and $\sum_{k=1}^{K} \alpha_k = 1$ (see Section 2.2), $p_k(x|\theta^l_k)$ is the density corresponding to the kth HMM as defined in Eq. 4.7, and $\Theta = (\alpha_1, ..., \alpha_K, \theta^l_1, ..., \theta^l_K)$.

We can estimate mixtures with the use the EM algorithm described in Section 2.3.1, by the inclusion of a missing variable Y, where $y \in \{1, ..., K\}$ indicates which mixture component produced the observation x. As previously discussed, the only necessary extension is the specification of the M-Step. In the M-Step, we need to obtain for a particular cluster assignment the maximum likelihood estimate of each of the models θ^l_k. While there is no analytic way for computing the maximum likelihood estimate for HMMs, we can apply the

Baum-Welch algorithm to (locally) maximize Eq. 4.8 in an iterative procedure.

Note that there is no need to extend the Baum-Welch algorithm for the specific topology of linear HMMs. All zeros entries in the transition matrix A, which are imposed by the linear topology, will remain equal to zero after the application of the Baum-Welch [121], and the topology will always be preserved. The parameter ϕ, related to the missing values, is kept fixed.

4.4.1 Model Initialization

The EM algorithm requires an initial model as starting point for the inference. This can be performed with random assignments, as described in Section 2.3.1. The lHMMs require the previous specification of few parameters, for example, the number of states M for each θ_k^l. Note that each component model can have a distinct number of states as long as $M \leq L$. Also, in the initialization of the duration parameters the property $\sum_{u=1}^{M} d_u = L$ should be respected, where L is the length of the time courses (see Section 4.2.1 for restrictions for the periodic extension of lHMMs). We will refer to this initialization method as the random model collection (`RMC`).

Another alternative is to use the results of other clustering methods as the initial assignment for the mixture [145]. Thus, we also consider the use of k-means for initializing the models, which we will reefer to as (`KMC`).

The inference of HMM topology can be solved for special topological cases with the use of priors and additional computational cost [61, 184, 205]. In a previous work [185], a collaborator proposed an adaptation of the Bayesian model-merging algorithm [205] to continuous emissions and linear HMMs. In this context, [185] takes advantage of the linear topology of the lHMM. It starts with a topologically unconstrained maximum likelihood model, which has one linear HMM per gene-expression time course x (or a branch), and one state per time-point in each branch, i.e., $L = M$. The method in [185] is based on merging states of the unconstrained model in two steps.

First, it merges states within the branches. The method identifies for each branch states whose merging decreases the likelihood the least. As the merging method assumes that variances are equal for all states, it only has to identify those successive states whose means are similar. A merging of components is performed until it reaches the desired number of states. Typical values range from a third to half the number of time-points. Second, the method merges the shrunken branches such that the loss of likelihood is minimal in each merging step with hierarchical clustering. For detailed description of the algorithm, which we will refer to as Bayesian Model Collection (`BMC`), see [185].

4.4.2 Viterbi Decomposition

The idea of the Viterbi decomposition is to find sub-groups of synchronous groups within genes following the same asynchronous prototypical patterns as modeled by a lHMM. An example on how the Viterbi path can be used in finding asynchronous groups is depicted in Figure 4.4. In that simple example, all time courses presented only three distinct paths. On real data and with a lHMM with more states, the number of distinct Viterbi paths is larger, and simple path enumeration would lead to a high number of asynchronous groups. One way to overcome this is to apply a clustering method for finding groups of similar Viterbi paths.

For a set of time courses x belonging to a cluster k, or $\mathbf{X_k} = \{x|y = k, x \in \mathbf{X}\}$, we measure the most probable Viterbi path q of each of the time courses $x \in \mathbf{X_k}$

$$\mathbf{Q_k} = \{q|q = \arg\max_q p(x, q|\theta), x \in \mathbf{X}_k\} \tag{4.19}$$

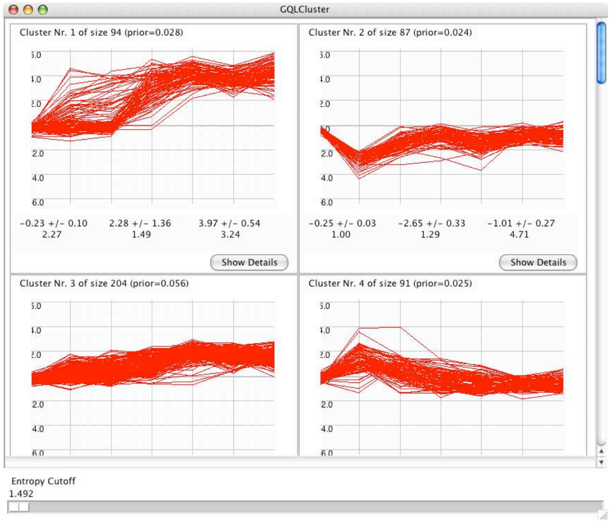
and look for groups of similar Viterbi sequences q_i.

In the approach proposed by a collaborator [132, 185] and used in the following analysis, the main assumption is that the sequence paths over one state will be the most significant in finding a group of synchronous time-courses. Take, for example, a group of cyclic time courses. Sorting these according to the time at which the first peak is reached is enough to produce subgroups of time courses with the same phase shift. Initially, the method searches the most appropriate state by calculating a Silhouette coefficient [178] for all states, and returns the one with lower Silhouette, i.e., the one with more compact and isolated sub-groups. After one state is selected, all possible unique paths are enumerated, and a greedy clustering method is performed by joining pairs of sub-groups leading to the highest increase in the Silhouette coefficient, until no more increase is possible. See [132, 185] for details of the method.

4.4.3 GQLCluster

All methods described in this section are implemented in the tool GQLCluster [53]. This software also includes some additional features useful in the analysis of time courses, such as methods for filtering gene expression data sets, an implementation of standard clustering methods such as k-means and hierarchical clustering, mixture of Gaussians and a semi-supervised approach described in [187].

After clustering by mixtures of lHMMs (or other clustering methods) has been carried out, GQLCluster offers several interactive tools for the analysis of the results. As a starting point, the graphical interface creates panels, which contain the time courses of each of the clusters/components (Figure 4.9). Then, for each cluster, it is possible to inspect the list of gene identifiers, which are linked to known web databases. In the mixture model methods, the time courses are assigned to the most likely model. The user can choose



GQLCluster

Cluster Nr. 1 of size 94 (prior=0.028)

-0.23 +/- 0.10 2.27 | 2.28 +/- 1.36 1.49 | 3.97 +/- 0.54 3.24

Cluster Nr. 2 of size 87 (prior=0.024)

-0.25 +/- 0.03 1.00 | -2.65 +/- 0.33 1.29 | -1.01 +/- 0.27 4.71

Cluster Nr. 3 of size 204 (prior=0.056)

Cluster Nr. 4 of size 91 (prior=0.025)

Entropy Cutoff
1.492

Figure 4.9: After estimation, GQL Cluster displays the time courses assigned to each cluster. The user can then perform a more detailed inspection of the modules, such as looking for gene annotation in known web databases, inspection for GO enrichment or computation of sub-groupings.

only genes that can be unambiguously assigned to one model by increasing the entropy cut-off threshold. A further refinement of the clusters can be obtained by the application of a Viterbi decomposition analysis, which finds sub-groups of synchronous time courses. All results, plots and estimated models can be saved for further analysis. A tutorial on the application, data and installation instructions can be found at [90].

4.5 Experiments

In order to illustrate the use of all the methods proposed in Section 4.2.1, we describe in the following the computational experiments based on two gene expression time course data sets. For benchmarking purposes, we make use of fully annotated data from yeast cell cycle. With this data set, we make a comparison with other clustering methods such as clustering with Splines [14], CAGED [172] and k-means. We also test the effects of initialization procedures: random model initialization (`RMC`), k-means initialization (`KMC`) and Bayesian initialization (`BMC`) (see Section 2.3.2 for details). Furthermore, we also test the use of the Viterbi decomposition (`VD`) for finding sub-groups in the clusters. For the second data set, which is based on a human HeLa cell cycle, we evaluate the use of the Viterbi decomposition and the entropy threshold proposed in Section 3.1. For the evaluation of the entropy threshold method, we make use of a specificity score based on the functional annotation of genes.

4.5.1 Data

Yeast Cell Cycle `YCC`**.** This data represents the expression level of around 6000 genes during two cell cycles from yeast measured in 17 time points [42]. As in [233], we used a subset of this data in the experiments. This data set, 5-phase criterion, abbreviated as `YCC`, contains 384 genes visually identified to peak at five distinct time points [42], each representing a distinct phase of cell cycle. The expression values of each gene are standardized. As showed in [233], such a procedure enhances the performance of model-based clustering methods, when the original data consists of intensity levels. This data set has class labels for the full range of genes, allowing the comparison of distinct methods.

HeLa Cell Cycle. We use published data from a time course experiment, in which the authors measured genome wide gene expression of synchronized HeLa cells [226]. We use the raw data from "doubly thymidine experiment three" as provided by the authors in the supplementary information. In this data set, HeLa cells, which have been arrested in S phase by a double thymidine block, are measured every hour from 0 to 46 hours. For reasons of comparison, we exclude clones showing missing values from further analysis. We obtain log ratios by dividing all expression values by the reference value (time point 0h) and taking the logarithm. Furthermore, the data is pre-processed by extracting all those genes with an absolute fold change of at least two in at least one time point. This results in a data set containing 2,272 expression time courses.

Specificity Score based on Gene Annotation. Gene Ontology (GO) describes genes in three distinct categories [9]: cellular component, molecular function and biological process. GO has a form of a directed acyclic graph (DAG), where the leaves are genes and the internal nodes are terms (or annotations) describing gene function, gene cellular localization or the biological process genes take part in. Leaves near the root describe very general processes, while nodes near the leaves describe specific ones. One should expect that the more unambiguous a cluster is, the more specific information it contains. Following this rationale, we evaluated the relation between ambiguity of gene clusters and the specificity (or level) of GO annotations (see Section 6.2.2 for a more detailed description of GO).

In order to find GO annotations related to a given subset of genes, we use an enrichment analysis [24], to look for annotation terms that are over-represented in this subset. The probability that this over-representation is not found by chance can be measured with the use of a hyper-geometric Fisher exact test [199]. The enrichment test returns for each cluster and GO term a p-value describing how statistically significant is a particular GO term for describing genes in a particular cluster[3]. See Appendix A for a description of the test.

[3] In the subsequent chapters, when a particular GO term is over-represented for a given cluster, we state GO Term X is enriched in cluster Y, or we found enrichment for GO Term X in cluster Y

Table 4.1: Results of the different methods on `YCC`*.*

Description	CR	Spec.	Sens.
HMM Mix. & `RMC`	0.330	0.488	0.475
HMM Clu. & `RMC`	0.331	0.474	0.490
Splines	0.362	0.494	0.516
HMM Clu. & `KMI`	0.380	0.534	0.502
HMM Clu. & `BMC`	0.388	0.520	0.543
HMM Mix. & `BMC`	0.390	0.531	0.527
HMM Mix. & `KMI`	0.391	0.538	0.517
HMM Clu. & `KMI` &`VD`	0.407	0.470	0.732
K-means	0.430	0.563	0.557
HMM Mix. & `KMI` &`VD`	0.432	0.502	0.718
HMM Clu. & `RMC` &`VD`	0.454	0.534	0.672
HMM Mix. & `RMC` & `VD`	0.458	0.540	0.664
HMM Clu. & `BMC` & `VD`	0.462	0.547	0.654
HMM Mix. & `BMC` & `VD`	0.467	0.551	0.658

The calculation of the specificity (or level) of the annotations from a set of genes is straightforward. Given a cluster, we repeat the enrichment test for each term in GO, and retrieve the ones exceeding a given p-value. Then, the length of the path from the root to each enriched GO term is counted and averaged. Since GO is a DAG, one node can be reached by more than one path from the root. Therefore, the average of all possible path lengths is taken. The final score reflects the mean distance of the enriched GO terms to the root of the Gene Ontology. The larger the score value, the higher is the specificity of the functional annotations enriched in the evaluated gene clustering.

4.5.2 Results

Yeast Cell Cycle. As Table 4.1 illustrates, the k-means algorithm obtains a good result with a corrected Rand (`CR`) of 0.43 (specificity of 0.54 and sensitivity of 0.56). Mixture estimation with `BMC` and a posterior Viterbi decomposition obtains the highest values for all indices, with a `CR` of 0.467, specificity of 0.55 and sensitivity of 0.66. The results of CAGED are not included, since it could only find one cluster, which makes the calculation of the indices impossible. Note that the number of clusters in CAGED cannot be controlled by the user. Moreover, model-based clustering with splines, another method taking temporal dependencies into account, has an overall poor result. Furthermore, the mixtures of HMMs (`HMM Mix.`), which perform “soft” cluster assignments during the EM, have a better performance than the use of model-based clustering with HMMs (`HMM Clu.`), which performs hard assignments during the EM method.

Looking at the results in more details, we find that most methods join genes from the cell cycle class“Late G1” with “S” and genes from the cell cycle class “M” with “Early G1”.

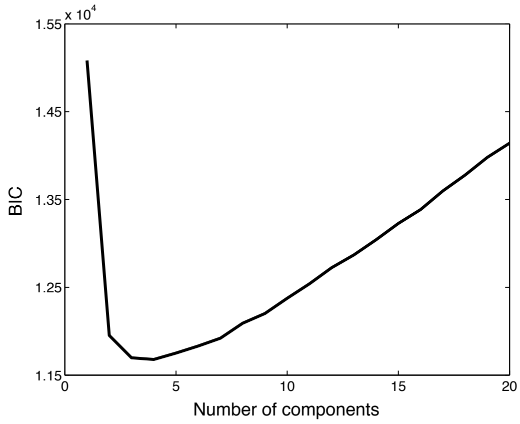


Figure 4.10: *BIC versus number of components for the data set* YCC. *The correct number is five.*

Note that these classes correspond to cell cycle phases with a small phase-shift and their genes have very similar time courses. Interestingly, the BIC index underestimate the number of components by one in YCC (Figs. 4.10), which is a further indication of the difficulty of separating these two classes. The application of the Viterbi decomposition after estimation of mixtures of HMMs improves the results, in particular for these “difficult” classes, which indicates the usefulness of the Viterbi decomposition in refining the clusters.

In relation to the initialization method, BMC obtains higher accuracy than KMI and RMC in most of method combinations. Another important characteristic in favor of BMC is its deterministic nature. As a consequence, there is no need to perform replicates of the experiments and to choose the best replicate, in contrast to use of RMC or KMI.

HeLa Cell Cycle. The initial collection used for mixture estimation for this data set consists of 35 lHMMs with 24-states obtained from RMC. Two groups have periodically expressed time courses. We apply the Viterbi decomposition to these groups. For cluster 1 (Figure 4.11), the first subgroup contains 26 genes known to be in cell cycle phase G2 and one gene to cell cycle phase G2/M, the second subgroup eleven G2 and 19 G2/M, the third subgroup 31 G2/M, two M/G1 and 1 G1/S (see Section 4.1 for description of cell cycle phases). The second group (not shown) contains twelve G1/S and two S-phase genes. Both CDC2 representatives are found in the same subgroup (Figure 4.11, phase 1). Furthermore, cyclin A (Figure 4.11, phase 2) and cyclin B (Figure 4.11, phase 3) are assigned to different subgroups, shifted in phase with respect to the one containing CDC2. Moreover, all time courses that are assigned to the different phases of our G2, G2/M phase cluster are known to be cell cycle regulated in their respective phase [226]. The same holds for the G1/S, S phase subgroups.

In relation to the entropy threshold, we observe an increase in the GO specificity score for lower entropy threshold values, followed by a decrease of specificity for very low threshold values (see Figure 4.12). The mean GO specificity score raises considerably (around 2.0)

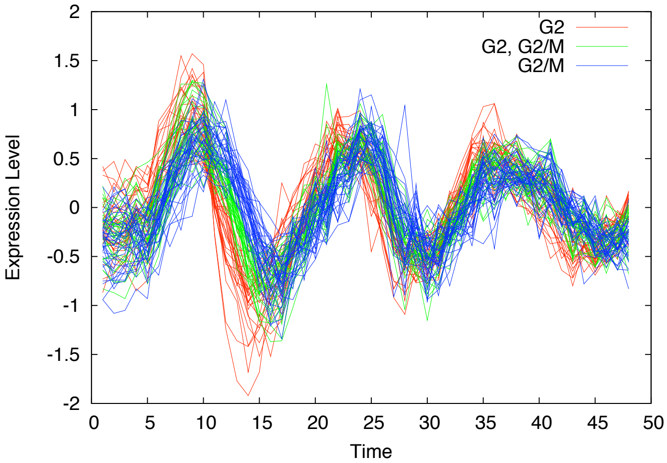


Figure 4.11: *One group has periodically expressed time courses in the* `HeLa` *data set. This group is subsequently decomposed into three subgroups (top to bottom), corresponding to groups of synchronous genes, with the Viterbi decomposition. The first subgroup (red) contains mainly cell cycle phase G2 genes, the second (green) G2 as well as G2/M genes and the third (blue) mostly G2/M genes.*

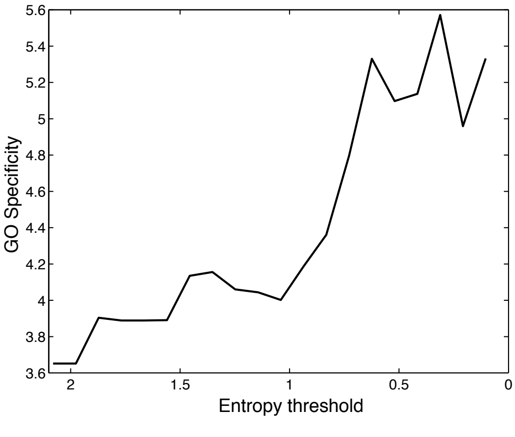


Figure 4.12: *The mean GO specificity level of clusters from* `HeLa` *versus the entropy threshold. The lower the threshold, the less unambiguous are the cluster assignments.*

until the threshold value 0.3. This result indicates that less ambiguous assignments, as derived by the posterior probabilities returned by the mixture model, lead to an enrichment of more specific GO annotations. Therefore, the entropy threshold is a valuable tool for refining the results returned by the mixture of lHMMs and deriving fine grained groups of functionally related genes.

Chapter 5

Analysis of Gene Expression in Lymphoid Development

The study of gene regulatory mechanisms controlling cell proliferation and differentiation is central in developmental biology. In particular, the development of lymphoid cells is well studied, as individual cell populations are easy to obtain and due to clinical interest [140, 177]. In Lymphoid development [25], all starts with the Hematopoietic stem cell (HSC), which differentiates into the Lymphoid progenitor, and later into B-cell, T-cell or Natural Killer cell lineages (see Figure 5.1 for a developmental tree). Recently, several studies have analyzed expression profiling of lymphoid cells in their distinguishable developmental stages [3, 34, 98, 100, 105, 156, 165, 220, 229]. Our main focus is on the analysis of patterns of gene expression in the distinct stages of the developmental tree, the developmental profiles of genes. In particular, we are interested in finding groups of genes displaying a particular pattern of expression, e.g., over-expression in T cells but under-expression in B cells.

As one of the major contribution of this thesis, we propose here a method for analyzing patterns of gene expressions in the course of development. Ideally, such method should exploit inherent dependencies arising from the data, as in methods for analyzing gene expression time-courses (see Chapter 4). We assume that, in development, the sequence of changes from a stem cell to a particular mature cell, as described by a developmental tree, are the most important in modeling gene expression from developmental processes. Motivated by this, we propose dependence trees (`DTree`) to model expression during the course of development [50]. We investigate here two approaches for obtaining the structure of dependence trees. In the first approach, we assume that the structure of the dependence tree is equal to the developmental tree as known by the biologists [50]. In a second approach, we additionally estimate the dependence tree structure from the data [49].

To find groups of co-expressed developmental profiles we use dependence trees in a mixture model [143]. Also, to minimize problems related to over-fitting, we propose Maximum-a-posteriori (MAP) estimates of parameters [80]. By doing so, we obtain a robust and flexible statistical model for clustering genome-wide mRNA expression data sets, which takes the intrinsic dependencies between developmental stages explicitly into account.

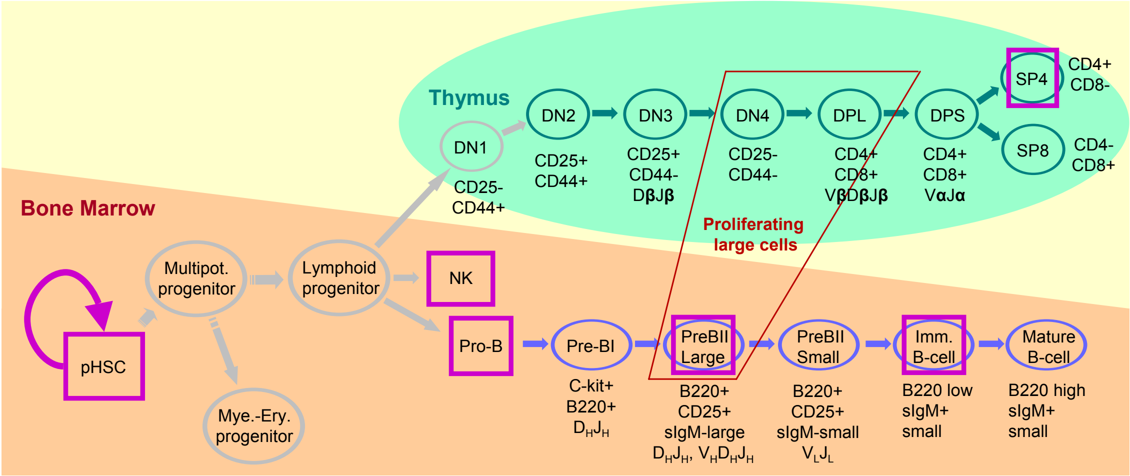


Figure 5.1: *Schematic view of lymphocyte cell development. Developmental stages are depicted as nodes and arrows indicate transition from one stage to another, i.e., specialization. Self-renewing hematopoietic stem cells give rise to T cells in the thymus (green), B cells in the bone marrow (blue) and natural killer cells (NK) via intermediate stages. DN stands for CD4-/CD8- double negative cells, DPL for CD4+/CD8+ double positive large cells, and DPS for CD4+/CD8+ double positive small cells. Cell surface antigens and rearrangement events are partially annotated. Some expression data sets investigated in this Chapter are denoted as follows: green ovals for T Cell and blue ovals for B Cell.*

This chapter is organized as follows. In Section 5.1, we give an overview of related work. Then, we present the dependence tree and the estimation of its parameters in Section 5.2. In Section 5.3, we describe mixtures of dependence trees, and derive the parameters of the MAP estimates (Section 5.3.2). Next, in Section 5.4, we show the results of the analysis of gene expression from lymphoid development. For the mixture of dependence trees with fixed tree structures (Section 5.4.1), we analyze two detailed data sets from B cells [100] and T cells [99]. Furthermore, we explore plausible regulatory roles of microRNAs known to be involved in hematopoiesis. For mixture of dependence trees with estimated structures (Section 5.4.2), we analyze a gene expression compendia with data from hematopoietic stem cells, T cells, B cells and Natural Killer cells. We perform a comparison of several clustering methods on a score based on enrichment analysis of biological pathways. For both methods, results on simulated data show the conditions under which our method has advantages. In Chapter 7, we present final remarks and future work.

5.1 Related Work

Dependence trees were first introduced for discrete variables by Chow and Liu [43], which showed that efficient computation using a maximum weight spanning tree algorithm is possible. They applied the method for pattern recognition of handwritten digits. Mixtures of dependence trees were first proposed in [148]. The authors also proposed extensions to the basic structure estimation algorithm from [43] for sparse data and the use of priors in the tree structure. This also allowed forests (or disconnected trees) to be estimated. It was also shown that the estimated structures of the dependence trees were a good indicator of relevant dependencies between variables. Both studies [43, 148], however, were only concerned with discrete variables, in contrast to our approach, which regards continuous variables.

Another closely related method is the mixture of directed acyclic graphs (DAG) [213]. Indeed, the mixture of DAGs is a more general graphical model than the mixture of dependence trees. The use of DAGs as component models allows to model high order dependencies. However, there is no exact solution for the structure estimation of DAGs. Thus, its estimation is based on heuristics and requires larger computational effort than mixture of dependence trees. Another related research field is the estimation of covariance matrices with zero entries. In [40], an iterative conditional fitting method was applied for computing sparse covariance matrices from arbitrary undirected graphs. While the method obtained better estimates than classical statistical approaches, such as [7], it does not offer a solution for inferring the graph structure. In [182], a similar problem in the context of gene dependence (or association) networks was investigated. The authors applied a shrinkage factor in an efficient way for defining zero entries in the covariance matrix, while keeping it well-conditioned. Both methods have a high computational cost. They are also not able to find association networks, which are specific for particular gene modules, as performed by mixtures of dependence trees.

In the context of mixtures, our method represents an alternative to the parameterization of the covariance matrix of a mixture of multivariate Gaussians (MoG) not previously characterized [37, 79] (see Section 5.2.4 for a discussion). When computing MLE estimates, the dependence tree model essentially imposes zeros in the inverse of the covariance matrix reducing the number of free parameters to $O(L)$. If we considered all the covariances between observations for L developmental stages, it would be straightforward to represent the data distribution by an L-variate Gaussian model with full covariance matrix. However, this parameterization has $O(L^2)$ parameters, which are often unreliable even for small values of L. Moreover, the parameter estimation is prone to over-fit to outliers often found in noisy and scarce data [143]. This was also indicated in our results with simulated data (Section 5.4.1 and Section 5.4.2), where mixtures of Gaussians with full covariance matrix were outperformed by most of the methods. Additionally, in a study in the context of gene expression time courses [232], MoG with full covariance matrix was outperformed by simpler parameterizations of the covariance matrices.

The estimation of the structure of mutagenic trees is a related problem in bioinformatics [62, 63, 223]. In this application, one is interested in inferring the mutation events occurring in cells, such as cancer, which follow a tree-like event structure. For this particular problem, the root is known a priori (a wild type cell without mutations) and only variables with observed mutation events are included as nodes in the tree [63]. The tree structure is estimated with a maximum weight branching algorithm (Edmonds' branching algorithm [46]). Recently, mixture of mutagenic trees, which combined the individual tree estimation from [63] with the EM algorithm, was applied to infer mutation events in HIV strains [23].

The problem approached in this chapter is closely related to the gene expression time-course analysis discussed in Chapter 4. Dependence trees can also be used for analyzing short time courses. We can define the dependence tree structure to be a linear chain connecting consecutive time points. In this scenario, `DTrees` will model only first-order temporal dependencies, but ignore higher order dependencies often present in gene expression time-courses (see Section 4.2.1). On the other hand, models employed in time course analysis [14, 185] cannot be extended to modeling tree like dependency structures arising in developmental processes.

Mixture of dependence trees with estimated structure has some relation to bi-clustering. Bi-clustering methods find not only co-expressed genes but also similarity of expression in the biological conditions. However, bi-clustering methods do not make explicit use of any dependencies (developmental or temporal) in these data sets (see [139] for a survey on bi-clustering algorithms). One of such method, Samba [210], is graph-based and finds strongly connected subgraphs in a bi-partite graph. The bi-partite graph has genes and biological conditions as nodes. The edges between nodes representing genes and biological conditions are weighted proportional to the gene expression value of the given gene in that particular biological condition. Another relevant approach is the use of a non-negative matrix factorization (NMF) [31]. This method decomposes the gene expression matrix in two matrices: one representing the K most significant "meta-conditions" and the other the K most significant "meta-genes". The authors proposed a consensus clustering method for choosing K (or the number of clusters) automatically and minimizing problems related to the random initialization of the method.

Regarding lymphoid development, lymphocyte cell populations can be purified by fluorescence activated cell sorting (FACS) exploiting the large variety of cell surface antigens, which appear in specific order during differentiation as the result of a linear sequence of genomic rearrangements at the T and B cell receptor loci [98, 100]. Based on this, lineage-specific expression and roles of transcription factors have been studied extensively [140, 177, 224]. Recently, a new class of regulatory RNAs, microRNAs, have been identified as being involved in lymphocyte cell development [41, 151, 171].

Several studies [3, 34, 98, 100, 105, 156, 165, 220, 229] have combined FACS mediated cell sorting and mRNA expression profiling to derive a more comprehensive picture of the lymphocytes in distinguishable developmental stages. Nevertheless, prior work on the analysis of gene expression from lymphoid development relies mostly on classical clustering methods, such as self-organizing maps [98, 100], hierarchical clustering [156, 220], k-means [3], principal component analysis (PCA) [229] or on performing tests of differential expression between cell types of interest [165]. One particular interesting study was pro-

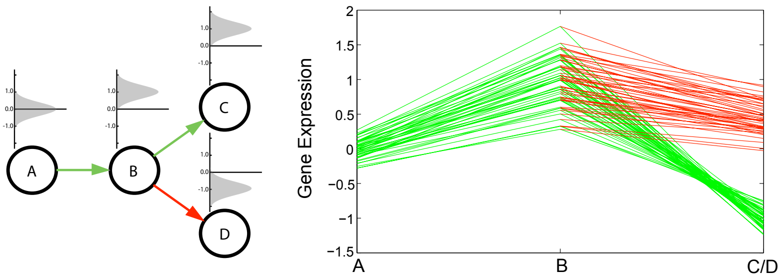


Figure 5.2: *Example of a simple developmental tree and a group of developmental profiles. On the left, we depict a simple developmental tree, where arrows represent dependencies between variables. Above each tree variable, we depict a pdf related to it. On the right, we display the gene expression values (y-axis) in the distinct development stages (x-axis). Each line corresponds to the developmental profile of a given gene of a particular path of the tree on the left, as in a time-course plot. Distinct paths have different colors, according to the tree on the left. In this particular example, we have the path A, B and C in green and B and D in red. By superimposing the lines corresponding to paths B to C and B to D, we can contrast the differences in expression values of genes in these two alternative differentiation lineages.*

posed in [105], where several publicly available data from lymphoid cells were combined and made available for further analyses through an interactive web tool. The authors applied PCA analysis to explore similarities of lymphoid cells based on their gene expression signatures. Furthermore, a simple method based on the correlation measure was used for inferring “networks” of genes. However, that work did not address any developmental aspect of lymphoid cells, as it was restricted to gene expression profiles from lymphoid cells at mature or immature cell stages (later developmental stages). Other studies concentrated on small-scale data, where selected genes are used to infer regulatory networks. One of these studies applied a state-space model to infer networks of T cell activation [173]. Troncale and colleagues adopted Petri Nets to model and infer regulatory networks of early pHSC development [216], while Basso and colleagues proposed a novel algorithm for a similar task [18].

5.2 Dependence Trees

The main assumption underlying dependence trees (`DTree`) is that expression levels of a particular developmental stage depend primarily on expression levels of the immediately preceding stage. For example, given the tree structure depicted in Figure 5.2, we assume the following approximation of the joint probability density function (pdf) of the observation vector from four random variables (x_A, x_B, x_C, x_D)

$$p(x_A, x_B, x_C, x_D) \approx p^T(x_A, x_B, x_C, x_D) = p(x_A)p(x_B|x_A)p(x_C|x_B)p(x_D|x_B). \quad (5.1)$$

In other words, we condition the probability of a given variable on its immediate predecessor in accordance with the tree structure shown in Figure 5.2. In Figure 5.2 right, a group of hypothetical genes with similar developmental profiles is illustrated. The genes display average expression in stage A, up-regulation in stage B, down-regulation in stage C and up-regulation in stage D. Furthermore, the genes have distinct expression intensities, but similar relative expression changes. Genes strongly up-regulated in B are also strongly down-regulated in C and strongly up-regulated in D. These dependencies are reflected in the correlation between these stages. For example, A and B (or B and D) are positively correlated, and stages B and C are negatively correlated. A statistical model for such developmental profiles should include these dependencies between subsequent stages, as it is provided by `DTrees`.

Formally, let $X = (X_1, ..., X_u, ..., X_L)$ be an L-dimensional continuous random vector where the variable X_u denotes the expression values of the developmental stage u and $x = (x_1, ..., x_L)$ denotes an observation of X representing a developmental profile of a gene. Consider a directed graph (V, E), where each vertex in V represents a variable in X, $|V| = L$, and a directed edge $(v, u) \in E$ indicates that variable X_u is dependent on variable X_v. The structure of a `DTree` is represented by a directed tree. A directed tree is a connected directed graph, whose vertices except the root have in-degree equal to 1, and there are no cycles in the graph. For simplicity, we represent the `DTree` structure by the parent map, $pa : \{1, ..., L\} \mapsto \{1, ..., L\}$, where $pa(u) = v$ indicates that $(v, u) \in E$. The root of the `DTree`, which has no incoming edges is represented by $pa(u) = u$. We define the pdf of a `DTree` as

$$p^T(x|\theta) = \prod_{u=1}^{L} p(x_u|x_{pa(u)}, \tau_u). \tag{5.2}$$

We denote the model parameters by $\theta = (pa, \tau_1, ..., \tau_u, ...\tau_L)$. Note, that a `DTree` can be also regarded as an approximation of the joint pdf of a L-dimensional continuous random vector by a product of $L - 1$ second order pdfs [43].

5.2.1 Equivalence of Dependence Trees

We can use the formalism of graphical models and Bayesian networks, which `DTrees` are a particular case, for analyzing characteristics of the model [125]. One interesting aspect is the existence of several `DTrees` with equivalent pdfs. Intuitively, the main information contained in the `DTree` structure are the connected pair of variables, but not the directions of the edges. For example, we can obtain an equivalent `DTree` pdf using an undirected tree representation. Formally, we can apply a graph factorization [125] to the undirected representation of the `DTree` structure, which yields the following pdf [148]

$$p^T(x|\theta) = \frac{\prod_{(u,v)\in E} p(x_u, x_v)}{\prod_{v\in V} p(x_v)^{\deg(v)-1}}, \tag{5.3}$$

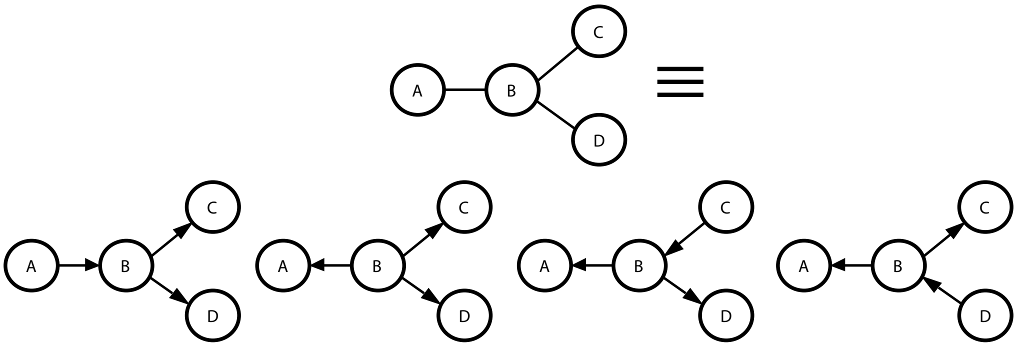


Figure 5.3: *We depict the undirected tree structure of the graph from Figure 5.2 (top), and the four possible directed versions obtained by choosing respectively edges A, B, C and D as a root (bottom).*

where deg(v) is the number of edges of v.

It can be shown with the application of the Bayes rule that the pdfs from Eq. 5.2 and Eq. 5.3 are equivalent,

$$\begin{aligned} p^T(x_A, x_B, x_C, x_D) &= \frac{p(x_A, x_B)p(x_B, x_C)p(x_B, x_D)}{p(x_B)p(x_B)} \\ &= \frac{p(x_A)p(x_B|x_A)p(x_B, x_C)p(x_B, x_D)}{p(x_B)p(x_B)} \\ &= p(x_A)p(x_B|x_A)p(x_C|x_B)p(x_D|x_B). \end{aligned}$$

For any undirected tree structure, we can also obtain a directed tree by choosing a vertex as a root, and directing the edges away from the root. Any arbitrary choice of root will lead to equivalent decompositions of the tree pdfs. For instance, in Figure 5.3 left-middle, we have X_B as a root, which leads to the following pdf

$$p^T(x_A, x_B, x_C, x_D) = p(x_B)p(x_A|x_B)p(x_C|x_B)p(x_D|x_B). \tag{5.4}$$

By Bayes rule we can show that Eq. 5.4 can be easily transformed into Eq. 5.1

$$\begin{aligned} p^T(x_A, x_B, x_C, x_D) &= p(x_B)p(x_A|x_B)p(x_C|x_B)p(x_D|x_B) \\ &= p(x_A, x_B)p(x_C|x_B)p(x_D|x_B) \\ &= p(x_A)p(x_B|x_A)p(x_C|x_B)p(x_D|x_B). \end{aligned}$$

In summary, any directed representation of an underlying undirected tree will lead to equivalent tree pdfs [148]. See [125] for a formal treatment based on the equivalence of pdfs (or distributions) in chain graphs. Given the simplicity and the intuitive representation, this chapter will mostly use directed versions of the tree structures. The choices of the direction of edges are based on the prior knowledge of the data, i.e., the underlying developmental tree.

5.2.2 Parameterization of Dependence Trees

We use conditional Gaussian density functions [126] as conditional densities, denoted by $p(x_u|x_{pa(u)}, \tau_u)$ in Eq. 5.2. Hence, for a given developmental profile x and a non-root developmental stage u with $pa(u) = v$, the pdf takes the following form

$$p(x_u|x_v, \tau_u) = (\sqrt{2\pi}\sigma_{u|v})^{-1} \exp\left(\frac{-(x_u - \mu_u - w_{u|v}(x_v - \mu_v))^2}{2\sigma^2_{u|v}}\right), \qquad (5.5)$$

where $\tau_u = (\mu_u, w_{u|v}, \sigma^2_{u|v})$ are the parameters for one conditional density in the model.

For a given expression data set $\mathbf{X}$ consisting of N gene observations at L developmental stages, let $x_i = (x_{i1}, \ldots, x_{iu}, \ldots, x_{iL})$ be the developmental profile of gene i, and x_{iu} be the expression value of the gene i in development stage u for $1 \leq i \leq N$ and $1 \leq u \leq L$. The maximum likelihood estimates (MLE) for the parameters of the conditional Gaussian are [125],

$$\hat{\mu}_u = \frac{\sum_{i=1}^N x_{iu}}{N}, \qquad (5.6)$$

$$\hat{w}_{u|v} = \frac{\hat{\sigma}_{uv}}{\hat{\sigma}^2_v}, \text{ and} \qquad (5.7)$$

$$\hat{\sigma}^2_{u|v} = \hat{\sigma}^2_u - \hat{w}^2_{u|v}\hat{\sigma}^2_v. \qquad (5.8)$$

These terms can be computed from the sufficient statistics as follows

$$\hat{\sigma}^2_u = \frac{\sum_{i=1}^N (x_{iu} - \hat{\mu}_u)^2}{N}, \text{ and} \hat{\sigma}_{uv} = \frac{\sum_{i=1}^N (x_{iu} - \hat{\mu}_u)(x_{iv} - \hat{\mu}_v)}{N}. \qquad (5.9)$$

The conditional normal pdf can be seen as estimating a linear fit between X_u and X_v, where $w_{u|v} > 0$ indicates a positive linear correlation and $w_{u|v} < 0$ a negative linear correlation between variables; $w_{u|v} = 0$ if the variables are independent. Furthermore, $w_{u|v}$ and $\sigma^2_{u|v}$ are related because the better the linear fit the smaller the variance. For the special case of the root (recall that $pa(u) = u$), $w_{u|u}$ is set to zero, and the conditional density is effectively an univariate normal. The model has $3L - 1$ free parameters. See Section 5.3.2 for the complete derivation of MAP estimates of the conditional Gaussians.

Returning to our example, the model estimates given the developmental tree and expression profiles from Figure 5.2 are the following

$$\begin{aligned}
\tau_A = &\quad (\hat{\mu}_A, \hat{w}_{A|A}, \hat{\sigma}^2_{A|A}) &= (-0.01, 0, 0.02),\\
\tau_B = &\quad (\hat{\mu}_B, \hat{w}_{B|A}, \hat{\sigma}^2_{B|A}) &= (0.97, 2.2, 0.02),\\
\tau_C = &\quad (\hat{\mu}_C, \hat{w}_{C|B}, \hat{\sigma}^2_{C|B}) &= (-0.99, -0.3, 0.01), \text{ and}\\
\tau_D = &\quad (\hat{\mu}_D, \hat{w}_{D|B}, \hat{\sigma}^2_{D|B}) &= (0.45, 0.53, 0.01).
\end{aligned}$$

As expected, $\hat{w}_{B|A}$ and $\hat{w}_{D|B}$ are positive, indicating a linear dependence between these variables. On the other hand, $\hat{w}_{C|B}$ is negative, which indicates a negative correlation between X_B and X_C.

5.2.3 Estimation of the Structure of Dependence Trees

As described in the previous section, in developmental processes the developmental tree structure is already known a priori. Although the developmental tree is an interesting candidate for modeling dependencies, we are also interested in the case of estimating the tree structure from the data. We summarize here our extension to continuous variables of the solution proposed in [43], which considers trees on discrete distributions. The solution is based on finding the `DTree` structure that minimizes the relative entropy between $p(x)$ and the approximation $p^T(x)$

$$p^{T*} = \text{argmin}_{p^T} \mathrm{D}(p||p^T). \tag{5.10}$$

The relative entropy between p and p^T is defined as [54],

$$\mathrm{D}(p||p^T) = \int_X p(x) \log \frac{p(x)}{p^T(x)}.$$

Replacing $p^T(x)$ by Eq. 5.2, we obtain,

$$\begin{aligned}\mathrm{D}(p||p^T) &= \int_X p(x)\log p(x) - \int_X p(x) \sum_{u=1}^{L} \log p(x_u|x_{pa(u)}),\\ &= \mathrm{H}(X) - \int_X p(x) \sum_{u=1}^{L} \log p(x_u) - \int_X p(x) \sum_{u=1}^{L} \log \frac{p(x_u|x_{pa(u)},)}{p(x_u)}\end{aligned}$$

We can simplify the previous equation by applying the Bayes rule and the definition of entropy (H) and mutual information (I) [54],

$$\begin{aligned}\mathrm{D}(p||p^T) &= \mathrm{H}(X) - \sum_{u=1}^{L} \mathrm{H}(X_u) - \int_X \sum_{u=1}^{L} p(x_u, x_{pa(u)}) \log \frac{p(x_u, x_{pa(u)})}{p(x_u)p(x_{pa(u)})},\\ &= \mathrm{H}(X) - \sum_{u=1}^{L} \mathrm{H}(X_u) - \sum_{u=1}^{L} \mathrm{I}(X_u, X_{pa(u)}). \quad (5.11)\end{aligned}$$

Since $\mathrm{H}(X)$ and $\mathrm{H}(X_u)$ are independent of p^T, then Eq. 5.10 can be reduced as follows,

$$pa^* = \text{argmax}_{pa} \sum_{u=1}^{L} \mathrm{I}(X_u, X_{pa(u)}). \tag{5.12}$$

The solution to this problem can be efficiently computed by applying a maximum weight spanning tree algorithm on a fully connected undirected graph, where vertices represent the variables and the weights of edges are equal to the mutual information between variables [43]. The computational complexity of this algorithm is $O(L^2 \log L)$.

Finally, we need to compute $\mathrm{I}(X_u, X_{pa(u)})$ for multivariate Gaussian. Given that $pa(u) = v$, the mutual information is defined as [54]

$$\mathrm{I}(X_u, X_v) = \int_{X_u} \int_{X_v} p(x_u, x_v) \log \frac{p(x_u, x_v)}{p(x_u)p(x_v)} dx_u dx_v. \tag{5.13}$$

Expanding the terms, we obtain

$$\begin{aligned}\mathrm{I}(X_u, X_v) = \; & \textstyle\int_{X_u} \int_{X_v} p(x_u, x_v) \log p(x_u, x_v) dx_u dx_v - \int_{X_u} \int_{X_v} p(x_u, x_v) \log p(x_u) dx_u dx_v \\ & - \textstyle\int_{X_u} \int_{X_v} p(x_u, x_v) \log p(x_v) dx_u dx_v,\end{aligned}$$

and by definition of $\mathrm{H}(X)$, it follows that

$$\mathrm{I}(X_u, X_v) = \mathrm{H}(X_u) + \mathrm{H}(X_v) - \mathrm{H}(X_u, X_v). \tag{5.14}$$

The entropy of an L dimensional multivariate Gaussian pdf is defined as [54],

$$\mathrm{H}(X) = \frac{1}{2} \log(2\pi e)^L |\Sigma_X|, \tag{5.15}$$

where Σ_X is the covariance matrix of X. By substituting Eq.5.15 into Eq.5.14, we obtain

$$\mathrm{I}(X_u, X_v) = \frac{1}{2} \log(2\pi e \sigma^2_{X_u}) + \frac{1}{2} \log(2\pi e \sigma^2_{X_v}) - \frac{1}{2} \log((2\pi e)^2 |\Sigma_{X_u, X_v}|),$$

and, as $|\Sigma_{X_u, X_v}| = \sigma^2_u \sigma^2_v - (\sigma_{u,v})^2$, it follows that

$$\mathrm{I}(X_u, X_v) = \frac{1}{2} \log \left(\frac{(2\pi e)^2}{(2\pi e)^2} \right) - \frac{1}{2} \log \left(\frac{\sigma^2_u \sigma^2_v - \sigma^2_{u,v}}{\sigma^2_u \sigma^2_v} \right),$$

and hence,

$$\mathrm{I}(X_u, X_v) = -\frac{1}{2} \log \left(1 - \frac{\sigma^2_{u,v}}{\sigma^2_u \sigma^2_v} \right). \tag{5.16}$$

Note that the mutual information is proportional to the correlation coefficient $\rho_{u,v} = \frac{\sigma^2_{u,v}}{\sigma^2_u \sigma^2_v}$. That is, it measures the dependence between the two variables; $\mathrm{I}(X_u, X_v) = 0$ if both variables are independent. Moreover, the mutual information is symmetric ($\mathrm{I}(X_u, X_v) = \mathrm{I}(X_v, X_u)$). Therefore, the estimation method does not determine direction of edges. To obtain a directed tree, we select one particular edge as root and direct all edges away from it (as discussed in Section 5.2.1, any direction choice would lead to equivalent `DTree` pdfs).

We propose a "treeness" index for evaluating how well a `DTree` performs in capturing dependence in the data. Intuitively, we measure the proportion of the mutual information represented in the tree edges, in comparison to the total mutual information on all pairs of variables. That is, for a tree structure pa the treeness index is defined as follows

$$T(pa) = \frac{\sum_{u=1}^{L} \mathrm{I}(X_u, X_{pa(u)})}{\sum_{u=1}^{L} \sum_{v=u+1}^{L} \mathrm{I}(X_u, X_v)}. \tag{5.17}$$

A value of zero indicates that no dependence is captured by the `DTree` and 1 indicates that all dependence is captured by the `DTree`.

5.2.4 Dependence Trees and Multivariate Gaussians

There is a close correspondence between the pdfs of multivariate Gaussians and `DTrees`. Given that $pa(u) = v$, a `DTree` pdf is equivalent to a multivariate Gaussian with mean vector $\mu = (\mu_1, ..., \mu_L)$, and entries of the covariance matrix (Σ^T) of the form [179]

$$\sigma_{u,v}^{T} = \sum_{t=pa(v)} w_{v|t} \sigma_{u,t}^{T} + \mathbf{1}(u = v)\sigma_v \tag{5.18}$$

For the example, for the `DTree` shown in Figure 5.2, the corresponding covariance matrix Σ^T is as follows

$$\left\{ \begin{array}{cccc} \sigma_A^2 & w_{B|A} * \sigma_A^2 & w_{C|B} * w_{B|A} * \sigma_A^2 & w_{D|B} * w_{B|A} * \sigma_A^2 \\ w_{B|A} * \sigma_A^2 & \sigma_{B|A}^2 - w_{B|A}^2 \sigma_A^2 & w_{C|B} * (\sigma_{B|A}^2 - w_{B|A}^2 \sigma_A^2) & w_{D|B} * (\sigma_{B|A}^2 - w_{B|A}^2 \sigma_A^2) \\ w_{C|B} * w_{B|A} * \sigma_A^2 & w_{C|B} * (\sigma_{B|A}^2 - w_{B|A}^2 \sigma_A^2) & \sigma_{C|B}^2 - w_{C|B}^2 \sigma_B^2 & w_{C|B} * w_{D|B} * (\sigma_{B|A}^2 - w_{B|A}^2 \sigma_A^2) \\ w_{D|B} * w_{B|A} * \sigma_A^2 & w_{D|B} * (\sigma_{B|A}^2 - w_{B|A}^2 \sigma_A^2) & w_{D|B} * w_{C|B} * (\sigma_{B|A}^2 - w_{B|A}^2 \sigma_A^2) & \sigma_{D|B}^2 - w_{D|B}^2 \sigma_B^2 \end{array} \right\}.$$

This represents a type of covariance matrix parameterization not yet characterized before (see Section 2.3.3 for a discussion and [10, 37] for others covariance matrix parameterizations).

5.3 Mixture of Dependence Trees

In order to find clusters of co-expressed genes, we combine several `DTrees` in a mixture model. Each `DTree` is a representation of a cluster or group of genes with co-expressed developmental profiles, i.e., each `DTree` models distinct patterns of gene expression in the course of development (see Figure 5.4 for an example). Throughout this chapter we refer to the proposed method as `MixDTrees`.

Formally, we combine a set of K `DTrees` in a mixture model

$$p(x|\Theta) = \sum_{k=1}^{K} \alpha_k p_k^T(x|\theta_k), \tag{5.19}$$

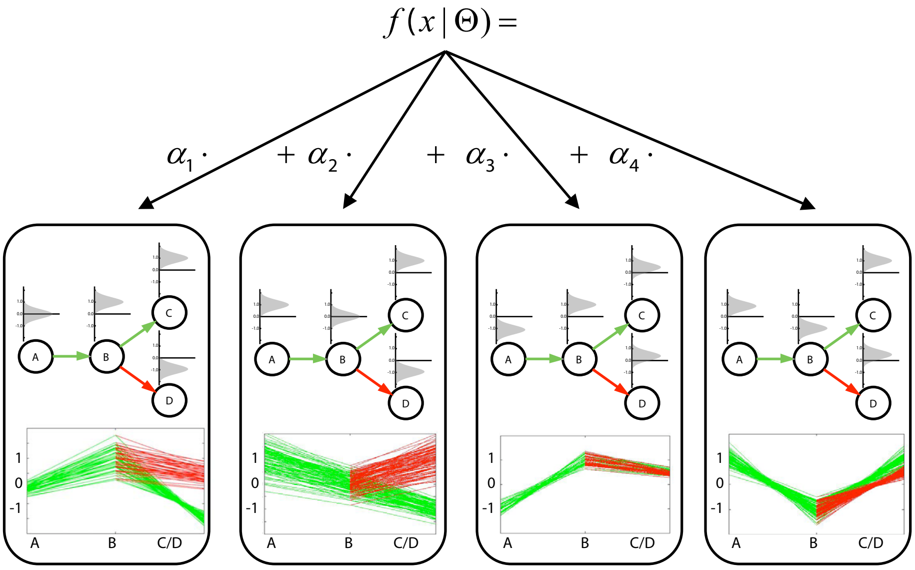


Figure 5.4: *Example of a mixture of four* `DTrees` *with the structure defined in Figure 5.2. Each of these* `DTrees` *models distinct developmental profiles found in the data set employed as example. Furthermore, clusters can have distinct sizes proportional to their α_i's. Note also that it is not necessary that clusters have distinct expression values in branching stages. For example, stages C and D have similar expression values for cluster 3 and 4. This can be interpreted as the genes being equally expressed in the two alternative lineages.*

where α_k is the mixture coefficient (see Section 2.2), $p_k^T(x|\theta_k)$ is the density corresponding to the kth `DTree` as defined in Eq. 5.2, and $\Theta = (\alpha_1, ..., \alpha_K, \theta_1, ..., \theta_K)$.

5.3.1 `MixDTrees` with Developmental Tree as Structure

The differentiation of cells in the course of development is conveniently represented as a developmental tree. The structures of these trees are well-known for most data sets under investigation. Thus, one approach explored in this work is the use of the developmental tree as prior knowledge, that is to define all `DTrees` structures in the mixture to be the same as in the developmental tree. We will call this method `MixDTrees-Dev`. For estimating `MixDTrees-Dev`, we apply the EM algorithm described in Section 2.3.1. In order to do so, we need to define the `DTree` estimates of the M-Step of the EM algorithm. We choose to use maximum-a-posteriori (MAP) estimates, as these minimize problems related to overfitting [80].

5.3.2 Maximum-a-posteriori Estimates

To prevent over-fitting of the `DTree`, we propose the use of a maximum-a-posteriori point estimate (MAP) approach, which regularizes the estimates from Eq. 5.7 and Eq. 5.8. In practice, we define prior distributions for these parameters, penalizing parameters with undesirable values. For example, a low $\sigma^2_{u|v,k}$ is an indication of over-fitting and should be avoided, unless there is enough data (or evidence) for that particular component. Maximum-a-posteriori estimates can be used in the EM algorithm. This can achieved by changing Eq. 2.8 to maximize the expected a posteriori distribution, instead of the complete likelihood function.

More precisely, our aim is to find estimates maximizing the posterior distribution

$$p(\Theta|\mathbf{X},\mathbf{Y}) = \frac{p(\mathbf{X},\mathbf{Y}|\Theta)p(\Theta)}{p(\mathbf{X},\mathbf{Y})} \tag{5.20}$$

where $\mathbf{X}$ is the observed data, $\mathbf{Y}$ indicates which mixture component generated a given observation and Θ are the model parameters. The pdf $p(\mathbf{X},\mathbf{Y}|\Theta)$ is the complete data likelihood (Eq. 2.7), $p(\Theta)$ is the prior distribution on the parameters Θ and $p(\mathbf{X},\mathbf{Y})$ is the prior of the data. We can ignore the last term ($p(\mathbf{X},\mathbf{Y})$) in our problem, as it is independent of Θ, and will be constant for a fixed data set.

Since `MixDTrees` are based on first-order dependencies, it is sufficient to find the parameters in a simple bivariate scenario $(X_u, X_{pa(u)})$, where $pa(u) = v$ and $\mathbf{X}_u$ corresponds to the observed data from variable X_u. This simplifies Eq. 5.20 to

$$p(\Theta|\mathbf{X}_u,\mathbf{X}_v,\mathbf{Y}) \approx p(\mathbf{X}_u,\mathbf{X}_v,\mathbf{Y}|\Theta)p(\Theta). \tag{5.21}$$

where

$$p(\mathbf{X}_u,\mathbf{X}_v,\mathbf{Y}|\Theta) = \prod_{k=1}^{K}\prod_{i=1}^{N}(\alpha_k \cdot p_k^T(\mathbf{X}_u,\mathbf{X}_v|\Theta_k))^{r_{ik}},$$

as shown in Section 2.3.1 and

$$p(\Theta) = \prod_{k=1}^{K} p(\Theta_k) = \prod_{k=1}^{K} p(w_{u|v}|\sigma^2_{u|v,k},\alpha_k)p(\sigma^2_{u|v,k}|\alpha_k)p(\alpha_k),$$

where $\alpha_k = \sum_{i=1}^{N} r_{ik}/N$, and $r_{ik} = p(y_i = k|x_i)$ is the posterior probability (Eq. 2.16) that observation i belongs to `DTree` k.

Priors on Parameters. We use conjugate priors to regularize the parameters $w_{u|v,k}$ and $\sigma^2_{u|v,k}$ and to avoid over-fitting, when there is low evidence for a given component model (or low α_k).

For simplicity of computation, we work with a precision parameter $\lambda_{u|v,k} = (\sigma^2_{u|v,k})^{-1}$. We define the prior of $\lambda_{u|v,k}$ to be proportional to

$$p(\lambda_{u|v,k}|\nu_{u|v,k},\alpha_k) \sim \text{Exponential}\left(\frac{\lambda_{u|v,k}}{\alpha_k}\right) = \frac{\sum_{i=1}^N r_{ik}}{\lambda_{u|v,k}} \exp\left(-\frac{\sum_{i=1}^N r_{ik}}{\lambda_{u|v,k}}\right) \tag{5.22}$$

where $\nu_{u|v,k}$ is a hyper-parameter. Intuitively, this prior penalizes variables with low variances and low evidence, enforcing higher $\sigma^2_{u|v,k}$.

The prior of $w_{u|v,k}$ is defined as follows

$$p(w_{u|v,k}|\lambda_{u|v,k},\sigma^2_{v|k},\alpha_k,\beta_{u|v,k}) = N(0,\beta_{u|v,k}(\lambda_{u|v,k}\alpha_k\sigma^2_{v|k})^{-1}), \tag{5.23}$$

which is invariant to the scale of the variables X_u and X_v, and has $\beta_{u|v,k}$ as a hyper-parameter. Intuitively, this prior penalizes variables with high covariance and low evidence, enforcing smaller $w_{u|v,k}$ values.

Derivation of MAP Estimates. By replacing Eq. 5.5, 5.23 and 5.22 into Eq. 5.21 and taking the logarithm, we obtain

$$\begin{aligned}
\log p(\Theta|\mathbf{X}_u,\mathbf{X}_v,Y) \;=\;& -\frac{1}{2}\sum_{k=1}^K\sum_{i=1}^N r_{ik}\log(\lambda_{u|v,k}) \\
& -\sum_{k=1}^K\sum_{i=1}^N r_{ik}\left((x_{iu}-\mu_{u|k}-w_{u|v,k}(x_{iv}-\mu_{v|k}))^2\lambda_{u|v,k}/2\right) \\
& -\frac{1}{2}\sum_{k=1}^K \log(\frac{\beta_{u|v,k}}{\lambda_{u|v,k}\sigma^2_{v|k}\sum_{i=1}^N r_{ik}}) - \sum_{k=1}^K \frac{w^2_{u|v,k}\sigma^2_{v|k}\sum_{i=1}^N r_{ik}\lambda_{u|v,k}}{\beta_{u|v,k}} \\
& -\frac{1}{2}\sum_{k=1}^K \log(\frac{\nu_{u|v,k}}{\sum_{i=1}^N r_{ik}}) - \sum_{k=1}^K \frac{\lambda_{u|v,k}\sum_{i=1}^N r_{ik}}{\nu_{u|v,k}}.
\end{aligned}$$

We can take the derivate of the MAP with respect to $w_{u|v,k}$ as follows

$$\begin{aligned}
\frac{\partial \log p(\Theta|\mathbf{X}_u,\mathbf{X}_v,Y)}{\partial w_{u|v,k}} \;=\;& \sum_{i=1}^N r_{ik}\left((x_{iu}-\mu_{u|k}-w_{u|v,k}(x_{iv}-\mu_{v|k}))x_{iv}\lambda_{u|v,k}\right) \\
& -\frac{w_{u|v,k}\sum_{i=1}^N r_{ik}\sigma^2_{v|k}\lambda_{u|v,k}}{\beta_{u|v,k}},
\end{aligned}$$

and setting this equation to zero

$$0 = \sigma_{u,v|k} - w_{u|v,k}\sigma^2_{v|k} - \frac{w_{u|v,k}\hat{\sigma}^2_{v|k}}{\beta_{u|v,k}},$$

yields the MAP estimate,

$$\hat{w}_{u|v,k} = \frac{\hat{\sigma}_{u,v|k}}{\hat{\sigma}^2_{v|k}(1+\beta^{-1}_{u|v,k})}. \tag{5.24}$$

The MAP estimate of $\lambda_{u|v,k}$ can be derived in the following way,

$$\begin{aligned}\frac{\partial \log p(\Theta|\mathbf{X}_u, \mathbf{X}_v, Y)}{\partial \lambda_{u|v,k}} &= -\frac{1}{2}\sum_{i=1}^{N} r_{ik}(\lambda_{u|v,k})^{-1} \\ &\quad -\frac{1}{2}\sum_{i=1}^{N} r_{ik}(x_{iu} - \mu_{u|k} - w_{u|v,k}(x_{iv} - \mu_{v|k}))^2 \\ &\quad -\frac{w^2_{u|v,k}\sum_{i=1}^{N} r_{ik}\sigma^2_{v|k}}{\beta_{u|v,k}} + \frac{\sum_{i=1}^{N} r_{ik}}{\nu_{u|v,k}}.\end{aligned}$$

Setting it to zero yields

$$0 = -(\lambda_{u|v,k})^{-1} + \hat{\sigma}^2_{u|k} - w^2_{u|v,k}\hat{\sigma}^2_{v|k} - \frac{w^2_{u|v,k}\hat{\sigma}^2_{v|k}}{\beta_{u|v,k}} - \frac{1}{\nu_{u|v,k}},$$

$$(\lambda_{u|v,k})^{-1} = \hat{\sigma}^2_{u|v,k} = \hat{\sigma}^2_{u|k} - w^2_{u|v,k}\hat{\sigma}^2_{v|k}(1+\beta^{-1}_{u|v,k}) - \nu^{-1}_{u|v,k}. \tag{5.25}$$

When $\beta_{u|v,k} \to \infty$ and $\nu_{u|v,k} \to \infty$, the prior becomes non-informative, and MAP and ML estimates are equal. All the estimates make use of the following sufficient statistics

$$\hat{\mu}_{u|k} = \frac{\sum_{i=1}^{N} r_{ik}x_{iu}}{\sum_{i=1}^{N} r_{ik}}, \tag{5.26}$$

$$\hat{\sigma}^2_{u|k} = \frac{\sum_{i=1}^{N} r_{ik}(x_{iu} - \hat{\mu}_{u|k})^2}{\sum_{i=1}^{N} r_{ik}}, \tag{5.27}$$

$$\hat{\sigma}_{u,v|k} = \frac{\sum_{i=1}^{N} r_{ik}(x_{iv} - \hat{\mu}_{v|k})(x_{iu} - \hat{\mu}_{u|k})^2}{\sum_{i=1}^{N} r_{ik}}. \tag{5.28}$$

Hyper-parameters Estimates via Empirical Bayes. In an empirical Bayes approach [36], by derivating Eq. 5.21 in relation to the hyper-parameters, we can estimate the maximum a posteriori values of $\beta_{u|v,k}$ and $\nu_{u|v,k}$ from the data as follows

$$\frac{\partial \log p(\Theta|\mathbf{X}_u, \mathbf{X}_v, Y)}{\partial \beta_{u|v,k}} = \frac{1}{2\beta_{u|v,k}} - \frac{w_{u|v,k}^2 \sum_{i=1}^N r_{ik}\sigma_{v|k}^2}{2\beta_k^2 \sigma_{u|v,k}^2},$$

setting it to zero

$$0 = -\beta_{u|v,k} - \frac{\sigma_{v|k}^2 \sum_{i=1}^N r_{ik} w_{u|v,k}^2}{2\sigma_{u|v,k}^2}$$

and by definition of $\sigma_{u|v,k}^2$ and $w_{u|v,k}$, this yields

$$\hat{\beta}_{u|v,k} = \frac{\sum_{i=1}^N r_{ik}}{\frac{2\sigma_{u|k}^2 \sigma_{v|k}^2}{\sigma_{u,v|k}^2} - 2}. \tag{5.29}$$

For $\nu_{u|v,k}$, we have

$$\frac{\partial \log p(\Theta|\mathbf{X}_u, \mathbf{X}_v, Y)}{\partial \nu_{u|v,k}} = -\frac{1}{2\nu_{u|v,k}} - \frac{\sum_{i=1}^N r_{ik}\lambda_{u|v,k}}{2\nu_{u|v,k}^2},$$

setting this equation to zero, we obtain

$$\hat{\nu}_{u|v,k} = -\frac{\sum_{i=1}^N r_{ik}}{2\sigma_{u|v,k}^2} \tag{5.30}$$

Both empirical priors penalize variables with large variances or with low evidence enforcing respectively lower $w_{u|v,k}$ and higher $\sigma_{u|v,k}^2$.

5.3.3 `MixDTrees` with Estimated Structure

We do not expect that all genes in a particular developmental process will share the same dependence structure, nor that the most likely `DTree` will exactly match the developmental tree per se. Indeed, we expect that some genes will be particularly correlated in particular developmental lineages, but not in others. For example, group 1 from Figure 5.5 has genes tightly over-expressed in the blue lineage ($\{X_D, X_E, X_F\}$), as does group 2 in the orange lineage ($\{X_B, X_C\}$). We also expect that some genes, which are important for earlier developmental stages, to have similar expression profiles in stages near the root, but not in mature cell types (leaf vertices of a developmental tree). See for example group 3 in Figure 5.5, which exhibits over-expression in all earlier stages ($\{X_A, X_B, X_D\}$).

To infer these group-specific dependencies, we estimate a mixture of K `DTrees`, where each component have its tree structure estimated from the data. We will call this approach mixture of dependence trees with estimated structure (`MixDTrees-Str`). Note that the mixture of dependence trees with estimated structure corresponds to a relaxation of `MixDTrees-Dev` (Section 5.3.1), when a single dependence tree structure is assumed.

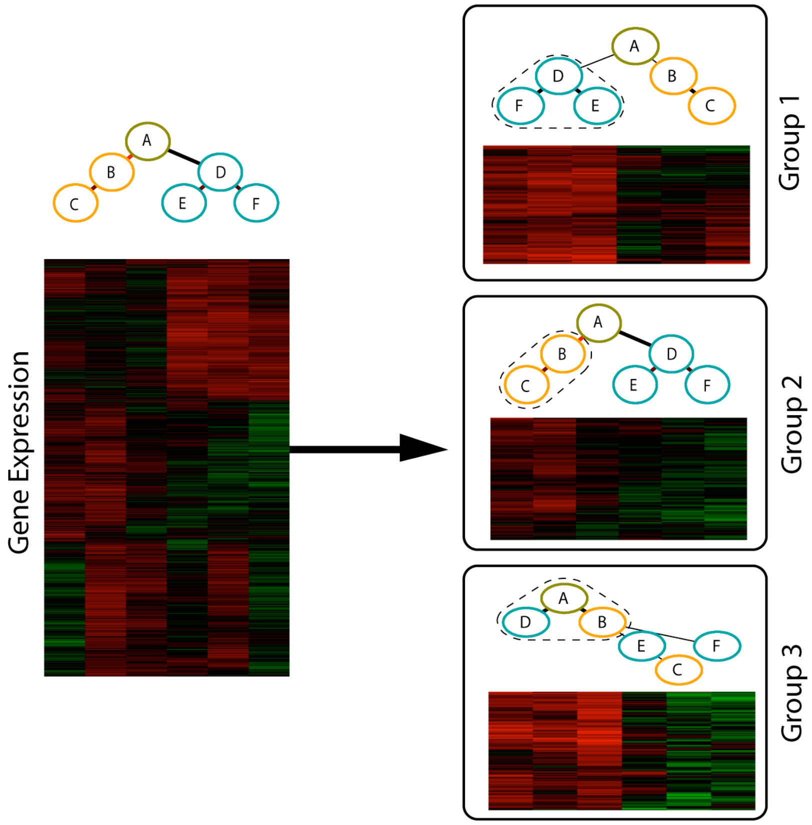


Figure 5.5: *Illustrative example of a developmental tree and its gene expression data (left). The developmental tree is constituted of a stem cell (stage A), an "orange" lineage (stages B and C) and a "blue" lineage (stages D, E and F). The red-green plot depicts the relative expression, where lines corresponds to gene profiles and columns to developmental stages ordered as in the above tree. In the right, we depict three groups of genes and their corresponding estimated tree structure as found by* `MixDTrees` *in the gene expression data in the left (see Section 5.3.3 for complete plot description).*

For estimation of `MixDTrees-Str`, we need to perform the method described in Section 5.2.3 for each `DTree` prior to the M-Step [148]. Once the structure is chosen, `DTree` parameters are set with the MAP estimates (see Section 5.3.2).

Visualization of `DTree` with Estimated Structure. The branches in the estimated tree structure reflect similarity in expression of developmental stages (stages in a same branch will share a similar expression profiles). To highlight these similarities, we propose the following plots. Gene clusters are depicted as a heat-map with red values indicating over-expression and green values indicating under-expression [71]. In this plot, the lines (gene profiles) are ordered as proposed in [16]. For the columns (developmental stage profiles), we compute all possible columns orderings and select the one that has a minimal difference in the mutual information of adjacent columns. To further help the interpretation of individual clusters, we compute strongly connected components [46] (SCC) in the graph returned after thresholding the mutual information matrix. An optimal threshold parameter is obtained by evaluating the resulting SCC with the silhouette index [115]. SCC indicate within a DTree, which developmental stages in a particular branch have similar expression profiles.

5.4 Experiments

We describe in the Section 5.4.1 our analyses performed in [50], where MixDTrees with the developmental tree as structure(MixDTrees-Dev) is evaluated with two detailed studies covering several stages of the B and T cell development [99, 100]. Also, putative roles of microRNAs related to lymphoid development are investigated. In Section 5.4.2, we evaluate the use of MixDTrees with estimated structure (MixDTrees-Str) in a gene expression compendia containing early hematopoietic development cells and three lineages of lymphoid cells: B-cells, T-cells and Natural killer cells [3, 156, 165, 220, 229]. The method performance is compared with other methods via a score based on the enrichment of biological pathways. In both cases, in order to evaluate general characteristics of the methods, we use also simulated data sets.

5.4.1 MixDTrees with Developmental Tree Structure

Data

T Cell Development (TCell). This data set contains measurements of gene expression during the development of T cells in mouse [98]. Based on cell surface markers seven stages have been distinguished: CD4 and CD8 double negatives (DN2, DN3, DN4), large double positives (DPL), small double positives (DPS), single positive CD4 (SP4) and single positive CD8 (SP8) (see Figure 5.1 for the corresponding tree, and the original publication for details [98]). Affymetrix MU11k chips with four or five replicates are used to measure the expression levels of 13,104 mouse genes. We perform variance stabilization [104] on all chips, and compute the median values of replicates. To facilitate comparisons, we use the same list of 1,318 differentially expressed genes that was used by Hoffmann and colleagues [98]. Furthermore, we normalize the expression levels separately for each gene to mean zero and standard deviation one, as is routine in gene expression analysis. Finally, we map each probe set to a gene symbol if it exists in the respective chip platform annotation provided by the GEO database [69].

B Cell Development (BCell). This data set contains expression levels of five consecutive stages of the B cell lineage: Pre-BI, large Pre-BII, small Pre-BII, immature, and mature B cells [100]. This study was also conducted on Affymetrix MU11k chips. We pre-process the data exactly as it is described for TCell.

Lymphoid Development Related microRNAs (LympMIR). We collect 17 microRNAs that have been found to be involved in Lymphoid development or, at least, differentially expressed between distinguishable lymphocyte cell types [41, 44, 76, 151, 171]: mmu-miR-24, mmu-miR-26a, mmu-miR-142-3p, mmu-miR-146, mmu-miR-150, mmu-miR-155, mmu-miR-181a, mmu-miR-181b, mmu-miR-181c, mmu-miR-191, mmu-miR-221, mmu-miR-222, mmu-miR-223 and mmu-miR-342. Additionally, we include mmu-miR-15a, mmu-miR-15b, and mmu-miR-16, as they participate in the regulation of cell proliferation and apoptosis [35, 55]. Since in this work we refer exclusively to microRNAs of the mouse, the species prefix mmu is omitted throughout the text. The lists of candidate targets of these microRNAs are obtained in the miRBase Targets database [91] (Version 2.0), which uses the Miranda algorithm [72] to search for possible microRNA binding sites in the gene sequences.

Simulated Data (SIM). To generate this data set, we use MixDTrees-Dev with random parameterizations. All DTrees have theirs structure fixed to the tree represented in Figure 5.2. Then, we randomly chose $\mu_{u|v,k}$ from the range $[-1.5, 1.5]$ and $\sigma^2_{u|v,k}$ from $[0, 1]$. We create five experimental settings to inspect the performance of the method in the presence of distinct levels of dependence. For these five settings, we sample $w_{u|v,k}$ from $[-\epsilon, \epsilon]$ (independent data), $[-0.5, 0.5]$, $[-1, 1]$, $[-1.0, -0.5] \cup [0.5, 1]$ and $[-1, -1+\epsilon] \cup [1-\epsilon, 1]$ (tree dependent data), respectively, where $\epsilon = 0.001$. We set K to five and mixture coefficients α equal to $(0.1, 0.15, 0.2, 0.2, 0.35)$. For each experimental setting, we generate ten such mixtures, and sample 500 development profiles from each.

Results

We apply MixDTrees-Dev to two biological data sets: TCell and BCell. We compare our results with the ones obtained in [98, 100], which use Self-organizing maps (SOM) [120] as clustering method. For estimating MixDTrees-Dev, we perform the following. The mixture estimation method is initialized with K random DTrees (see Section 2.3.2). We, then, estimate then the mixture model using the EM-algorithm with MAP estimates. For both TCell and BCell, we use the same number of clusters (20) as [98, 100]. For evaluating the results, our analysis is complemented with information from OMIM [158], the Gene Ontology database [9] and from the literature. Furthermore, we perform a microRNA enrichment analysis in the clusters founds in both data sets to investigate putative roles of microRNAs related to lymphoid development.

We resort to simulated data to compare our method with established clustering methods, such as SOM, k-means and mixture of Gaussians, when inferring tree components in complex mixtures for varying levels of dependence between the individual variates. As we have class labels in the simulated data, we can evaluate the clusters with the use of external indices.

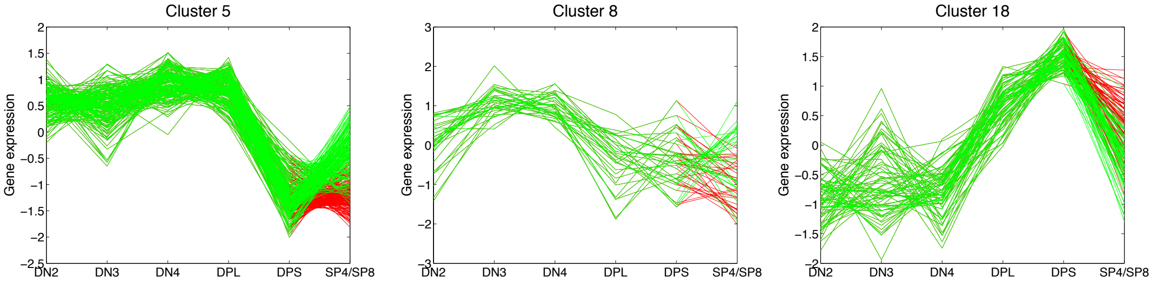


Figure 5.6: *Selected clusters from* `MixDTrees-Dev` *for* `TCell`*. We depict the clusters 5, 8 and 18 found in* `TCell`*, expression values on the y-axis, and cell types on the x-axis. Lines corresponding to developmental profile values between stages DN2, DN3, DN4, DPL, DPS and SP4 are in green and between DPS and SP8 in red.*

T Cell Development (`TCell`). `TCell` is a gene expression data set from seven differentiation stages of the T cell development (see Figure 5.1 for the developmental tree). The only branch in this tree is the final differentiation of DPS precursors into CD4 single positive SP4 cells and CD8 single positive SP8 cells. Most clusters found by `MixDTrees-Dev` from `TCell` show a distinctive pattern of differential expression along the developmental path, but they do not differ between SP4 and SP8 cells (clusters 4, 7, 11, 13, 14, 15, 16, 19, and 20). The most noticeable changes occur at the DPL stage in which the cells are proliferating and, subsequently, start to rearrange the TCRα-locus. This is also reflected in the overall correlation matrix[1]. Although the expression values of all neighboring stages are positively correlated, the correlation between the DPL stage and the DPS stage is much smaller in comparison to the double negative stages, all of which show high correlation. The correlation matrix suggests that SP4 and SP8 cells are more similar to each other than to their precursor DPS cells, which is expected since the two types of mature T cells share many cellular functions [98]. The largest differences with respect to SP4 and SP8 are found in clusters 5 and 18 (Figure 5.6). GO enrichment analysis shows that cell-cycle genes are clearly enriched in cluster 5. In contrast, cluster 18 mainly contains regulatory proteins involved in transcription and signaling (see Figure 5.6).

In order to demonstrate that our method is able to extract additional biological information, we concentrate our discussion on clusters showing distinct developmental profiles that could not be detected by the use of SOM [98]. For such a cluster, we assign functions to genes using the GO term annotation and complementary literature. In our analysis, we find that genes of cluster 8 are over-expressed in DN3 and DN4 cells 5.6), a developmental profile that had not been identified by SOM. With SOM, the genes of this cluster are dispersed over the two clusters (see Table B.1). Out of the 30 genes of cluster 8, seven are related to vesicle transport or to the Golgi/ER system. Additionally, we find five cell-cycle related genes, three involved in mitochondrial function, and seven genes of other functions,

[1] A simple way to check for similarities in the expression between developmental stages is to compute the correlation matrix of the data set at hand. As discussed in Section 5.2.3, the correlation matrix is proportional to the mutual information matrix.

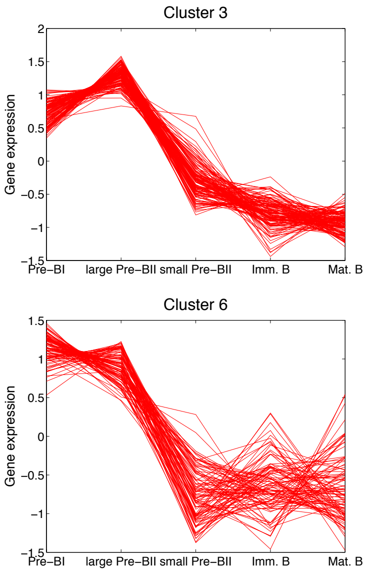


Figure 5.7: *Selected clusters from* `MixDTrees-Dev` *for* `BCell`*. We depict clusters 3, 5, 6 and 20 found in* `BCell`*, expression values on the y-axis, and cell types on the x-axis. Lines corresponding to developmental profile values between all stages are in red.*

which are mainly involved in signaling. These findings agree with the functions of DN3 and DN4 cells, which is the transport of precursor receptor molecules to the cell surface membrane and the initiation of proliferation. All these facts supports our claims that our method is able to identify functionally relevant groups of genes.

B Cell Development (`BCell`). Like in the `TCell` study, we investigated gene expression for five consecutive stages during B cell development (Figure 5.1). The correlation matrix of `BCell` suggests dependencies between gene expression values of successive stages, with the largest correlation between pre-BI and large pre-BII cells and between immature and mature B cells. When we compare, our clustering results to those obtained by SOM [98], we observe similar average developmental profiles, although the contingency table indicates differences in the cluster compositions (Table B.2). Clusters 3, 5 and 6, for example, contain genes that are up-regulated in pre-BI and large pre-BII cells and down-regulated in later developmental stages (Figure 5.7). Consistent with the phenotype of these cells, the function assigned to the genes of this cluster are mainly related to proliferation. GO categories that are associated with mitosis, cell-cycle and chromatin remodeling are clearly over-represented in these clusters.

Cluster 20 shows an average developmental profile that was not detected with SOM [98, 100]. The genes of this cluster are down-regulated in pre-BI cells, in which the first rearrangement of the D^H and J^H segments on the H chain loci has taken place, and up-regulated in all the following developmental stages (Figure 5.7). With SOM [98], these 23 genes are found distributed over the four clusters 11, 13, 14 and 17 (Table B.2). The most

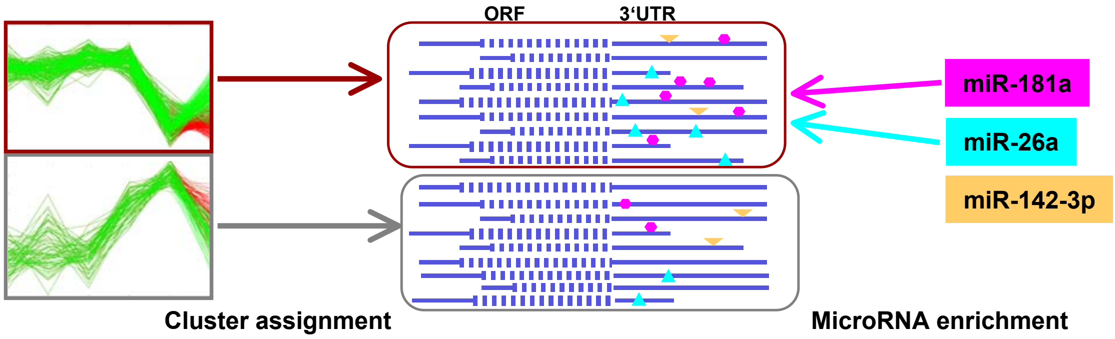


Figure 5.8: *Strategy to identify microRNAs and their target genes over-represented in clusters of co-expressed genes (indicated left) as part of a post-transcriptional regulatory mechanism. In the middle mRNAs clustered according to our mixture results are depicted and potential microRNA binding sites in their 3'UTRs are illustrated.*

plausible common function of some genes from cluster 20 is the regulation of survival and apoptosis during B cell development. The gene products *Nfkbia*, *Traf5* and the Src-family protein tyrosine kinases *Lyn* and *Syk* are known regulators of NF-kappa B activity, which in turn has been found to be involved in B cell fate decision and survival [2, 97, 152]. Similarly, Krupel-like factor 2 (*Klf2*) protects cells against TNF-alpha induced apoptosis [86]. Furthermore, *Icam-2* and *Rhoh*, whose encoding genes are two other members of cluster 20, regulate the adhesiveness of primary B cells depending on their activation state and protect them from apoptosis [158, 164].

MicroRNA Target Discovery. LympMIR contains a set of 17 microRNAs that are potentially involved in lymphocyte cell development. It has been proposed that microRNAs bind target mRNAs specifically via base pairing. This, subsequently, leads to interference of the translational machinery or mRNA degradation, and thus can control whole groups of genes simultaneously [17]. Recent microarray studies have demonstrated that the microRNA expression negatively correlates with mRNA target expression in a tissue specific manner [129, 133, 200].

Having identified clusters of co-expressed genes with `MixDTrees-Dev` for the B cell and T cell data sets, we ask whether a certain microRNA could be a potential regulator of one of these clusters (see Figure 5.8). For this task, we first obtain lists of potential target genes for each microRNA from the miRBase Targets database [91], which contains predictions made by sequence based methods. Given our clustering results, we use an enrichment analysis to obtain a list of microRNAs, whose potential targets are over-represented in a cluster. This is an approach similar to finding Gene Ontology terms over-represented in a cluster of genes, as described in Appendix A. A lower p-value indicates a high count of microRNA targets in a particular cluster, i.e., higher “microRNA enrichment”. By choosing a p-value cut-off, we can construct a list of enriched microRNAs for each cluster as well as a list of target genes related to the enriched microRNAs.

Table 5.1: *List of* `LympMIR` *enriched in the clusters from* `MixDTrees-Dev` *for data sets* `TCell` *and* `BCell`. *We display the cluster and data set id, the list of microRNA and list of target genes, with* p*-values* < 0.05 *and at least four target genes per cluster. Genes involved in cell proliferation or DNA repair are depicted in bold. The indices indicate to which microRNA a gene is related to, when there is more than one enriched microRNA in a cluster.*

Cluster ID	MicroRNA	Target Genes
`TCell` 3	miR-222	*Elovl6, Nme1, Rcn1, Rps3*
`TCell` 5	miR-15a^{1},miR-181a^{2},miR-221^{3}, miR-24^{4},miR-26a^{5}	*2410015N17Rik*^{4}, *Alad*^{1,4}, *Atpif1*^{1,5}, ***Aurkb***^{2}, ***Cdc25a***^{1}, ***Chek1***^{1} ***Cks1b***^{2,4}, ***Cks2***^{5}, *Eed*^{2}, ***H2afx***^{4}, *Kpnb1*^{3}, ***Mcm5***^{3}, *Nasp*^{3,5}, *Pex7*^{2}, *Psmd12*^{2}, *Ranbp5*^{2}, *Rars*^{1}, ***Tk1***^{3}, *Trip13*^{1}, *Uchl5*^{5}
`TCell` 10	miR-142-3p^{6},miR-150^{7}	*Gfi1*^{6}, *Marcks*^{6}, *Msh6*^{6}, *Pp11r*^{7}, *Psmc1*^{6,7}
`TCell` 11	miR-146^{8},miR-16^{9},miR-181b^{10}	*Atp1b3*^{10}, *Ipo4*^{9}, *Klhdc2*^{10}, *Mrpl30*^{8}, *Orc5l*^{8}, *Tuba4*^{9}
`BCell` 3	miR-181b^{1},miR-181c^{2},miR-26a^{3}	*Atpif1*^{3}, ***Aurkb***^{1,2}, *Cbx1*^{3}, ***Cdc45l***^{2}, ***Cks1b***^{1,2}, ***Cks2***^{3}, *Cox5a*^{3}, *Hmgb2*^{1,2}, *Melk*^{1,2}, ***Ttk***^{1,2}, *Uchl5*^{3}
`BCell` 5	miR-15a^{4},miR-15b^{5},miR-221^{6}, miR-223^{7}	***Cdca4***^{4,5}, ***Chek1***^{4,5}, ***Mcm4***^{7}, *Nasp*^{6}, *Nfyb*^{6}, ***Smc4l1***^{7} *Tuba2*^{4,5,7}
`BCell` 6	miR-155^{8},miR-191^{9}	***Ctps***^{9}, *Ddx1*^{8}, *Hint1*^{9}, ***Mcm2***^{8}, *Phf17*^{8}, *Prdx4*^{9}, *SNrpd1*^{9}
`BCell` 19	miR-142-3p^{14},miR-342^{15}	*2410002F23Rik*^{14}, *H2-Eb1*^{14}, *Ltb*^{15}, *Tap2*^{14,15}

For TCell, our target prediction scheme identified, in four out of the 20 clusters, significant enrichment for eleven out of the 17 initial microRNAs (Table 5.1). In these four clusters, we detect 35 candidate target genes in total, which is a considerable reduction of the set of 229 targets that had been previously predicted by sequence based methods alone [91]. For BCell these numbers are respectively, eleven out of the 17 microRNAs, four out of the 20 clusters, and 29 out of the 273 predicted targets (Table 5.1). In particular, we find the five microRNA families miR-15, miR-181, miR-221, miR-26, and miR-142-3p to be enriched in both TCell and BCell. See Table B.3 and Table B.4 for p-values of microRNA enrichment of all data sets.

As mentioned earlier, the BCell clusters 3, 5, and 6 show a similar expression profile. We find that cluster 5 from TCell overlaps substantially with clusters 3 and 5 from BCell (Table 5.1). In TCell cluster 5, we find miR-15a, miR-181a, miR-26a, miR-24, and miR-221 as potential regulators and 20 potential target genes, seven of which are also present among the 18 BCell candidate genes of clusters 3 and 5. The developmental profiles of the clusters of both lineages show similar phenotypical features, namely up-regulation in the proliferating large cell populations (DN4, DPL, large pre-BII) and from then on strict down-regulation. In TCell cluster 5 there are eight genes and in the BCell clusters 3 and 5 there are nine target genes that are known to be involved in DNA metabolism, cell-cycle and mitosis (Table 5.1). This suggests a regulatory role for the identified microRNAs in reducing the transcript levels of genes that are important for cell proliferation. This is supported by the fact that a similar role for microRNA was found in Drosophila germline stem cells [94].

At the individual gene level, we identify some candidate microRNA targets for further detailed analysis: the three known genes (*H2-Eb1*, *Ltb*, *Tap2*) of BCell cluster 19 are all involved in the antigen presentation by MHC class II molecules [158, 166]. In the context of the cell cycle, *Chek1* (clusters TCell 5 and BCell 5) and *Cdc25a* (cluster TCell 5) are important for the transition between G1/S and G2/M phases [32]. Furthermore, both genes are candidate targets of the same microRNA, miR-15a, which is related to apoptosis in chronic lymphoid leukemia cells [44]. Another interesting gene codes for the nuclear factor Y (*Nfyb*; cluster BCell 5), which regulates *Hoxb4* [85], *Cdc34* [170] and the major histocompatibility complex in mice [237]. These are all important genes for lymphoid development. The mRNA of the growth factor independence-1 transcription factor (*Gfi1*; cluster TCell 10) is a potential target of miR-142-3p. *Gfi1* has as function the restriction of cell proliferation and maintenance of the functional integrity of lymphocyte cells [116]. Moreover, *Gfi1* is implicated in the transition from CD4/CD8 double negative to double positive T cells [188].

Simulated Data (SIM). We used MixDTrees-Dev with MAP and MLE estimates, mixture of Gaussians (MoG), k-means and SOM to compute clusters. We can compare to the classes used in data generation with cluster results to compute specificity (Eq. 3.14) and sensitivity (Eq. 3.13) of the clustering solutions. To compare the significance of differ-

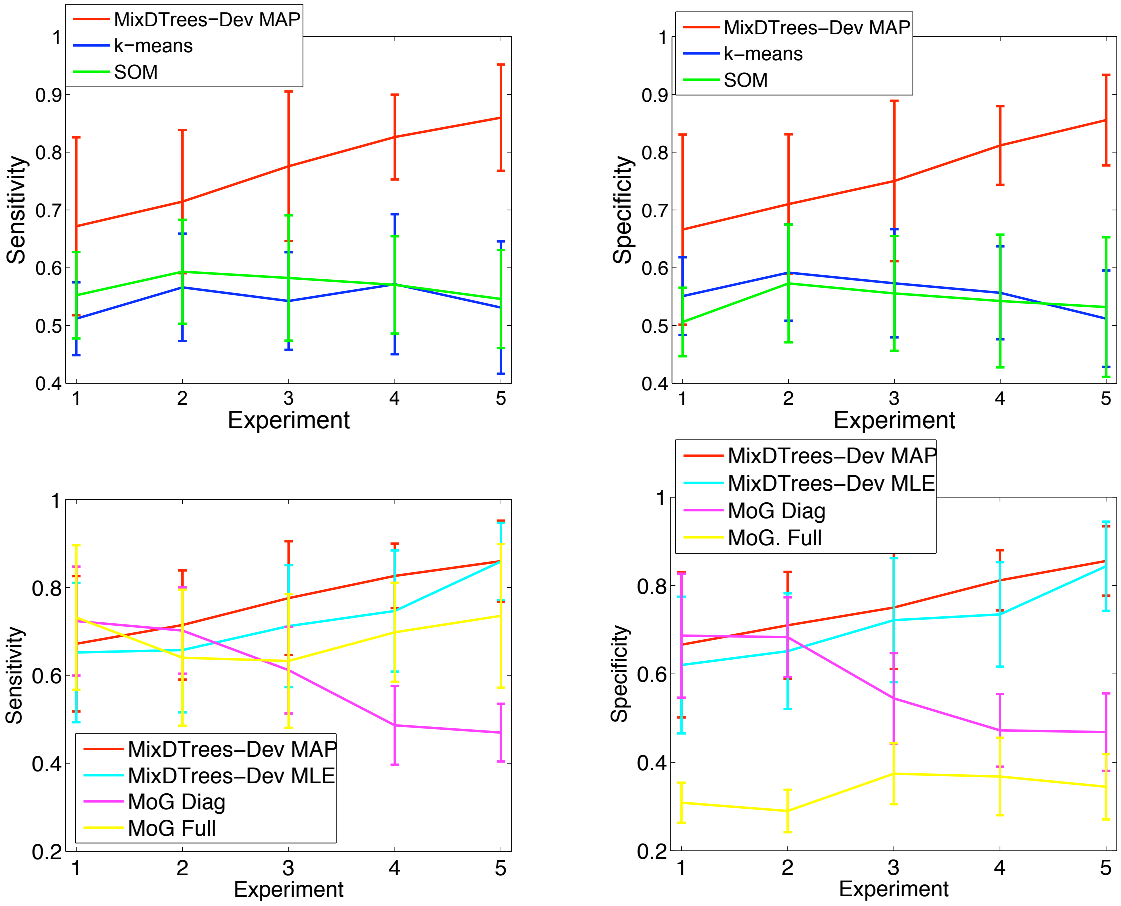


Figure 5.9: *We display the mean sensitivity (left plots) and mean specificity (right plots) against five experimental settings: (1) $w_{u|v,k} \in [-\epsilon, \epsilon]$ (independent data), (2) $w_{u|v,k} \in [-0.5, 0.5]$, (3) $w_{u|v,k} \in [-1, 1]$, (4) $w_{u|v,k} \in [-1.0, -0.5] \cup [0.5, 1]$ and (5) $w_{u|v,k} \in [-1, -1 + \epsilon] \cup [1 - \epsilon, 1]$. The dependence increases with experiment number.*

ences, we apply an one tailed paired t-test to evaluate the null hypothesis that two methods have the same mean specificity (or sensitivity) in a given experimental setting. Hereafter, for short, we simply state—method M_1 has a higher sensitivity than method M_2 (p-value below 0.05)—when the null hypothesis is rejected.

We observe that the `MixDTrees-Dev` with MAP estimates (`MixDTrees-Dev MAP`) have a higher specificity and sensitivity than k-means and SOM in all experimental settings (Figure 5.9 top) (p-value < 0.005). In the independent case ($w_{u|v,k} \in [-\epsilon, \epsilon]$), this is not expected, since the data agrees well with the assumptions of k-means and SOM. This also explains the large standard deviations of `MixDTrees-Dev MAP` in that case. As expected, the `MixDTrees-Dev MAP` clearly improves the cluster recovery in settings with noticeable dependence structure, while the performance of k-means and SOM deteriorates slightly.

In comparison to others mixture model methods (Figure 5.9 bottom), `MixDTrees-Dev MAP` also obtains a significantly higher specificity and sensitivity in almost all experimental settings. The mixture of Gaussians with diagonal covariance matrices performs well in the independent case (experimental setting 1), which meets the model assumptions, but it has poor results in experiments with higher dependence (p-values < 0.05 for settings 3, 4 and 5). The mixture of Gaussians with full covariance matrix (`MoG-Full`) has a reasonable sensitivity in all settings, but very poor specificity (p-value < 0.05 in settings 3, 4 and 5 for sensitivity and in all settings for specificity). The reason for these results is that `MoG-Full` tends to have some clusters with few data points, as a reflection of over-fitting [143]. Note that we use a MAP estimate for `MoG-Full` to minimize this problem. `MixDTrees-Dev` with MLE estimates (`MixDTrees-Dev MLE`) has good overall performance, but it is outperformed by `MixDTrees-Dev MAP` in all cases, except for experimental settings 1 and 5 (p-value < 0.05 for settings 2, 3 and 4). In experimental setting 5, where data are highly dependent, by definition, both methods work similarly.

These results demonstrate that the `MixDTrees-Dev` is a better alternative than SOM and k-means in all cases. In relation to other mixture models, `MixDTrees-Dev` represents a good trade-off between a complex model class, such as multivariate Gaussian with full covariance matrices, and the simple Gaussian with diagonal covariance matrices. Furthermore, MAP estimates of the `MixDTrees-Dev` represent a more robust alternative to the MLE counterpart.

5.4.2 `MixDTrees` with Estimated Structure

To evaluate the application of our method in real biological data, we make use of gene expression from lymphoid cell development. First, we compare a `DTree` inferred from the whole data with the lymphoid developmental tree. Then, we apply `MixDTrees-Str` to find modules of co-regulated genes, and evaluate the results with GO and KEGG enrichment analysis (Section 5.4.2). Finally, we compare our method with other unsupervised learning methods. Additionally, to investigate characteristics of `MixDTrees-Str` and compare it with other methods, we use simulated data from mixture models with different degrees of variable dependence.

Data

Lymphoid Tree (`LymphoidTree`). We produce an expression compendium of mouse lymphoid cell development by combining measurements of wild-type control cells from several studies [3, 156, 165, 220, 229] based on the Affymetrix U74 platform. Our data contain four stages of early development hematopoietic cells [3] (hematopoietic stem cell (HSC), pluripotent progenitor (PPP), common lymphoid progenitor (CLP), common myeloid progenitor (CMP)); three B cell lineage stages [220] (pro-B cells (Bpro), pre-B cells (Bpre) and immature B cells (Bimm)); one Natural Killer (NK) stage [165]; and four T cell lineage stages (double negative T cells (TDN) [156], cd4 T cells (TCD4), cd8 T cells (TCD8) and natural killer T cells (TNK) [229]). The developmental tree describing the

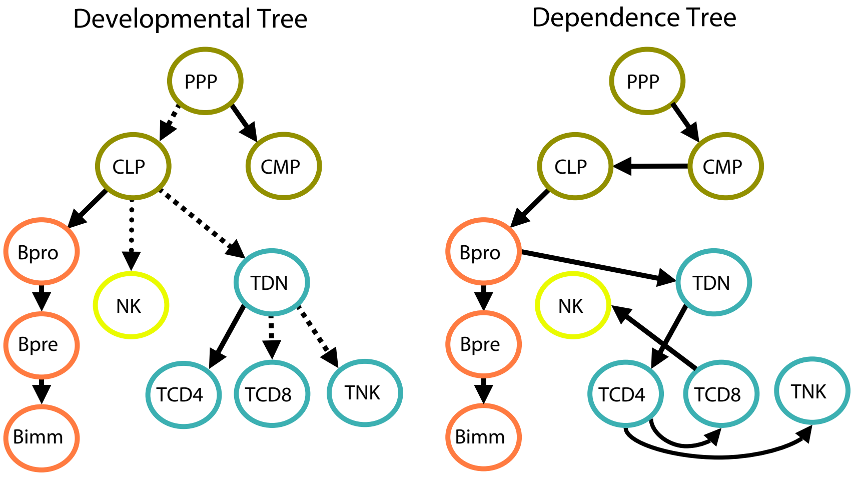


Figure 5.10: *We depict in the left the developmental tree with the stages contained in the Lymphoid data set. The dashed edges represent edges "wrongly" assigned in the* DTree *estimated from the Lymphoid data. Such edges connect pairs of vertices where the path length between these vertices in the developmental tree and estimated tree differs by one, while the dotted edge represents the case with path length differs by three. In the right, we have the* DTree *estimated from the Lymphoid data.*

order of differentiation of the cells is depicted in Figure 5.10 left. We pre-process the data as follows: we apply variance stabilization [104] on all chips, take median values of stages with technical replicates, use HSC values as reference values and transform all expression profiles to log-ratios. We keep genes showing at least a 2-fold change in one developmental stage. The final data set consists of 11 developmental stages and 3697 genes.

Simulated Data. We generate data from mixtures with four types of variable dependence ranging from: Gaussians with diagonal covariance matrix (Σ^{diag}), DTree with low variate dependence ($\Sigma^{DTree^{-}}$), DTree with high variate dependence ($\Sigma^{DTree^{+}}$) and Gaussians with full covariance matrix (Σ^{full}). These choices range from the independent case (Σ^{diag}) to the complete dependent case (Σ^{full}). For each setting, we generate ten such mixtures, and sample 500 development profiles from each. In all cases, we chose the μ from the range $[-1.5, 1.5]$, $L = 4$, $K = 5$ and mixture coefficients equal to $\alpha = (0.1, 0.15, 0.2, 0.2, 0.35)$. For Σ^{diag} , diagonal entries are sampled from $[0.01, 1.0]$, and non-diagonal entries are set to zero. For Σ^{DTree}, we randomly generate tree structures, one for each mixture component, and then chose $\sigma^2_{u|v,k}$ from $[0.01, 1.0]$ and $w_{u|v,k}$ from $[0.0, 0.5]$ for $\Sigma^{DTree^{-}}$ and $w_{u|v,k}$ from $[0.0, 1.0]$ for $\Sigma^{DTree^{+}}$. The generation of Σ^{full} is based on the eigenvalue decomposition of the covariance matrix ($\Sigma = Q\Lambda Q^T$) as in [168], where Λ is drawn from $[0.01, 0.5]$. The orthogonal matrix Q is obtained by sampling values from a lower triangular matrix M from the range $[20, 40]$, followed by the Gram-Schmidt Orthogonalization procedure.

We apply MoG with full and diagonal covariance matrices and MixDTrees-Str with

MLE and MAP estimates to all data sets. The mixture estimation method is initialized with $K = 5$ random DTrees (or Multivariate Gaussians) as described in Section 2.3.2. Next, we train the mixture model using the EM-algorithm. We also performed clustering with k-means [146], self-organizing maps (SOM) [221] and spectral clustering [154]. We compare the class information from the data generation to compute the corrected Rand index [103] and evaluate the clustering solutions.

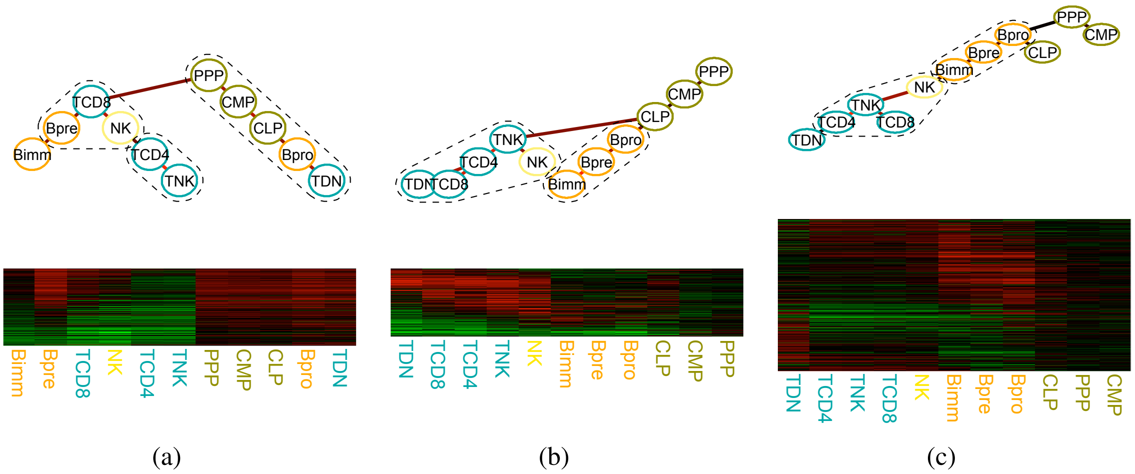
Enrichment of Gene Ontology and KEGG Pathways. Gene group validity is assessed by the results of Gene Ontology (GO) enrichment analysis [24], which helps the indication of functional roles of genes in a particular group. A more reliable and smaller alternative is the Kyoto Encyclopedia of Genes and Genomes (KEGG) [114], which has manually annotated gene pathways. In particular, several pathways related to lymphoid development such as signal transduction, immune system and cell cycle pathways, are described by KEGG. For the GO (or KEGG) enrichment analysis, we use the statistic of the Fisher-exact Test to obtain a list of GO terms (or KEGG pathways), whose participating genes are over-represented in a group as described in Appendix A.

Results

Inferring the DTree Structure. An initial question is how well we can recover the original developmental tree, as agreed upon by developmental biologists (Figure 5.10 left), if we apply the structure estimation method described in Sec 5.2.3 to the complete gene expression data (see Figure 5.10 right for the estimated DTree). To quantify the difference between these trees, we compute the path distance between all pairs of vertices, and calculate the Euclidean distance between the resulting distance matrices [202], which indicates a distance of 15.74. To assess the statistical significance of this distance, we generate 1000 random trees with the same distribution of out-degrees per vertex as the developmental tree. For each random tree, we compute the distance with the developmental tree. This test indicates a p-value of 0.002 of finding a distance as low as 15.74. Looking at these differences in detail, we can observe that 5 out of the 10 edges are correctly assigned, 4 edges connects vertices pairs with a path distance equal to 1, i.e., PPP and CLP, CLP and TDN, TDN and TCD8, and TDN and TNK, and one edge connect vertices with a path distance of 3 (NK is connected to TCD8 instead the CLP). Furthermore, “wrong” edges have a tendency to be connected to vertices in the same level of the developmental tree (e.g. TCD8 and TNK both connected with the TCD4).

Another important question is how well does the DTree capture dependence in the data? One simple way to assess this is to measure the proportion of the mutual information represented in the tree edges, in comparison to the total mutual information of all pairs of variables with the “treeness” index (Eq. 5.17).

For example, the score for the developmental tree (Figure 5.10 left) is 0.22, whereas for the estimated DTree (Figure 5.10 right), the “treeness” index is 0.42. For measuring the



(a)

(b)

(c)

Figure 5.11: *We depict the* `DTree` *and expression profiles of groups 1 (a), 4 (b) and 5 (c) from* `MixDTrees-Str MAP` *for the Lymphoid data. Dashed shapes around developmental stages represent the strongly connected components. See Section 5.3.3 for complete description of the plotting procedure.*

statistical significance of this, we generate random data by shuffling values of gene expression profiles x_i and estimating a `DTree` from this random data, which indicates a p-value of 0.0001.

Inferring Gene Modules with `MixDTrees-Str`. We estimate `MixDTrees-Str MAP` from the Lymphoid data following the protocol used for the simulated data. The Bayesian information criteria [145] indicates 13 groups as optimal.

First, we measure the average treeness of the `MixDTrees-Str` (we calculate Eq. 5.17 and take the sum weighted by α). For the `MixDTrees-Str MAP` this value is 0.54, which indicates an increase of 28% over the treeness index for the single `DTree`. This supports our claim that mixture of Dependence Trees with estimated structures is more successful in modeling dependencies in the data.

In relation to the groups of co-expressed genes found by `MixDTrees-Str`, in general, stages from the same developmental lineage are at same branches of the estimated `DTree` structure. Furthermore, groups present prototypical expression patterns such as over-expression in cells from a particular lineage, but not in other lineages (e.g., groups 2 and 5 for B cells, groups 4 and 6 for T cells and group 11 for Natural Killer cells) or groups displaying under-expression in particular lineages (e.g., groups 7 and 12 for T cells and groups 10 and 12 for B cells).

In Figure 5.11, we display some of these groups, which we discuss in more details. Group 1 (Figure 5.11 (a)) is an interesting case, where the `DTree` structure differs considerably from the developmental tree. On the right branch, we found a SCC (stages PPP, CLP, CMP, TDN, Bpro) with only early developmental stages, and all of them display high over-expression patterns. On the other hand, the majority of stages in the SCC on the left branch (Bimm, Bpre, TCD8, NK, TCD4, TNK) are immature developmental stages (leaves in the developmental tree depicted in Figure 5.10 left). Enrichment analysis using GO and KEGG shows that group 1 is over-represented for *cell cycle* and *dna repair* (p-values < 0.001). This matches the biological knowledge that earlier differentiation stages of development are cycling cells, while immature cells are resting [140, 177]. Group 4 (Figure 5.11 (b)) contains a SCC (left branch) with all T cell stages plus the closely related NK cell. At these stages, genes display an over-expression pattern. Enrichment analysis indicates over-representation for Gene Ontology terms as *T cell activation, differentiation and receptor signaling*; and KEGG pathways such as *T cell signaling* and *NK cell mediated cytotoxicity* (p-values < 0.001). Similarly, group 5 (Figure 5.11 (c)) has a SCC with all B cell stages. Furthermore, for B cell stages, genes are preferentially over-expressed. GO analysis indicates enrichment for terms such as *B cell activation* (p-values < 0.001), while KEGG analysis indicates enrichment in pathways such as *Hematopoietic cell lineage* and *B Cell receptor signaling* (p-values < 0.05). These results show how `MixDTrees-Str` can be used to find groups of biologically related genes, as the associated `DTree` structure adds relevant information regarding expression similarity of developmental stages.

Comparison with other Clustering Methods . For comparison purposes, we also perform clustering of the Lymphoid data with other methods: k-means, self-organizing maps (SOM), MoG with full covariance matrix, MoG with diagonal matrix and the bi-clustering methods Samba [210] and non-negative matrix factorization [31]. Additionally, we evaluate distinct variations of `MixDTrees`: `MixDTrees-Str` with MAP and MLE estimates, and `MixDTrees-Dev` with the developmental tree from Figure 5.10 (left) as structure.

To evaluate the performance of the methods, we use a heuristic of comparing p-values of KEGG enrichment analysis in a similar way as in [73]. The results of the comparison of `MixDTrees-Str MAP` and MoG diag can be see in Figure 5.12. In short, the best method should present a higher enrichment for a higher number of KEGG pathways. As illustrated in Figure 5.12, `MixDTrees-Str MAP` is superior to MoG diag in 9 out of 11 pathways. Furthermore, most of the 11 KEGG pathways enriched with a p-value < 0.05 in one of the methods (points depicted in Figure 5.12) are directly involved in immune system and developmental processes. We apply the same procedure for all pairs of methods and count the events $\{p\text{-value } \mathrm{m}_1 < p\text{-value } \mathrm{m}_2\}$, where m_1 and m_2 are the two methods in comparison. As can be seen in Figure 5.13 (left), `MixDTrees-Str MAP` outperforms all methods, while `MixDTrees-Str MLE` and k-means also obtained higher enrichment than other methods. Overall, SOM, MoG Full and Samba obtain poor enrichment results. In fact, these methods are outperformed by all other methods. We repeat the same analysis for GO enrichment (see Figure 5.13 right). The result are in agreement with the KEGG enrichment analysis, that is, `MixDTrees-Str MAP` has higher enrichment than all other methods, while SOM and MoG Full obtain poor results.

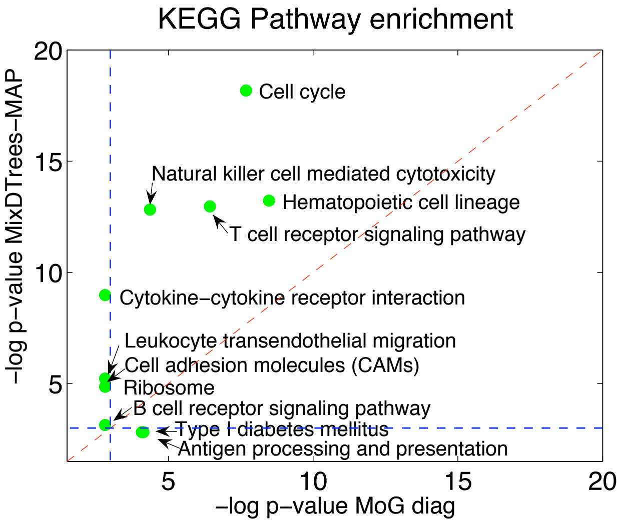


Figure 5.12: *We depict the scatter plot comparing the KEGG pathway enrichment of* `MoG diag` *(x-axis) and* `MixDTrees-Str-MAP` *(y-axis). We use $-log(p)$-values, where higher values indicate a higher enrichment. The blue lines correspond to $-log(p)$-value cut-off used (p-value of 0.05). Only KEGG pathways with a $-log(p)$-value higher than (2.99) in one of the results are included.* `MixDTrees-Str-MAP` *has a higher enrichment for 9 out of the 11 KEGG pathways.*

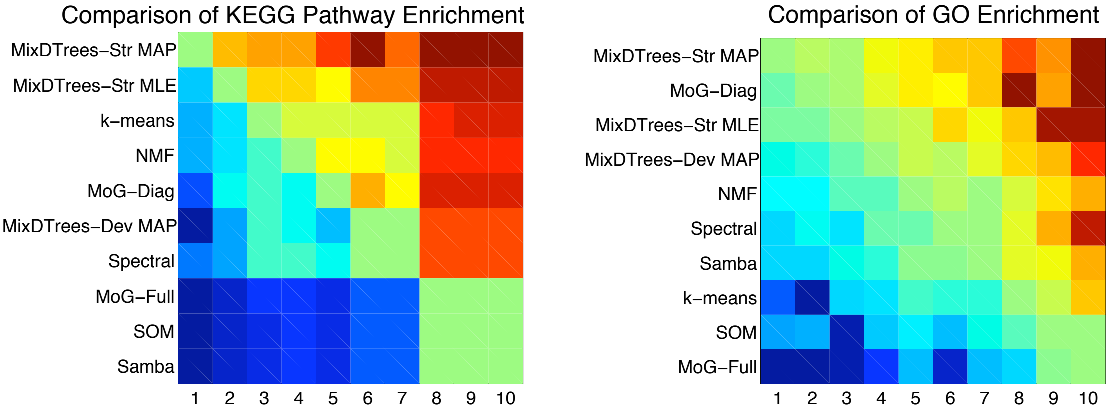


Figure 5.13: *Heat-maps plot displaying the comparison of KEGG (left) and GO (right) enrichment for 10 distinct clustering methods. Red (or blue) values indicate that the method in the y-axis has a higher (or lower) count of enriched KEGG pathways (GO terms) than the method on the x-axis. The numbers on x-axis correspond to the methods in the y-axis.*

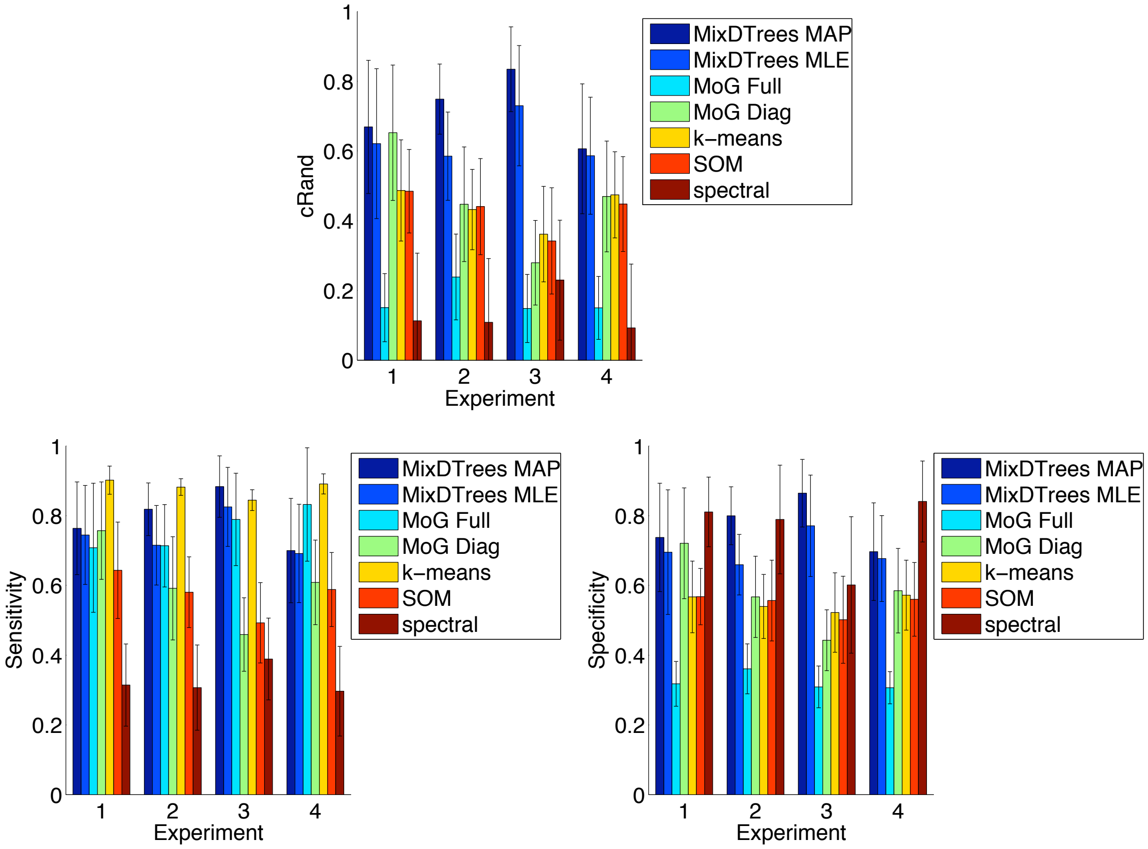


Figure 5.14: *We depict the mean corrected Rand (top), sensitivity (bottom left) and specificity (bottom right) of true label recovery for distinct clustering methods (y-axis) against data generated with distinct model assumptions (x-axis) (1 for Σ^{diag}, 2 for Σ^{DTree^-}, 3 for Σ^{DTree^+} and 4 for Σ^{full}). These choices range from the independent case Σ^{diag} to the complete dependent case Σ^{full}.*

Simulated Data. As expected, every method performs well on the data generated with the corresponding model assumptions (see Figure 5.14). An exception is the MoG with full covariance matrices, which has low corrected Rand for all data sets. An analysis of the specificity index indicates that the poor performance of MoG Full is caused by over-fitting, since it tends to merge real groups (see Figure 5.14 bottom right). Moreover, spectral clustering presents very low sensitivity values (see Figure 5.14 bottom left), which indicates a tendency to split real groups. In both data from Σ^{DTree}, MixDTrees-Str MAP has higher values than MixDTrees-Str-MLE, which indicates a higher robustness of the MAP estimates (a paired t-test indicated superiority of MixDTrees-Str MAP with p-value < 0.05 in both Σ^{DTree^-} and Σ^{DTree^+}). Also, MixDTrees-Str MAP obtains the highest values in all settings (p-value < 0.05), outperforming MoG Full, MoG Diagonal, k-means, SOM and spectral clustering, with the exception of MoG Diagonal in the Σ^{diag} data. These results show that MixDTrees-Str-MAP has a better performance than compared methods in data coming from distinct dependence structures, and it is robust against over-fitting.

Chapter 6

Clustering with Constraints for Integration of Heterogeneous Biological Data

The transcriptome of cells measured with microarrays gives an important and informative snapshot of the genetic information flow. However, it only reflects one particular aspect of the cell control dynamics: the number of specific RNA molecules present in a cell. Recently, several other large-scale technologies, which explore distinct aspects of the cell information flow, became available. For example, protein-protein interaction screens reveal the composition of proteins complexes [83, 108]; chromatin immunoprecipitation experiments detect where a particular protein binds in DNA genomic regions [128]; and in-situ hybridization techniques elucidate the spatial patterns of gene expression within an organism [214]. Other useful sources of large-scale data are biological databases. For example, Gene Ontology is a controlled vocabulary of biological concepts and gene annotations [9]; the Kyoto Encyclopedia of Genes (KEGG) catalogs manually annotated biological pathways [114]; and PubMed indexes titles and abstracts of most biological and medical journals [167]. Combining one (or more) biological sources of information with gene expression data is a natural next step to achieve better functional hypotheses. Indeed, several methods have been proposed for this problem (see [217] for a general review). Among others, probabilistic methods have been widely applied in this context, since they are flexible, can be easily extended to accommodate new data sources, and allow a statistical evaluation of the results [15, 192–194, 209, 218, 231].

We propose in this chapter the use of a simple, intuitive and mostly assumption-free framework of semi-supervised learning for the joint analysis of data from heterogeneous biological sources [39]. Semi-supervised learning is appropriate if there is a number of labels available for some of the observations, while the majority of data points carry no label. The main idea is to take advantage of both the labeled (supervised) and unlabeled data (unsupervised) in order to obtain better estimates than when analyzing each data source separately. For example, in [187] it was shown that few high quality labeled genes were able to improve the clustering of gene expression time courses, in comparison to a purely unsupervised method. One particular type of semi-supervised learning is called clustering with constraints, or constrained clustering. It only makes weak assumptions about the labels by encoding secondary information as pairwise constraints. These methods search for clustering solutions, which violate the fewest number of constraints. We can, for example, derive constraints from Gene Ontology annotation (GO) [9] by constraining pairs of genes with similar GO annotation to be in the same cluster. Likewise, we can also constrain pairs of genes with distinct GO annotations to be in different clusters (negative constraints). The use of clustering with constraints for integration of heterogeneous data is based on two assumptions not explored by previous approaches [15, 192, 209, 218, 231]: (1) the secondary information is usually not available for all genes from expression experiments; and (2) gene expression data sets provide one view of the biological process under investigation, which is very unlikely to provide the same level of detail as in the secondary information. Using additional data as secondary information, we simply limit the gene expression based clustering results to biologically more plausible solutions.

In this chapter, we investigate the use of clustering with constraints for finding groups of co-expressed genes with the aid of secondary information. First, we describe related work in Section 6.1. A general formulation of the clustering with constraints problem will be introduced in Section 6.2. In Section 6.2.1 we describe the method previously proposed in [123], which we adopt in our biological applications. One contribution of this chapter is an experimental analysis of data sets commonly used in studies integrating heterogeneous biological data. The main purpose of this analysis is to evaluate the feasibility of clustering with constraints in this problem scenario [52]. We apply the clustering with constraints to yeast cell cycle data [42], using either Gene Ontology [9] or transcription factor location analysis [128] as secondary information (see Section 6.2.2 for constraints definitions). As the yeast cell cycle data set has full class labels, we can evaluate the improvements resulting from the addition of the secondary information in the analysis (see Section 6.3.1 for results). The second contribution of this chapter is a novel bioinformatics application for finding syn-expressed genes [48]. More precisely, we analyze gene expression time courses of Drosophila development using in-situ RNA hybridization images as secondary data. The constraints derived from the in-situ data are described in Section 6.2.2 and the results are presented in Section 6.3.2. Finally, we present a discussion and future work in Chapter 7.

6.1 Related Work

Semi-supervised learning (SSL) is a topic of great interest in the machine learning community [39]. SSL methods try to combine characteristics of supervised and unsupervised learning methods in problem scenarios where only part of the observations are labeled. Such data arises in many practical applications. For example, in text categorization problems, it is easy to retrieve thousands of texts from the web, but manually labeling texts is expensive [45]. Similarly, for gene expression derived from microarrays, we have the measurements of the transcription of whole genomes, but only a small fraction of genes have a functional characterization [187]. One can implement semi-supervised learning with different machine learning paradigms [39]: transductive learning, such as transductive support vector machines [112]; graph-based approaches, such as spectral methods [207]; methods based on change of representations, which use labeled data to recompute distance matrices [118, 228]; and generative models, such as probabilistic clustering methods [19]. We are mainly interested in the latter category, as they can be used together with the mixture model framework used in other chapters of this thesis.

Semi-supervised clustering methods consider SSL from an unsupervised learning point of view. In particular, one assumes that the total number of classes and the coverage of labels in these classes are both unknown [19]. With generative models, we view the clustering problem in a probabilistic setting, and include the constraints in the model prior, in order to restrict the solution space to clustering solutions respecting the constraints derived from class labels. The semi-supervised clustering problem can be described in a complete likelihood formulation and be solved with extensions of the EM algorithm (see Section 6.2). One alternative is to use the labels as hard constraints [45, 161, 185]. A more flexible, simple and assumption free approach is to consider only constraints between pairs of objects. Most methods of clustering with constraints are based on defining a hidden random Markov field (HRMF) in the constraints [153]. They employ distinct approximation methods for estimating the posterior assignment of the EM algorithms. Among other proposals, there are: chuncklet model [196], iterated conditional modes [19], Gibbs sampling, [137], mean field approximation [123], and re-sampling chunklet model [153]. The work in [153] performed a comparative analysis of the previous methods [19, 123, 137, 153, 196] with benchmarking data sets, and with the inclusion of noise in the constraints. In general, methods like [19, 123, 153] performed well after the addition of noise, while the exact method based on hard constraints [196] had poor results. This is explained by the fact that particular sets of "hard" constraints will have no feasible solutions for a specific number K of clusters [57]. For example, the constraints in Figure 6.1 (c) cannot be satisfied for $K = 2$. Thus, exact methods should be avoided, such the one in [196], when one expects errors in the constraints. On the other hand, [153] shows that approximate methods, such as [19, 123, 137], which are based on local update rules of the posterior assignments, can get easily trapped in local maximum solutions, in particular when large constraint weights are used. The use of distinct Bayesian classifiers in a semi-supervised clustering with hard labels was proposed in [45]. The authors investigated the effects of the size of labeled and unlabeled data on UCI benchmark data sets. Their results showed that unlabeled data can deteriorate the overall results, if the assumptions of the model do not match the distribution of the data. They suggest that cross-validation on labels (or constraints) is a relevant approach for performing model selection.

Analysis of heterogeneous biological data has been tackled with several distinct methodologies. See [218] for a broad review of the area. We describe below only those studies based on semi-supervised methods. In [185, 187], it was shown how a few number of high quality labels ($< 2\%$ of observations), which were used as hard labels in a mixture model, could improve clustering of gene expression time courses. In [193], a gene expression data set was analyzed in conjunction with protein-protein interaction data. The author also proposed a model-based clustering method with a HRMF over the protein-protein interaction graph. A belief network propagation method was used for estimation of the posteriors. In [198], pathway information from KEGG was modeled also as a HRMF, which was estimated with the interactive conditional modes method. In [194], gene expression data was analyzed together with transcription factor binding site (TFBS) data with an EM based method. Also, a model-based approach similar to [187] was proposed in [161] for clustering gene expression data with labels derived from functional annotation data. That work, however, makes an *ad hoc* selection of few functional classes used as labels, and ignores the fact that genes can be assigned to multiple functions. The same authors also investigated the use of a semi-supervised method based on the modification of the distance function according to the labeled data on similar data sets [102]. Furthermore, [189] performed a case study using the mean-field approximation for clustering with constraints [123]. They used a fully labeled yeast cell cycle data set (as in the study described in Section 6.3.1) and TFBS data for deriving the constraints. They could show that with a more conservative choice of constraints the TFBS data yielded improvements in the recovery of Gene Ontology terms.

The work presented in this chapter differs from [102, 161, 185, 187], as they are all based on hard constraints and ignore the existence of noise in the constraints. In relation to [189, 193, 198], all share a similar computational method with the one used in this thesis, but they differ in the data used as secondary information.

In the context of syn-expression, [214, 215] performed a large-scale study of gene expression in the Drosophila embryos by in-situ RNA hybridizations. The images were manually curated and annotated using a controlled vocabulary—ImaGO—following the example of the Gene Ontology [9]. The final result was a hierarchical clustering of genes based on the manual annotations; the gene expression time-courses were not included in the analysis. Recently, a similar study was performed in Drosophila embryogenesis using high-resolution fluorescent in-situ hybridization technique [127]. This technique allows the sub-cellular location of expression. They also extended the vocabulary from ImaGO to include sub-cellular location terms. Recently, studies investigated pattern formation in Drosophila based on 3D in-situ images [96, 117] for a small number of genes. Further work concentrated on mining the image database for genes with a spatial expression pattern similar to a query [160, 163] and on the extraction of relevant features in the images [160], for example by clustering images on an eigenvector based representation [162]. All these syn-expression studies restricted themselves to the analysis of the images with gene expression location. In contrast, the application proposed in this chapter is the first one combining gene expression from microarrays with gene expression location for deriving groups of syn-expressed genes.

Recently, [181] proposed the use of gene expression time courses of Drosophila development as an input for a classifier distinguishing modules of gene expression location. The modules of expression location were derived from the manual annotation of in-situ patterns from [214] and no image processing was performed.

6.2 Mixture Model Estimation with Constraints

The main idea of clustering with constraints is to include additional data in the form of pairwise constraints in order to restrict or penalize particular cluster solutions. These constraints can be of two types: *positive* constraints, which indicate that two objects should be in the same cluster, and *negative* constraints, which indicate that two objects should be in separate clusters. Moreover, the constraints can be interpreted in two ways: “hard constraints”, which have to be fulfilled in the solutions, and “soft constraints”, which might be violated. For the latter, a penalty violation value can be defined for each pairs of objects. See Figure 6.1 for an example of how the “hard” and “soft” pairwise constraints can be used to restrict clustering solutions.

In this chapter, we are interested in probabilistic methods using “soft constraints” [123, 137]. One way to achieve this is to extend the basic EM approach (Section 2.3.1) to include the constraints. In the following, we describe the basic formalism of this extension. In Section 6.2.1, we describe one particular method for performing mixture model estimation with soft constraints.

Formally, for a data set $\mathbf{X}$ with N observations, we specify the positive constraints as a matrix W^+, where $w^+_{ij} \in [0, \infty]$ is the positive constraint penalty for the pair of observations i and j ($1 \leq i \leq N$ and $1 \leq j \leq N$). Likewise, we specify a negative constraints matrix W^-, where $w^-_{ij} \in [0, \infty]$. We use W to denote the pair (W^+, W^-). Recalling Section 2.3.1, the EM algorithm is based on maximizing the complete data likelihood (Eq. 2.7),

$$\mathbf{P}(\mathbf{X}, \mathbf{Y}|\Theta) = \mathbf{P}(\mathbf{X}|\mathbf{Y}, \Theta)\mathbf{P}(\mathbf{Y}|\Theta),$$

where $\mathbf{Y}$ indicates the cluster assignments of observations in $\mathbf{X}$.

The constraints can be added into the previous equation making the prior of the cluster assignments $\mathbf{Y}$ to be dependent on W,

$$\begin{aligned}
\mathbf{P}(\mathbf{X}, \mathbf{Y}|\Theta) &= \mathbf{P}(\mathbf{X}|\mathbf{Y}, \Theta)\mathbf{P}(\mathbf{Y}|\Theta, W), \\
&= \mathbf{P}(\mathbf{X}|\mathbf{Y}, \Theta)\mathbf{P}(\mathbf{Y}|\Theta)\mathbf{P}(W|\mathbf{Y}, \Theta).
\end{aligned}$$

The only term depending on the constraints is $\mathbf{P}(W|\mathbf{Y}, \Theta)$. This can be interpreted as a weighting function penalizing cluster assignments $\mathbf{Y}$, which violate the constraints W. As it is common in probabilistic clustering with constraints methods [39], we assume that the constraints impose a hidden Markov random field (HMRF) on the (hidden) variable Y representing the unknown cluster assignments. In short, a hidden Markov random field is a graphical representation of the joint distribution of a hidden variable. The HMRF assumes that the conditional distribution of the variables obeys the Markov property, i.e., the probability of a variable is only dependent on neighboring variables (see [131] for a complete description of HMRF). In our context, the HMRF graph is represented by a set of nodes, where node i represents the observation y_i, and the neighborhood graph is represented by the constraints, where w_{ij} indicates the weight of the edge between nodes i and j. Hence, it follows from [92] that the prior probability of a particular cluster assignment $\mathbf{Y}$ follows a Gibbs distributions,

$$\mathbf{P}(W|\Theta, \mathbf{Y}) = \frac{1}{Z} \exp\left(\sum_{i}^{N} \sum_{j \neq i}^{N} -w_{ij}^{+} \mathbf{1}(y_j \neq y_i) - w_{ij}^{-} \mathbf{1}(y_j = y_i) \right), \tag{6.1}$$

where $\mathbf{1}$ is the indicator function and $Z = \sum_{\mathbf{Y} \in \mathcal{Y}} \mathbf{P}(W|\Theta, \mathbf{Y})$ is the normalizing function.

In this formulation, however, we cannot assume independence between cluster assignments $\mathbf{Y}$ in the E-step, as it is required by EM algorithm (Section 2.3.1). Exact inference of the posterior would now require the complete evaluation of the following equation

$$\mathbf{P}(y_i = k|\mathbf{X}, \Theta, W) = \sum_{\mathbf{Y} \in \mathcal{Y}_{y_i = k}} \mathbf{P}(\mathbf{X}|\mathbf{Y}, \Theta) \mathbf{P}(\mathbf{Y}|\Theta, W), \tag{6.2}$$

where $\mathcal{Y}_{y_i=k}$ is the space of all cluster assignments $\mathbf{Y}$ and y_i is fixed to the value k. Several approximations have been proposed for estimating the posterior, such as the chuncklet model [196], iterated conditional modes [19], Gibbs sampling, [137], and mean field approximation [123]. We adopt the approach in [123], as it allows for modeling soft-constraints, does not require sparsity of the matrices W^{+} and W^{-}, and performs well on benchmarking [153].

Note also that in this formulation, as $\mathbf{P}(\mathbf{X}|\mathbf{Y}, \Theta)$ is independent of W, no modification is required in the M-Step of the EM algorithm. As a result, the component models proposed in Chapter 4 and 5 can be used in this clustering with constraints setting.

6.2.1 Mean Field Approximation

It was shown in [123] that the distribution in Eq. 6.1. follows the Maxent principle,

$$\mathbf{P}(W|\Theta, \mathbf{Y}) = \frac{1}{Z} \exp\left(\sum_{i}^{N} \sum_{j \neq i}^{N} -\lambda^{+} w_{ij}^{+} 1\{y_j \neq y_i\} - \lambda^{-} w_{ij}^{-} 1\{y_j = y_i\} \right)$$

where λ^{+} and λ^{-} are Lagrange parameters defining the penalty weights of positive and negative constraint violations.

A mean field approximation is used in the inference of the posterior distributions from

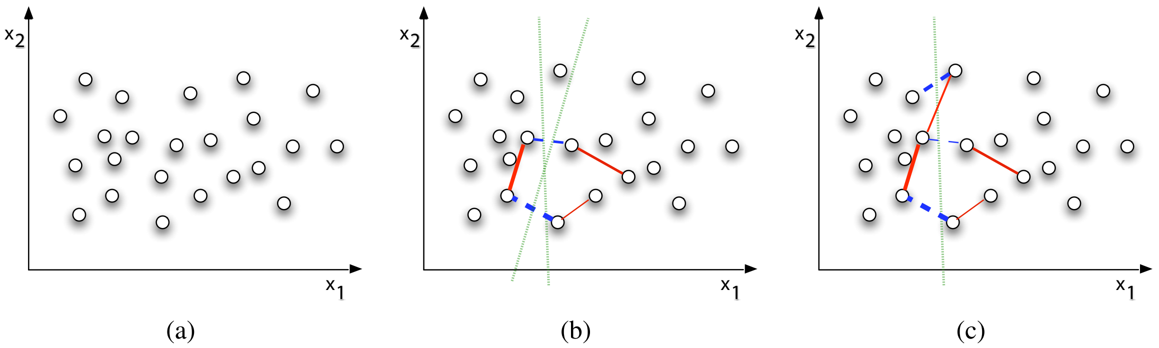


Figure 6.1: *The effectiveness of the use of pairwise constraints, cases (b) and (c), is shown by contrasting them with the unsupervised case (a). Assuming a two-dimensional space, it is hard to distinguish the two clusters from the data points alone, and the boundary between them (a). The addition of positive pairwise constraints, depicted as red edges, and negative constraints, depicted as blue edges (b), indicate the existence of two or more clusters and possible cluster boundaries, depicted as green dotted lines. In (c), there is no boundary, which respects all constraints, and methods based on "hard" constraints would fail in this scenario. With the use of "soft" constraints, where the penalty of constraint violation is proportional to the edge widths, there is an optimal solution (green dotted line), which violates one positive constraint, in the cost of respecting a negative constraint with higher penalty value (or edge width).*

the given HMRF. Formally, the posterior distribution is approximated with a factorial distribution $q(\mathbf{Y}) = \prod_{i=1}^{N} q_i(y_i)$, by minimizing the relative entropy of the real posterior distribution $\mathbf{P}(\mathbf{Y}|\mathbf{X}, \Theta, W)$ (Eq. 6.2),

$$q^* = \operatorname*{argmin}_q \sum_{\mathbf{Y} \in \mathcal{Y}} q(\mathbf{Y}) \log \left(\frac{q(\mathbf{Y})}{\mathbf{P}(\mathbf{Y}|\mathbf{X}, \Theta, W)} \right)$$

where $\sum_{k=1}^{K} q_i(y_i = k) = 1$.

As demonstrated in [123], the posterior assignments is approximated as follows

$$q_i(y_i = k) = \frac{\alpha_k p(x_i|y_i = k, \theta_k)}{\sum_{k'=1}^{K} q_i(y_i = k')} \exp \left(\sum_{j \neq i} -\lambda^+ w_{ij}^+ (1 - q_j(y_j = k)) - \lambda^- w_{ij}^- q_j(y_j = k) \right).$$

where α_k is defined as in Eq. 2.22 and $p(x_i|y_i = k, \theta_k)$ is the pdf of the component model (see Eq. 2.24 for the multivariate Gaussian case).

Note that this formulation allows several alternatives regarding the use of constraints. When there is no overlap in the annotations, or more precisely $w_{ij}^+ \in \{0, 1\}$, $w_{ij}^- \in \{0, 1\}$, $w_{ij}^+ w_{ij}^- = 0$, and $\lambda^+ = \lambda^- \sim \infty$, we obtain hard constraints. Alternatively, by fixing $\lambda^+ = 0$ (or $\lambda^- = 0$), we make use of only positive (or negative) constraints.

6.2.2 Deriving Constraints

We describe in this section how we can derive constraints from biological information.

Gene Ontology

The Gene Ontology (GO) project is a collaborative effort to address the need for consistent descriptions of gene products in different databases [9]. Three structured controlled vocabularies (ontologies) describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner. Cellular component describes biological compartments in which genes are active (e.g., *rough endoplasmic reticulum*); molecular function contains concepts related to gene function (e.g., *catalytic activity*); and biological process describes the processes that a gene can take part of (e.g., *cellular physiological process*).

Formally, a given Gene Ontology (GO) is represented by a directed acyclic graph (DAG), in which each node t_i in a set $T = \{t_1, ..., t_M\}$ represents a biological term (controlled vocabulary or GO term) and the edges stand for relationships among these terms. A relationship $R(t_i, t_j) \in \mathcal{R}$ indicates that term t_i is a parent of term t_j. Such a relation is interpreted as t_j being a subclass of t_i, i.e., t_i is a more general concept than t_j. For instance, the biological term "*cell cycle*" is related to the more specific terms "*mitotic cell cycle*" and "*meiotic cell cycle*".

A set of genes $G = \{g_1, ..., g_N\}$ is related to a given GO term by an annotation set $\mathcal{A}$, where $A(t_i, g_n) \in \mathcal{A}$ indicates that gene g_n is annotated with term t_i. Genes often have multiple biological roles, hence they are usually annotated with several GO terms. Furthermore, the parent-child relation of GO implies that genes annotated with a term are also annotated with all parents of this term. That is, for all $R(t_i, t_j) \in \mathcal{R}$, given a gene g_n, $A(t_j, g_n)$ implies that $A(t_i, g_n)$.

The intuition for the use of Gene Ontology as a secondary data is that genes participating in the same biological process should be co-expressed [71]. Hence, we positively constrain genes annotated with the same GO terms, and negatively constrain pairs of genes annotated with distinct GO terms.

More formally, let $D(g_i) = \{t | A(t, g_i) \in \mathcal{A}, t \in T\}$ be the set of GO terms annotating g_i. We can define constraints by calculating the number of GO terms common to a pair of genes. That is, for all pairs of genes g_i and g_j (corresponding to the observations x_i and x_j in $\mathbf{X}$), we define the following constraints

$$w_{ij}^{+} = \frac{\#D(g_i) \cap D(g_j)}{\#D(g_i) \cup D(g_j)}, \tag{6.3}$$

and

$$w_{ij}^{-} = \frac{\#D(g_i) \uplus D(g_j)}{\#D(g_i) \cup D(g_j)}. \tag{6.4}$$

where w_{ij}^{+} will take values in $[0, 1]$ with $w_{ij}^{+} = 1$ indicating perfect agreement for positive constraints and $w_{ij}^{-} = 1$ perfect disagreement for negative constraints. Non-annotated genes have constraints equal to zero.

Location Analysis

Location analysis allows the detection of the binding sites of transcription factors (TF) in a genomic scale [128]. The binding of a TF to an upstream region of a gene is a pre-requisite and indicator that regulation occurs. Similarly as in the case of Gene Ontology, pairs of genes being bound by the same transcription factor are likely to be co-regulated [212].

For a set of transcription factors $F = \{f_1, ..., f_M\}$, location analysis will return relations $A'(f_l, g_i) \in \mathcal{A}'$, which indicates that factor f_l binds to g_i. Let $D(g_i) = \{f_m | A'(f_m, g_i) \in \mathcal{A}, f_m \in F\}$ be the set of TFs bound to g_i. Then, we can use Eq. 6.3 and Eq. 6.4 to obtain constraints.

In-Situ Images

An important aspect of gene expression, which has been studied in great detail in embryonic development of Drosophila melanogaster [214], is its precise localization. While the initial motivation for these sensitive experiments is to understand the role of individual genes in organ development, we can incorporate spatial expression patterns with gene expression time courses from microarrays for improving the generation of functional hypotheses.

In fact, genes that share the same temporal-spatial expression pattern are more likely to form a functional module [157]. If they are synchronously co-expressed in one tissue, or in multiple tissues, this is refereed to as *syn-expression* [157]. The spatial expression patterns can be determined with in-situ experiments where a mRNA-specific stain is produced by mRNA-binding oligonucleotides and a suitable dye [211]. Then, image analysis produces either 2D or 3D images of spatial patterns of gene expression. Drosophila embryos are morphologically rather simple, however the image analysis task is not trivial as in-situ images are taken of many subjects with large fluctuations in shape. In addition, the staining intensity has higher, gene-specific error rates compared to DNA microarrays [214].

To compare in-situ hybridization patterns of a pair of registered embryo images [159], we compute the Pearson correlation as a co-location index, as proposed in [159]. This index takes both the spatial distribution and the strength of hybridization into account. Despite its simplicity, this index had comparable performance to a more complex method previously described in [163].

More formally, let Z be an L-dimensional continuous random variable defining the pixel intensities of an image with L pixels. For a data set of images $\mathbf{Z}$, where z_i and z_j describe the pixel intensities of two registered embryo images; and z_i is an L-dimensional vector

$(z_{i1}, ...z_{il}, ..., z_{iL})$, the Pearson correlation coefficient is calculated as follows

$$\text{PC}(z_i, z_j) = \frac{Cov(z_i, z_j)}{\sqrt{Var(z_i)}\sqrt{Var(z_j)}}, \tag{6.5}$$

where $Var(z_i) = \sum_{l=1}^{L}(z_{il} - \mu_i)^2/L$, $Cov(z_i, z_j) = \sum_{l=1}^{L}(z_{il} - \mu_i)(z_{jl} - \mu_j)/L$, and $\mu_i = \sum_{l=1}^{L} z_{il}/L$.

Note that there is no annotation of the orientation of the embryo. Furthermore, automatic registration of the image is a difficult task. Hence, for each pair of images, we estimate the correlation between all possible orientations and take the maximum correlation value.

For a given gene, we have in-situ images for several developmental periods, and for each period and gene we have zero or more in-situ images. Formally, let $I_i = \{I_i^1, ..., I_i^t, ..., I_i^T\}$ indicate the sets of in-situ images related to gene i and time periods 1 to T, and let $I_i^t = \{z_1, ..., z_m, ..., z_M\}$ be the set of images related to gene i at period t. For a pair of genes and a developmental period, we compute the Pearson correlation (Eq. 6.5) for all pairs of images in sets I_i^t and I_j^t; and keep the maximum value. This yields the co-location index (CL)

$$\text{CL}(I_i^t, I_j^t) = max_{z_m \in I_i^t, z_n \in I_j^t} \text{PC}(z_m, z_n). \tag{6.6}$$

By an inspection of the distribution of the co-location index, we select a value s of gene pairs to constrain. In other words, for all pairs of genes (i, j) at period t, the sth highest $\text{CL}(I_i^t, I_j^t)$ values are positively constrained ($w_{ij}^{t+} = 1$). Similarly, the pairs (I_i^t, I_j^t) with sth lowest CL values are negatively constrained ($w_{ij}^{t-} = 1$). Using this criterion, we obtain a constraint matrix W^{t+} (or W^{t-}) for a particular developmental period t.

As a last step, we need to combine the constraints from the distinct developmental periods. See Figure 6.2 for an example. We require that a pair of genes is only constrained if it is constrained in at least p developmental periods

$$w_{ij}^{+} = \begin{cases} 1, & \sum_{t=1}^{T} w_{ij}^{t+} \geq p \\ 0, & \text{otherwise} \end{cases} \text{, and} \tag{6.7}$$

$$w_{ij}^{-} = \begin{cases} 1, & \sum_{t=1}^{T} w_{ij}^{t-} \geq p \\ 0, & \text{otherwise} \end{cases} . \tag{6.8}$$

6.3 Experiments

In this section, we describe the application of clustering with constraints in two different data sets. In the first case, for a proof of concept evaluation, we use a simple benchmarking data set—yeast during cell cycle—, which is also analyzed in Chapter 4. We use either Gene Ontology or location analysis information as secondary data. For the case of the

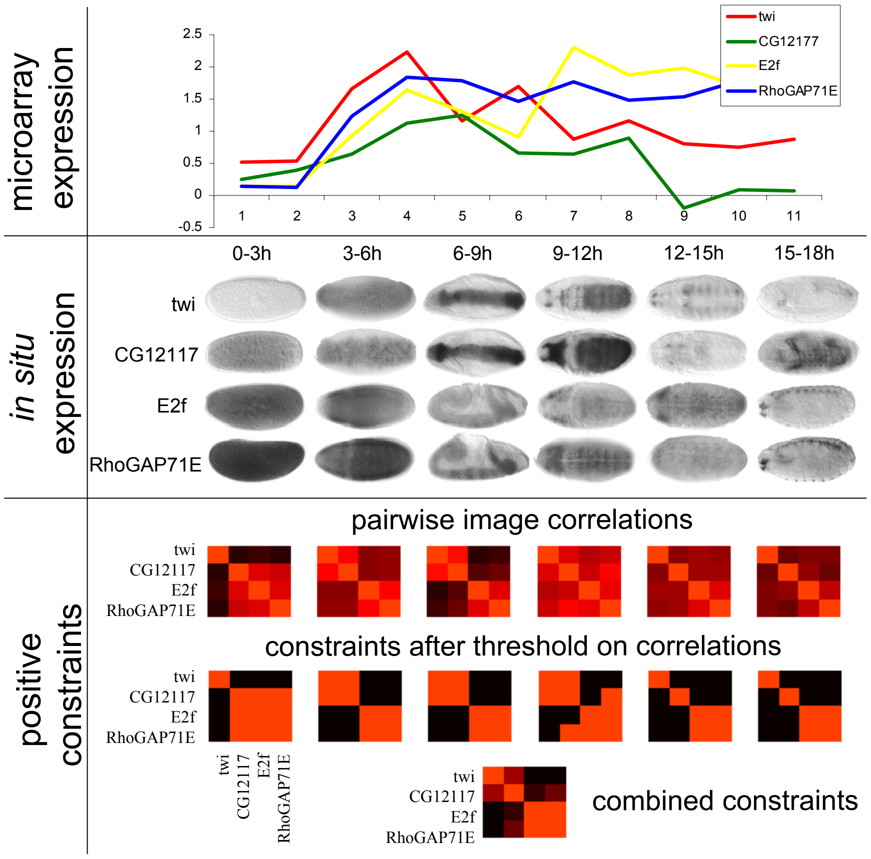


Figure 6.2: *Time course expression (top) and registered in-situ images (middle) of 4 genes twi, CG12177, Ef2 and RhoGAP71E indicate the gene expression patterns. From left to right, the embryo images are categorized into the time periods 0-3, 3-6, 6-9, 9-12, 12-15 and 15-18h. The time-courses display similar expression patterns with maximal expression after 3 hours for all genes, but weakly diverging pattern at later time points. The in-situ images indicate that twi and CG12177 have syn-expression at time periods 3-6, 6-9 and 9-12h; while Ef2 and RhoGAP71E have syn-expression at time periods 0-3, 3-6, 6-9, 9-12 and 15-18h. At the bottom, we display how positive constraints are derived from in-situ hybridization patterns. Heat-maps display the correlation coefficients between all pairs of in-situ images of the corresponding time period (red values indicate positive correlations). A constraint matrix for each time period is obtained by thresholding the corresponding correlation matrix. For example, constraint matrices from periods 3-6 and 6-9h indicate syn-expression of pairs (twi, CG1217) and (Ef2, RhoGAP71E), whereas the constraint matrix from period 9-12h indicates that (CG1217, RhoGAP71E) are syn-expressed. Matrices are combined into one, which constrains genes that display syn-expression in at least 3 periods, as indicated in the matrix at the bottom.*

second data set, we present a more detailed and exploratory analysis of Drosophila development. In this context, we use gene expression time courses as the main data set and information from in-situ images as secondary data.

6.3.1 Yeast Cell Cycle with Gene Ontology and Location Analysis

We use the expression profiles of 384 genes during Yeast mitotic cell division assigned to one of the five cell cycle phases classes [42], which we refer to as YCC. See section 4.5.1 for a detailed data description. Although this data set is biased towards profiles showing periodic behavior, and some of the class assignments are ambiguous, it is one of the few data sets with a complete expert labeling of genes.

The relation between regulators and target genes are obtained from large-scale location analysis, comprising data from 142 candidate TFs [128]. Relations $A'(f_l, g_i) \in \mathcal{A}'$ are obtained after thresholding the confidence that the TF binds to a particular gene as performed in the source literature [128]. We will refer to this data as TR.

In relation to GO, the SGD Saccharomyces cerevisiae annotation [195] is used, and for simplicity, we only included the DAG molecular process in our analysis.

Results

Multivariate normal distributions with diagonal covariance matrix are used as component models of the mixture model (see Section 2.3.3). We initialize the EM algorithm with random models, as described in Section 2.3.2. For all experiments, we vary values of λ^+ and λ^-. We use the class labels to compute sensitivity (Eq. 3.13), specificity (Eq. 3.14) and corrected Rand (Eq. 3.12).

As a proof of concept, we use the class labels from YCC to generate pairwise constraints for 5% of all pairs of genes—positive if the genes belong to the same class, negative otherwise—and observe the performance of the method with distinct constraints settings (Figure 6.3 top). In all cases, CR, Spec and Sens tend to one for λ near ten, with the exception of the experiments with positive constraints. In this case, one of the five clusters always remains empty, and two classes are joined in one single cluster. Furthermore, the use of positive constraints only has a stronger effect on the sensitivity, while the negative constraints affect the specificity. This is expected since positive (negative) constraints only penalize false negatives (false positives). It also explains the merged classes in the experiments with positive constraints, since the secondary data gives no penalty for merging two classes.

We observe similar results with GO and TR as secondary data. There is a slight but significant increase of CR and Sens for the methods with positive constraints (t-test indicates an increase at $\lambda^+ = 0.5$ with p-value $= 2.38e - 10$). However, for high λ^+ values (> 0.7), CR and Sens values decrease. No improvements are obtained with the use of positive and

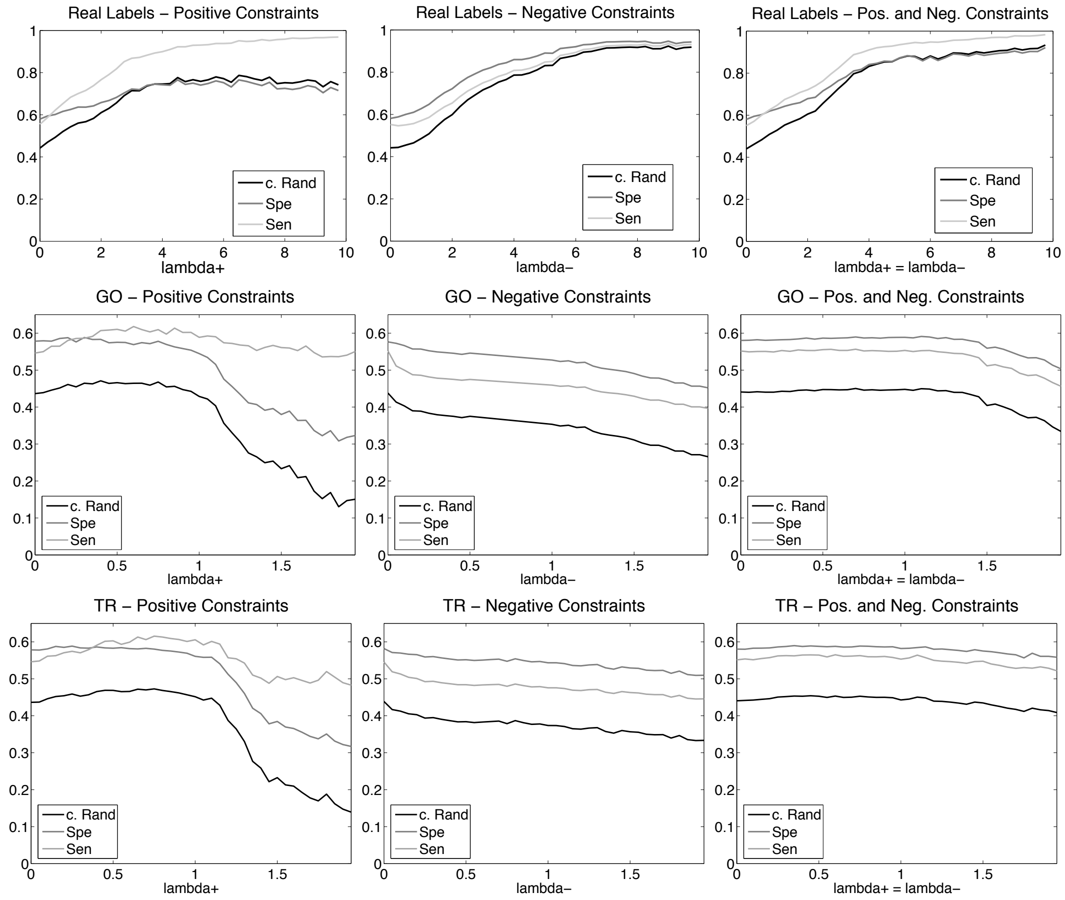


Figure 6.3: *We depict the CR, Sens and Spec after clustering YCC with positive (left), negative (middle) and positive and negative (right) constraints. We used either real class labels (top), GO (middle) or TR (bottom) as secondary information.*

negative constraints, and the negative constraints alone only deteriorate the results.

In order to understand these results, we repeat the experiments with real labels, but this time including also random labels. In total, we generate constraints for 5% of gene pairs. As seen in Figure 6.4, the addition of random labels have a great impact on the recovery of the clusters. The inclusion of 20% of random labels deteriorate the results considerably. For $\lambda = 5$, we have a CR near 0.45 for the data with 20% of noise and a CR near 0.75 for the data with no noise in the constraints. For 60% of random labels, the corrected Rand displays a behavior similar to TR and GO, obtaining low CR values (< 0.2) for high λ (> 5.0). This indicates that (1) the method is not robust with respect to noise in the data, and (2) indicates the presence of noise or non-relevant information in TR and GO.

This is not too surprising, therefore we attempt to estimate the maximal positive effect one can obtain from this secondary data. We perform the computation of enrichment analysis [24] for GO term and TR enrichment, a procedure commonly used in cluster validation, to obtain informative terms from the *true classes*. We repeat the experiments described before with the most informative TF (or GO terms) only. However, we observe only a slight improvement for the negative constraints and a relevant improvement with the use

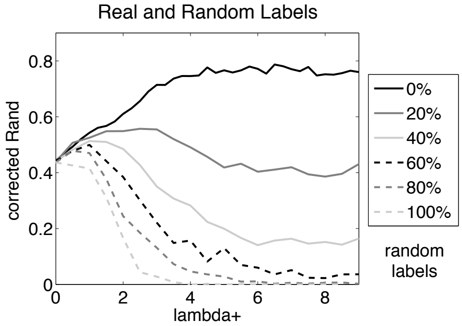


Figure 6.4: *We depict the CR obtained by clustering YCC with positive constraints from 5% of real labels with the inclusion of 0%, 20%, 40%, 60% and 100% random labels.*

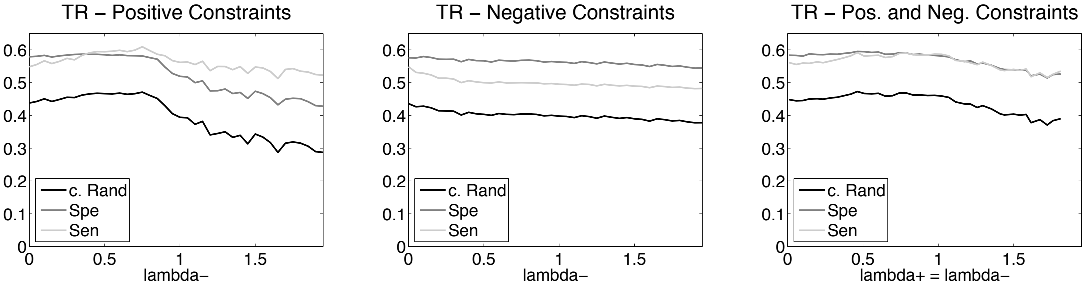


Figure 6.5: *We depict CR, Sens and Spec after clustering YCC with positive (left), negative (middle) and positive and negative (right) constraints after filtering of relevant TR.*

both positive and negative constraints in the TR data set (a CR from 0.454 to 0.472). On the other hand, no improvement is obtained after filtering terms in GO (data not shown).

These results indicate that secondary data has little power for clustering, unless it is of very high quality, free of errors and have no ambiguities. Furthermore, only as few as 20% of error in labels deteriorate the CR by more than 40%. The results for GO and TR indicate that this is the case for both biological data, and unless the procedures for obtaining constraints for GO and TR can be improved, we are more likely to deteriorate results by integrating these data. Note also that we can only obtain the best choice of λ, because the data sets are fully annotated, which is not the case of most biological data sets. Furthermore, high values of λ deteriorate results.

6.3.2 Drosophila Syn-Expression

Data

Time Courses of Drosophila Development. For twelve consecutive one-hour time windows of embryogenesis mRNA levels are measured using the Affymetrix GeneChip Drosophila Genome array. This array targets about 14,000 genes. Results were processed with the standard Affymetrix tool suite [214]. We use the median from three biological replicates. Expression values are transformed to log-ratios by using time point 1 hour as reference. We remove genes not exhibiting at least a two-fold change, which leaves us with 2,684 genes.

In-situ Image Processing. Embryos of Drosophila Melanogaster were collected and aged to produce embryos 0-3, 3-6, 6-9, 9-12, 12-15 and 15-18 hours old [214]. The in-situ reactions were based on a cDNA library of 2,721 clones; in the end images were collected for 1,388 genes. The difference is caused either by a failure of in-situ reactions or by a lack of tissue-specific expression. Images were taken with a dissecting microscope in different focal planes and different orientations.

We use the procedure proposed in [159] for pre-processing the in-situ images. We summarize below the main steps of this image processing pipeline. The majority of in-situ hybridization images in the BDGP database contain the projection of exactly one centered embryo [22]. However, there is a noticeable portion of images with multiple touching embryos. To exploit as many data as possible, the goal of image pre-processing is to locate and extract exactly one complete embryo from each image, even for touching embryos.

To distinguish between embryo and non-embryo pixels we estimate the local variance of gray level intensities for each pixel in a 3×3 neighborhood, following [163]. It suffices to apply a fixed predefined threshold for segmentation using variance estimates because of a homogeneous background in contrast to the embryo. To eliminate erroneous embryo regions, a sequence of morphological closing and opening using a circular mask of radius four is applied [87]. Next, the largest connected component is extracted. The resulting region may be the projection of a single complete or partial embryo or the projection of a set of multiple touching embryos. To distinguish these different cases we apply a series of simple filters based on ellipticity, compactness and area of the extracted region. For regions of multiple touching embryos we introduce a procedure to separate the individuals and to extract a single complete high quality embryo. Further details are given in [159].

The final step of image processing is to register the embryos extracted to a standardized orientation and size to allow for comparison of different expression patterns. The embryo is rotated to align horizontally to the principal axis. Then, the bounding box is scaled to a standard size. Figure 6.6 shows the steps of the image processing pipeline for one example image.

We obtain constraints as described in Section 6.2.2. The 18 developmental stages of the embryo are divided into six developmental periods (0-3, 3-6, 6-9, 9-12, 12-15 and 15-18). Given the results obtained in Section 6.3.1, we would like to have only high quality constraints. Hence, we use conservative thresholds in the procedure for deriving the constraints. More specifically, we select the value s (Section 6.2.2) so that only a small percentage of gene pairs should be constrained (less than 2% of genes with in-situ images). We observe a correlation coefficient exceeding our threshold in at least three or four developmental periods, i.e., we set $p = 3$ or $p = 4$ in Eq 6.7. See Figure 6.2 for an example of how the constraints are obtained. With support of at least three periods, there are 1,756 positive constraints within 170 genes and 2,544 negative constraints within 360 genes. With

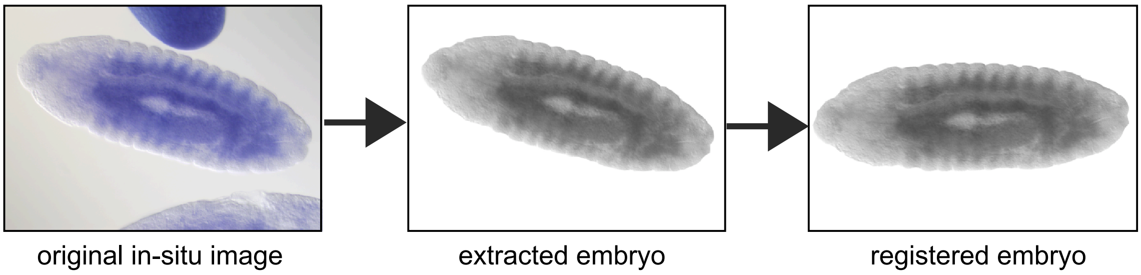


Figure 6.6: *The image pipeline combines registration, morphological operations and further processing steps to automatically process raw images, even if they include multiple touching embryos. Shown here is the image* `in-situ8784` *from gene CG5353. Image reproduced from [159]*

support of at least four stages, there are 270 positive constraints within 66 genes and 640 negative constraints within 151 genes.

ImaGO Term Enrichment. A controlled vocabulary, which follows the Gene Ontology standard [9], is used to annotate spatial gene expression patterns [214]. All images deposited in BDGP are annotated with at least one of these terms. Like with Gene Ontology enrichment analysis described in Appendix A, we can use a statistical test to list ImaGO terms that are over-represented in a cluster. Lower p-values indicate an enrichment in ImaGO terms and, consequently, better results.

This strategy is useful for evaluating the biological quality of a single cluster, but it gives no global assessment for comparing the results obtained by two clustering solutions. A heuristic to perform such an analysis is to compare the p-values obtained for two solutions [73]. A method is said to be better than another method if it has a larger number of ImaGO terms with lower p-values.

Results

We use multivariate Gaussians with diagonal covariance matrices [145] as our components in all mixture estimations. We refer to the results of the unsupervised method as `MoG` and to the clustering with constraints method as `cMoG`. We initialize the EM algorithm with random models, as described in Section 2.3.2. In the unsupervised setting, we estimate the optimal number of clusters with the BIC (Section 2.3.5), which indicates 28 clusters. We use this number for all other runs described below.

Clustering of Gene Expression Data using Mixture of Multivariate Gaussians (`MoG`). The gene expression time-courses cover the period from 1 to 12 hours of the embryo development and expression values are given as log-ratios. Overall, our clustering results reflect two typical classes (see Figure 6.7): the maternal and zygotic genes [68]. Maternal genes appear strongly expressed in the first three hours, usually followed by a decline. Clusters 18 to 28 clearly follow this pattern. These transcripts are deposited in the oocyte; typically the embryo does not transcribe these genes in early development. They are responsible for the determination of body axes and the first phases of the cell cycle and other functions. The period from 2 to 3 hours coincides with the cellularization and the formation of three germ layers following gastrulation, when primary tissues start to develop [130].

On the other hand, genes actively transcribed in the embryo are not expressed in the early time points and expression rises to significant levels only in later stages (3 hours and later). Many of these genes are important to organogenesis. Transcripts in clusters 1 to 4, and 8 to 11 follow the pattern of embryonic activation unambiguously. The functional association can be observed in the over-represented GO terms For other clusters shapes cannot be matched to the maternal or the zygotic expression patterns. Several cluster have maximal expression in the midst of embryonic development. Note that those clusters are less populated than the ones in the maternal and in the zygotic classes.

Using in-situ Images as Secondary Information. We use semi-supervised learning to obtain better solutions for the maximum-likelihood estimation. In order to do so, we restrict the mixture estimation with constraints between pairs of genes. The principle underlying this is shown in Figure 6.1. These constraints will, ideally, differentiate between genes showing co-expression only by chance from those temporal co-expression supported by spatial co-expression (syn-expression).

We use the ImaGO enrichment analysis (Section 6.3.2) to select the best parameterization for cMoG. More precisely, we evaluate the use of constraints shared by either three or four developmental periods, the use of positive constraints and both positive and negative constraints, and four choices of the parameter λ^+ (and λ^-) (0.5, 1.0, 1.5 and 2.0) with $\lambda^+ = \lambda^-$. There is no theory guiding the choices of λ^+ and λ^-, neither is there a definitive “gold standard” or class labels to optimize them. Hence, we made the simple choice to give positive and negative constraints equal weights.

As shown in Table 6.1, all constraint combinations lead to an increase in ImaGO term enrichment, except the use of positive and negative constraints from three stages. Furthermore, values of λ around 1 lead to an improvement, while higher values tend to deteriorate the results. Thus, we choose to use the cMoG results with only positive constraints derived from three developmental periods and a constraint weight of $\lambda_+ = 1.0$.

Changes in the Biological Annotations with cMoG. To investigate the effects of the constraints in the clustering, we compare the results of MoG with cMoG (see Figure 6.8 for cMoG clusters). As explained in the previous section, we choose to use positive constraints, which are supported in at least three developmental stages, as they yield a good recall of in-situ image annotations.

As a sanity check, we inspect the number of constraints satisfied in the final solutions.

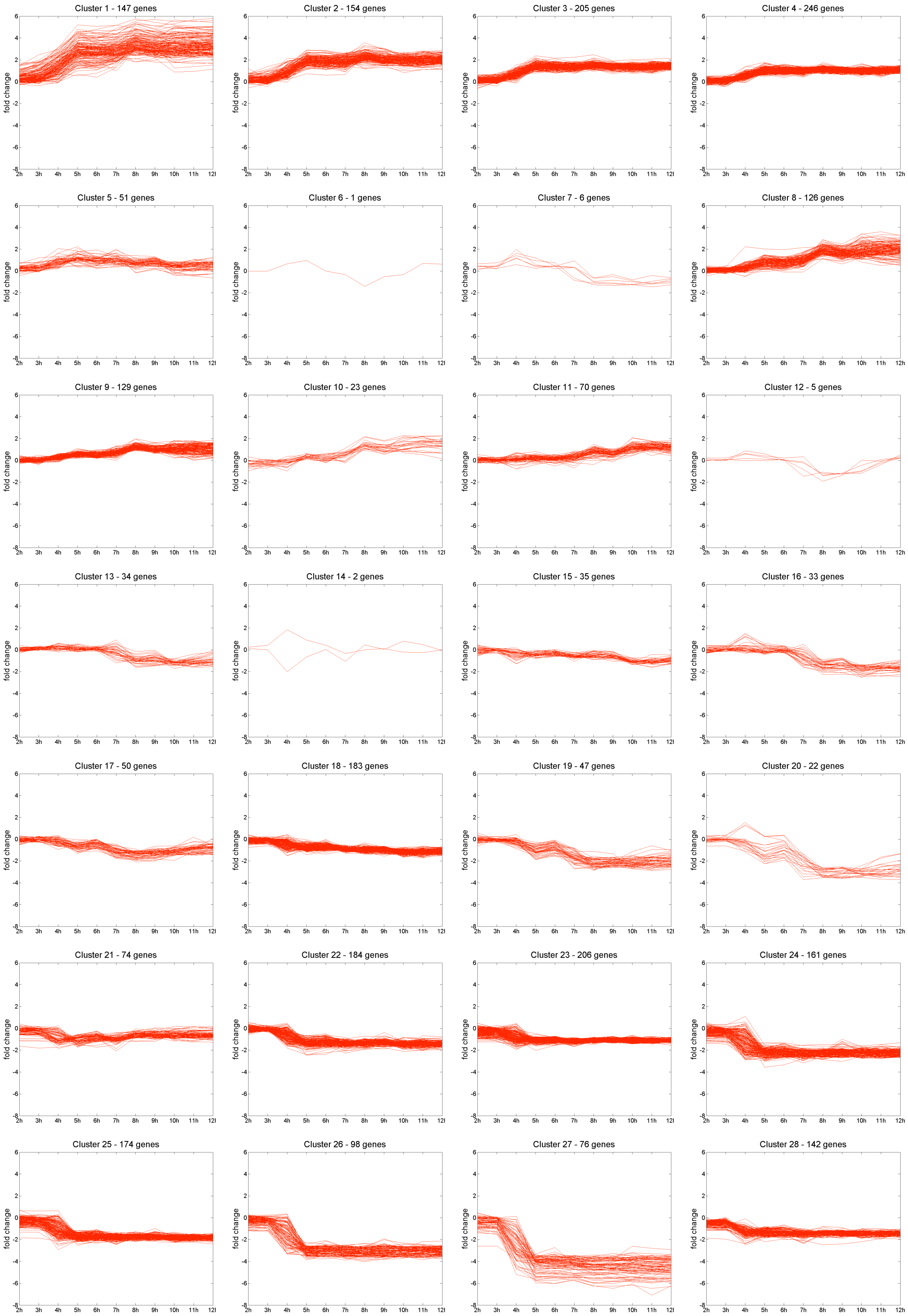


Figure 6.7: *We display the 28 clusters from* MoG.

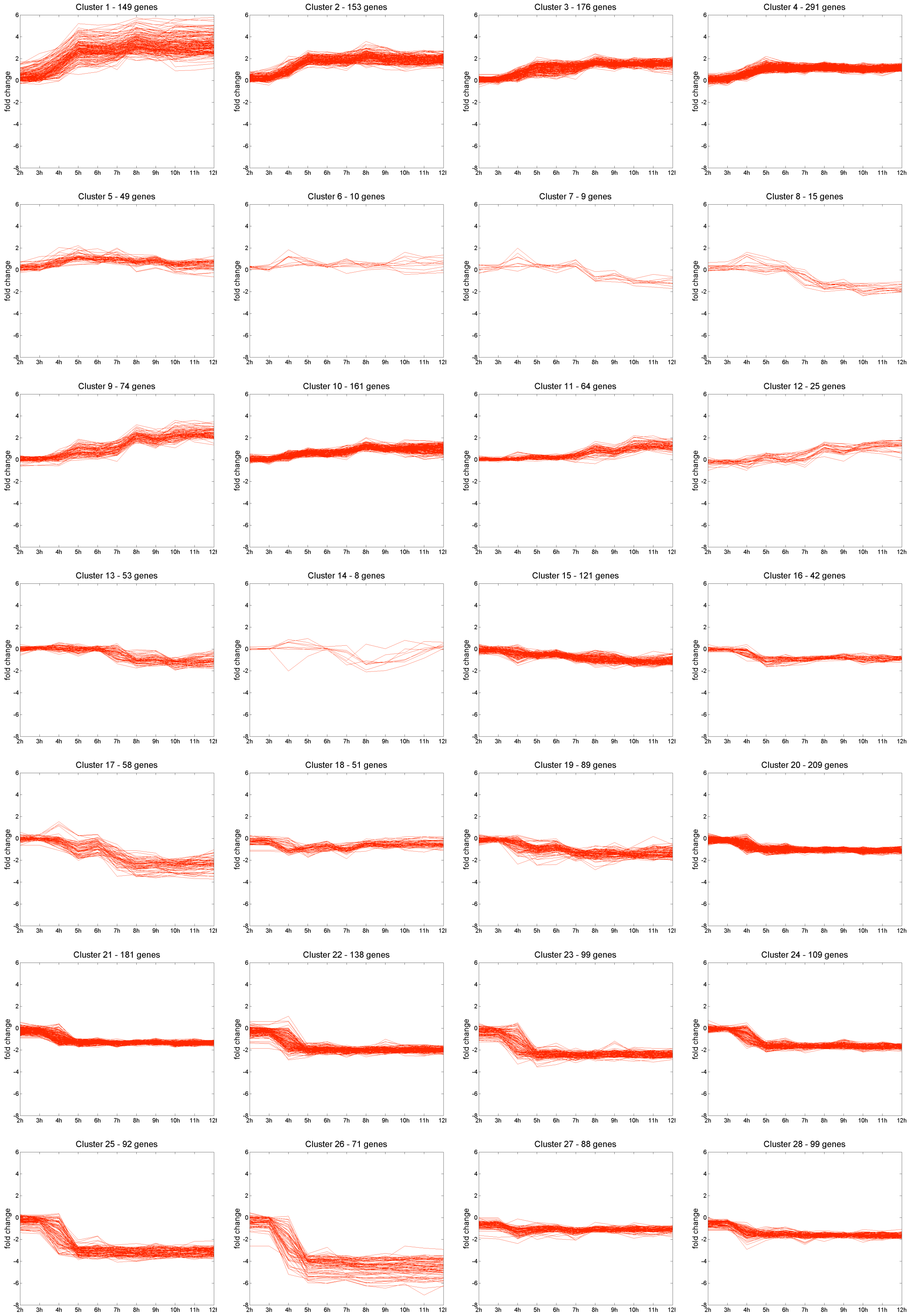


Figure 6.8: *The 28 clusters from* cMoG *show tightly co-regulated pattern and a refinement of the clustering solution of* MoG.

Table 6.1: *We compare the performance of distinct constraints and parameter choices with the ImaGO enrichment analysis. More specifically, we show the proportion of ImaGO terms with lower p-values in* cMoG *compared to* MoG *for constraints derived from a 3 or 4 stages, and distinct weights λ^+ and λ^-. Values exceeding 50% indicate an advantage of* cMoG

		Proportion of terms with lower p-values	
λ^+	λ^-	# stages ≥ 3	≥ 4
0.5	0.0	51%	48%
1.0	0.0	**60**%	**56**%
1.5	0.0	57%	49%
2.0	0.0	43%	46%
0.5	0.5	49%	44%
1.0	1.0	**49**%	52%
1.5	1.5	40%	**59**%
2.0	2.0	43%	47%

With MoG, a sizable proportion of the constraints are already satisfied (656 out of 1,756 pairwise positive constraints), as part of the expression data agrees with the constraints. With cMoG, 1,127 out of 1,756 pairwise positive constraints are satisfied. This value is nearly twice the number found with MoG. This demonstrates that cMoG benefits from the constraints in deriving the clusters of genes exhibiting syn-expression.

Another helpful analysis is the comparison of enrichment of in-situ image annotations (ImaGO), as described in Section 6.3.2. We display in Figure 6.9 a scatter plot of all ImaGO terms, which has an enrichment with a p-value lower than 0.01 in at least one cluster from cMoG or MoG. Based on Figure 6.9, we observe that cMoG has a higher enrichment than MoG in 67 out of 112 relevant ImaGO terms. A binomial test for testing the event of having 67 successes in 112 trials is rejected with a p-value of 0.0232, which indicates that the counts of ImaGO terms with higher enrichment for cMoG is significantly higher than expected by chance. Furthermore, if we take only ImaGO terms with a higher enrichment gain for one of the methods into account (points distant from the diagonal line in Figure 6.9), the advantage of cMoG is even greater (see Figure 6.10 and Figure 6.11). This indicates that even without direct use of the annotation information from ImaGO, cMoG has a greater sensitivity in grouping syn-expressed genes.

Overall, the individual clusters of MoG and cMoG differ only partially. Mainly, cMoG has fewer clusters a smaller amount of genes. One way to quantify the distinctions is to calculate the sensitivity and specificity of cMoG taking the results from MoG as the ground truth. These values are respectively 0.53 and 0.97, which indicate that cMoG has a tendency to subdivide clusters from MoG.

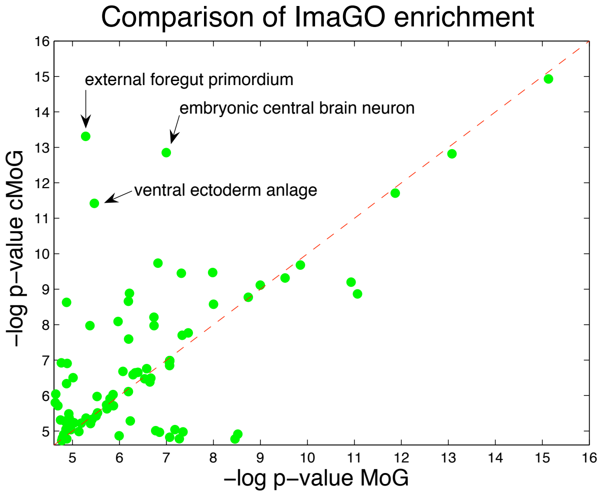


Figure 6.9: *We compare ImaGO term enrichment of* MoG *(x-axis) and* cMoG *(y-axis) in a scatter plot. We use $-log(p)$-values, thus larger values indicate a larger degree of enrichment. Points above the red line indicate a higher enrichment in* cMoG *clusters, and points below in* MoG *clusters. The distance from the diagonal is proportional to the increase in enrichment. For 67 out of 112 ImaGO terms we observe a higher degree of enrichment in* cMoG *clusters.*

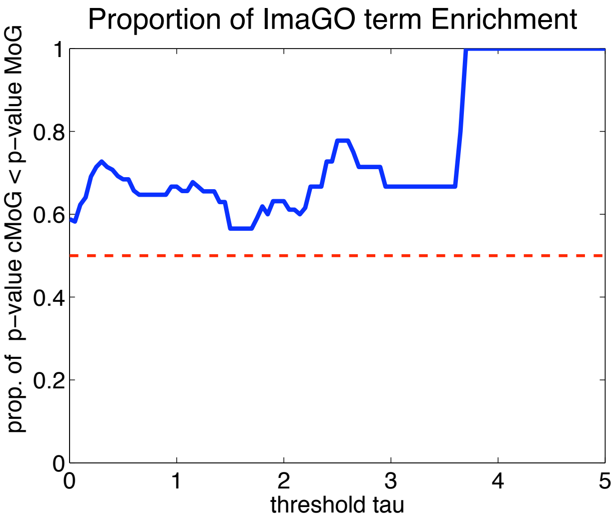


Figure 6.10: *For each threshold τ (x-axis), we depict the proportion of ImaGO terms for which we observe a smaller p-value in* cMoG *than in* MoG *(y-axis). The threshold τ discards ImaGO terms, where the difference in the log of the p-value of cMoG and MoG in smaller than τ. As can be observed, the proportions are higher than 0.5 for all τ values, which indicate an advantage of* cMoG*. Furthermore, the proportions have an increasing tendency for higher τ values.*

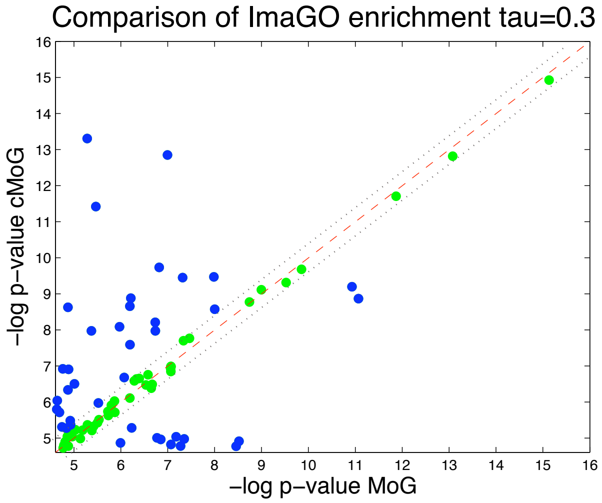
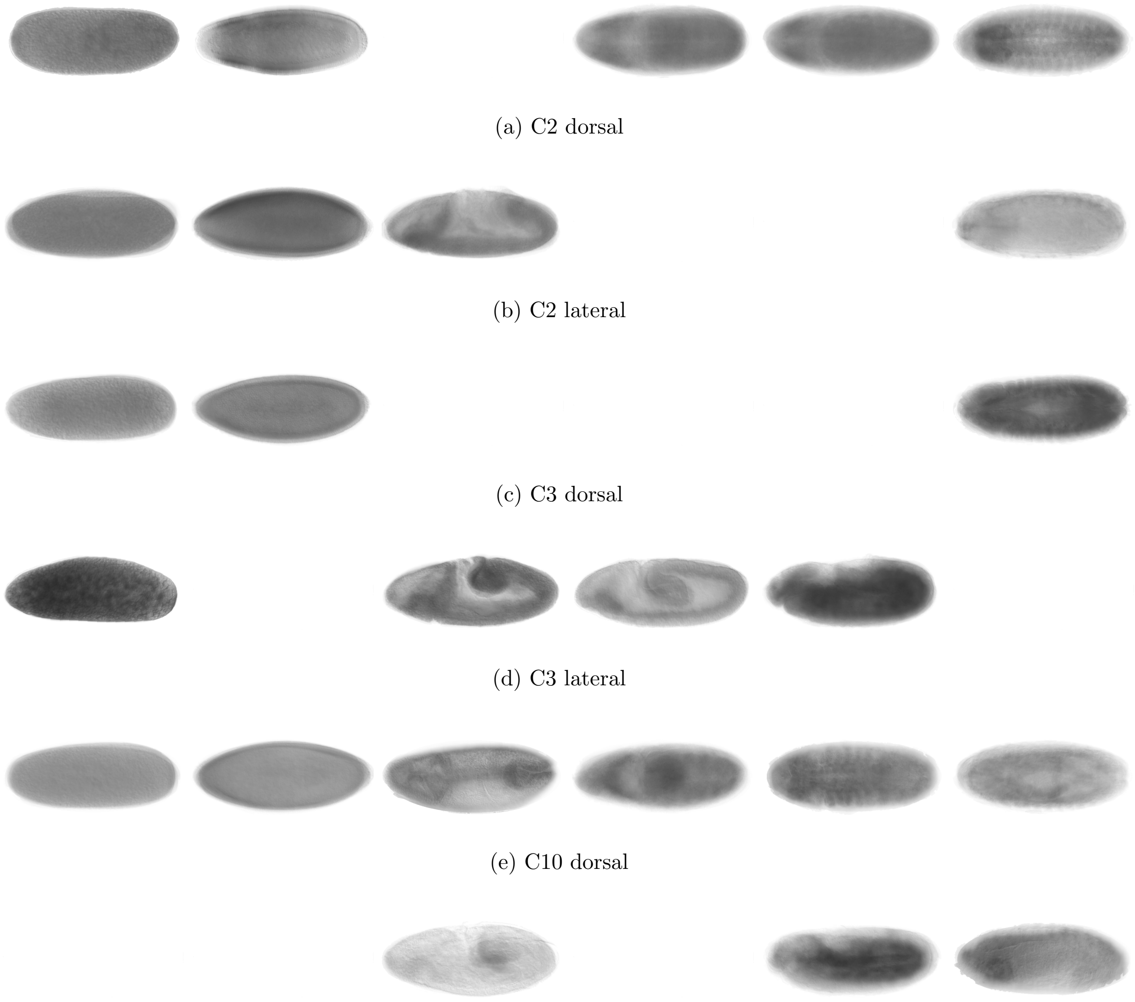


Figure 6.11: *We compare ImaGO term enrichment of* MoG *(x-axis) and* cMoG *(y-axis) in a scatter plot for $\tau = 0.3$. We use $-log(p)$-values, thus larger values indicate a larger degree of enrichment. Points above the red line indicate a higher enrichment in* cMoG *clusters, and values below in* MoG *clusters. Green points between the dotted lines represent ImaGO terms not satisfying the threshold $\tau = 0.3$, where τ indicates the distance from the diagonal line to the dotted lines. We clearly observe a higher proportion of non-filtered ImaGO terms (points in blue) above the diagonal line (32 ImaGO terms) against (12 ImaGO terms) below the diagonal. A binomial test is rejected with a p-value of 0.0018, which indicates an significant advantage of* cMoG.



(a) C2 dorsal

(b) C2 lateral

(c) C3 dorsal

(d) C3 lateral

(e) C10 dorsal

(f) C10 lateral

Figure 6.12: *Averaged in-situ images of clusters C2, C3 and C10 from lateral and dorsal views.*

Functional Annotations in `cMoG`**.** Even for a well characterized genome like Drosophila, the high dimensionality in the annotation data provides only limited information for any single gene. For evaluating the results, we need to identify the corresponding functional modules in the unconstrained and the constrained sets. It is also necessary to show improvements rather than simple correct functional assignments in either solution. In the following, we will refer to the ith cluster from `cMoG` and `MoG` as Ci and Ui respectively.

For some cases, the mapping from clusters of `cMoG` to `MoG` is simply one to one (e.g., C1 to U1, C5 to U5, C11 to U11 and C12 to U10). However, the majority of clusters show larger differences. For simplicity, we focus the functional analysis on clusters with zygoticly expressed genes (i.e., C1 to C4 and C9 to C12 in Figure 6.8).

Cluster `C2` represents a good example of the changes resulting from the introduction of constraints. It contains most of the genes from U2 (135 genes) and 16 genes from U3. Out of the seven genes, which show similar expression patterns and have co-location constraints (*CG6930, E2f, Iswi, neur, Set, RhoGAP771e, trx*), only four (*G6930, E2f, Iswi, trx*) are found in U2. All these genes have ImaGO annotations related to *ventral nerve cord primordium* and related terms (see Figure 6.12 (a) and (b) for mean in-situ images of these genes). Related genes that have no constraints but are annotated as part of the *embryonic central nervous system* are included in C2 (*CG7372, CG14722, fzy*). The analysis of GO term enrichment returns terms such as *nervous system development* (p-value of 3.38e-23) and *system development* (p-value of 9.54e-21) (similar term enrichment is found for cluster U2). It should be noted that clusters U2 and U3 have a similar mean expression pattern. They mainly differ in the time when genes reach the plateau of maximal expression.

An example for larger changes is cluster C3, which is mainly composed of genes originally found in U3 (101 genes) and U8 (63 genes). C3 has constraints between three genes (*rhea*, *Rsf1* and *vig*) of which *rhea* and *vig* come from cluster U8 and *Rsf1* from U3 (see Figures 6.12 (c) and (d) for mean in-situ images of C3). This cluster presents higher enrichment for ImaGO terms related to *muscle primordium* (genes *CG5522, CG9253, Dg, Mef2, betaTub60D, htl, mbc, vig*) than U3 and U8. Furthermore, GO term analysis reveals that this cluster shows enrichment for *nervous system development* (p-value of 1.33e-11) and *axis specification* (p-value of 9.31e-05). For the latter term, seven genes are originally from U3 (*Dfd, Lis-1, sti, Syx1A, sqd, Ras85Dm, tup*) and five from U8 (*baz, Dg, pnt, Rac2, tok*), demonstrating that the changes introduced increase the number of syn-expressed genes within C3.

The cluster C9 represents only a subset of U8 (59 out of the 126 genes) but has no genes with constraints. It consists of genes from U8 that are not constrained to genes from C3 (see previous paragraph). Still, it is enriched in the ImaGO term *embryonic central nervous system* and related terms (genes *HLHmbeta, NetB, Oli, lin-28, scrt, sd, tap, uzip* and *zfh2*). The cluster is also enriched in the terms *organ* (p-value 2.66e-05) and *ectoderm development* (p-values 8.54e-05), which are significantly enriched in U8. In other words, this cluster is a specialization of U8, whose genes are specific to *organ development*.

C10 is formed by the addition of most genes in the U4 cluster (39 genes) to U10 (118 genes). There are seven genes constraining this cluster (*CG6751, CG18446, CG13912, CG10924, CG8745, dm, Klp61F*) (see Figures 6.12 (e) and (f)). ImaGO term enrichment relates this cluster to *yolk nuclei* and *amnioserosa*. It is also enriched in the GO term *nervous system development* (p-value 1.06e-08), all of which are insignificantly enriched in the U10 cluster.

It is also worthwhile to look at those few cases where MoG performs better. From Figure 6.9, two ImaGO terms with higher enrichment increase in MoG are *maternal* and *procephalic ectoderm anlage in statu nascendi*. The former term is enriched in cluster C22 and U21, where MoG has some more genes related to the term *maternal* (34 genes in MoG compared to 31 genes in cMoG). For the latter term, clusters U2 and C2 are both enriched, and there was only one annotated gene in U2 not in C2. As none of these annotated groups of genes has pairwise constraints, we cannot detect any direct effect of the clustering with constraints on these results.

In summary, the refined clusters improve the generation of testable hypotheses for the role of uncharacterized genes. Overall, we observe improvement in annotation of genes related to development of the Drosophila, in particular with respect to the ImaGO annotations, which increases our confidence in the delineation of syn-expressed functional modules.

Chapter 7

Discussion

Clustering is a crucial first step in the analysis of large-scale gene expression experiments. Peculiarities of gene expression data from microarray experiments, require the development of novel clustering methods. While mixture models provide a statistical framework to perform clustering, the specification of proper component density functions, which take characteristics inherent to the multi-variate data at hand, remains an open problem. In this thesis, we propose two novel component models for analyzing gene expression measured over time or developmental processes. Furthermore, we approach the problem of integrating additional sources of biological data to enhance the analysis of gene expression. This is done by using a semi-supervised method for estimating the mixture model.

In the next sections, we present the final remarks and future work of each specific contribution of this thesis.

Mixture Models and Cluster Validation

We introduce, in Chapter 3, a novel validation index for comparing overlapping partitions obtained by mixture model based clustering algorithms. This index is an extension of the well known corrected Rand (CR). In the context of mixture models, our experimental work shows that the extended corrected Rand index yields significant improvements when compared to the results obtained by the traditional corrected Rand. Finally, it is important to point out that there are still many theoretical and practical aspects of cluster validation in the context of mixture models. The definition of the extended Corrected Rand represents an initial contribution to these problems.

Analysis of Gene Expression Time Courses

We present in Chapter 4 an application of mixture models and linear HMMs for the analysis of gene expression time course data. We take advantage of several characteristics of this robust statistical model, which is of great value in the analysis of gene expression time course data. With a benchmark data set, we show that mixture of HMMs have better class recovery than model-based clustering methods with splines or autoregressive models as components. We also evaluate different methods for model initialization. In this context, a Bayesian approach exploring the linear topology of the HMMs obtained the best practical results. Moreover, we show that the Viterbi decomposition is able to enhance the mixture of HMMs for the yeast cell cycle data set. In an anedoctal analysis with HeLa cell data, we also show that the Viterbi decomposition refines clusters in a biological meaningful way. The use of the entropy threshold for discarding ambiguous cluster assignments improve the specificity of Gene Ontology annotation, which reassures the usefulness of the soft assignments of the mixtures in detecting unambiguous clusters. Our flexible framework, combined with an effective graphical user interface implemented in the GQL application, supports interactive and exploratory knowledge discovery of gene expression time course data.

There are several aspects still to be explored in the use of linear HMMs and mixtures of linear HMMs. First, the use of other pdfs as emission functions such as the Gamma pdf, as explored in [235], could produce better results for gene expression data. One issue that have been recently explored is the fact that most data sets have few time points [73]. This can be addressed by biasing the topology learning method towards models with fewer number of states. Furthermore, an extension of our framework to perform simultaneous topology learning and mixture estimation should also improve the performance of the mixture of linear HMMs. Recently, a great deal of data sets with multiple time courses of a given species have been made available. These data sets present time course measurements over distinct gene knockouts [236], environmental conditions [82, 174] or patients [225]. In fact, such data sets pose new methodological questions about conditions at which genes are differentially expressed, or what are the temporal dynamics of these differences? For such tasks, we could extend the linear HMM to multiple linear HMM models. Then, we could apply structural learning methods to explore issues concerning detection of time-lag relationships, temporal dynamics of these differences, and finding groups displaying similar differential expression profiles.

Analysis of Gene Expression in Lymphoid Development

The regulatory processes underlying cell proliferation and differentiation are of central interest to developmental biologists and clinicians. They are frequently the focus of large-scale studies in which gene expression along paths of differentiation are investigated. To make use of these data in a principled manner, as the main contribution of this thesis, we presented in Chapter 5 a novel statistical framework, called `DTrees`, that models gene expression in the course of development. By combining `DTrees` in a mixture model (`MixDTrees`), we facilitate interactive querying and visualization of data and, more importantly, the detection of clusters of co-expressed genes, which provide a basis for the identification of key players in the regulatory mechanism and their mode of action.

In particular, with `MixDTrees` with structure set to the developmental tree as provided by biologists (`MixDTree-Dev`), we detect groups of genes not found by classical clustering methods such as Self-organizing maps (SOM). By incorporating microRNA binding data, we show how to identify complex regulatory relationships. In comparison to an analysis based only on sequence data, we predict a manageable number of plausible microRNA targets [91]. Moreover, by the inspection of the developmental profiles of gene targets associated with microRNAs, our method offers some insights into the biological role of the predicted microRNAs.

We show that the `DTree` inferred from the complete Lymphoid data set approximates the dependencies intrinsic to Lymphoid development well. Furthermore, by combining the methods for mixture estimation and for the inference of the `DTree` structure, we find `DTrees` structures specific to groups of co-regulated genes. These groups display different differentiation pathways reflected by the distinct estimated dependence structures. Furthermore, groups have a lineage specific expression pattern. Enrichment analyses of gene annotation using KEGG and GO indicate development-specific function of the groups found.

For simulated data, `MixDTrees` compares favorably to other methods widely used for finding groups of co-expressed genes, even for data arising from variable dependence structures. In particular, our method is not susceptible to over-fitting, which is otherwise a frequent problem in the estimation of mixture models from sparse data.

Interesting extensions to our analysis are possible, even when one only considers gene expression data and the basic method. None of the currently publicly available data sets offer both a tree with a large number of branches and a detailed view of all development stages. An interesting compendia of gene expression data from lymphoid cells [105], concentrates on mature and immature cells in final development stages. The creation of an expression compendium such as the one in [105], where many intermediary stages of differentiation of the developmental tree are present, will be of great value as computational methods can exploit characteristics intrinsic to cell development.

It is also important to point out that developmental biologists are still redrawing developmental trees with the discovery of new intermediary stages and “alternative” paths of development [25, 140, 177]; a particular developmental stage might also be formed by a mixture of distinct cell types not yet well-characterized. An example of an alternative path is the fact that DN1 T cells can be originated not only from the lymphoid progenitor as depicted in Figure 5.1, but also from the earlier multi-potent progenitor cells [25]. It is an interesting prospect to extend the structure estimation approach to infer confidence values for branches and stages of a developmental tree from gene expression; as well as to estimate graphs of arbitrary structures. The estimation of graphs with arbitrary structures has already been explored. For example, see [40, 213] for approaches based on graphical models and [182] for an approach based on estimation of covariance matrices. However, in contrast to the method used for the estimation of `DTree` structures, those methods do not provide an efficient and exact solutions for inferring dependence graphs.

Clustering with Constraints for Integration of Heterogeneous Biological Data

If high-quality secondary data is available, semi-supervised learning is an effective framework for the analysis of heterogeneous data as previous experiments using class labels demonstrate [185, 187]. In our experiments based on yeast cell cycle data set (Chapter 6), we use biological information routinely used to support cluster validity as secondary data, i.e., GO annotation and location analysis. Surprisingly, this data can deteriorate cluster quality drastically, if parameters are not chosen properly. Furthermore, we can show that the addition of noise can drastically reduce the performance of clustering with constraints. Although there are other parameter choices to explore, further theoretical questions to address, and more data sets to perform experiments, a main point of our analysis remains valid and clear: secondary data can have little power for clustering, unless it is of good quality, free of errors and have no ambiguities.

These issues discussed in the previous paragraph indicate the need for methodological improvements in clustering with constraints methods. One possible solution for the selection of parameters is the use of cross-validation procedures, as suggested in [45]. Moreover, the inclusion of a step to evaluate the constraint "quality" during clustering execution can be an interesting strategy for preventing problems related to noise in the constraints.

For the Drosophila development case study, we show how to automatically fuse temporal and spatial gene expression patterns by clustering with few high quality constraints derived from in-situ data. Our results demonstrate that the clusters found are biologically meaningful and that we can improve the detection of syn-expressed genes. In particular, the cluster results, obtained after applying the constraints, are better at recovering the functional annotation of ImaGO terms than the clustering solution without constraints. Inferred groups are worthwhile targets for further investigation, either with classical biological analysis or as input for methods to infer gene networks.

There are several open questions regarding the detection of syn-expressed genes. One direction is to improve the image processing pipeline by, for example, using higher quality images, such as 3D models from [96, 117] or images with sub-cellular localization [127]. In relation to the constraints it would be desirable to model the temporal nature of the constraints derived from the in-situ images. A quite challenging problem is to combine an automatic image annotation of expressed cellular compartments with a tree describing the Drosophila development. This would allow us to obtain a detailed developmental profile of genes for this complex multicellular organism, i.e., at which tissues a particular gene is expressed. Hence, we could use Mixture of Dependence Trees, as proposed in Chapter 5, to analyze gene expression of Drosophila development.

Bibliography

[1] J. Aach and G. M. Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 17(6):495–508, 2001.

[2] S. Aizawa, H. Nakano, T. Ishida, R. Horie, M. Nagai, K. Ito, H. Yagita, K. Okumura, J. Inoue, and T. Watanabe. Tumor necrosis factor receptor-associated factor (TRAF) 5 and TRAF2 are involved in CD30-mediated NFkappaB activation. *Journal of Biological Chemistry*, 272(4):2042–2045, 1997.

[3] K. Akashi, X. He, J. Chen, H. Iwasaki, C. Niu, B. Steenhard, J. Zhang, J. Haug, and L. Li. Transcriptional accessibility for genes of multiple tissues and hematopoietic lineages is hierarchically controlled during early hematopoiesis. *Blood*, 101(2):383–389, 2003.

[4] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell, Fourth Edition*. Garland, 2002.

[5] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000.

[6] O. Alter, P. O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences of the United States of America*, 97(18):10101–10106, 2000.

[7] T. W. Anderson. Asymptotically efficient estimation of covariance matrices with linear structure. *The Annals of Statistics*, 1(1):135–141, 1973.

[8] I. P. Androulakis, E. Yang, and R. R. Almon. Analysis of time-series gene expression data: Methods, challenges, and opportunities. *Annual Review Of Biomedical Engineering*, 9:205–228, 2007.

[9] M. Ashburner. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[10] J. D. Banfield and A. E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49(3):803–821, 1993.

[11] Z. Bar-Joseph. Analyzing time series gene expression data. *Bioinformatics*, 20(16):2493–503, 2004.

[12] Z. Bar-Joseph, G. Gerber, D. K. Gifford, T. S. Jaakkola, and I. Simon. A new approach to analyzing gene expression time series data. In *RECOMB '02: Proceedings of the sixth annual international conference on Computational biology*, pages 39–48, New York, NY, USA, 2002. ACM.

[13] Z. Bar-Joseph, G. Gerber, L. Simon, D. K. Gifford, T. S. Jaakkola, and T. S. Jaakkola. Comparing the continuous representation of time-series expression profiles to identify differentially expressed genes. *Proceedings of the National Academy of Sciences of the United States of America*, 100(18):10146–10151, 2003.

[14] Z. Bar-Joseph, G. K. Gerber, D. K. Gifford, T. S. Jaakkola, and I. Simon. Continuous representations of time-series gene expression data. *Journal of Computational Biology*, 10(3-4):341–356, 2003.

[15] Z. Bar-Joseph, G. K. Gerber, T. I. Lee, N. J. Rinaldi, J. Y. Yoo, F. Robert, D. B. Gordon, E. Fraenkel, T. S. Jaakkola, R. A. Young, and D. K. Gifford. Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21(11):1337–1342, 2003.

[16] Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17 Suppl 1:i22–i29, 2001.

[17] D. P. Bartel. MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116(2):281–297, 2004.

[18] K. Basso, A. A. Margolin, G. Stolovitzky, U. Klein, R. DallaFavera, and A. Califano. Reverse engineering of regulatory networks in human B cells. *Nature Genetics*, 37(4):382–390, 2005.

[19] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 59–68, New York, NY, USA, 2004. ACM.

[20] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite Markov chains. *Annals of Mathematical Statistics*, 37:1554–1563, 1966.

[21] L. H. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique ocurring in the statistical analysis of probabilistic functions of markov chains. *Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[22] BDGP. Berkeley Drosophila Genome Project, http://www.fruitfly.org.

[23] N. Beerenwinkel, J. Rahnenfuhrer, M. Daumer, D. Hoffmann, R. Kaiser, J. Selbig, and T. Lengauer. Learning multiple evolutionary pathways from cross-sectional data. In *RECOMB '04: Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*, pages 36–44, New York, NY, USA, 2004. ACM Press.

[24] T. Beissbarth and T. P. Speed. GOstat: find statistically overrepresented Gene Ontologies within a group of genes. *Bioinformatics*, 20(9):1464–1465, 2004.

[25] A. Bhandoola and A. Sambandam. From stem cell to t cell: one route or many? *Nature Reviews Immunology*, 6:117–126, 2006.

[26] C. Biernacki and G. Govaert. Using the classification likelihood to choose the number of clusters. *Computing Science and Statistics*, 29(2):451–457, 1997.

[27] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, International Computer Science Institute, 1997.

[28] N. Bolshakova and F. Azuaje. Cluster validation techniques for genome expression data. *Signal Processing*, 83(4):825–833, 2003.

[29] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

[30] L. Brehelin. Clustering gene expression series with prior knowledge. In *Algorithms in Bioinformatics, Proceding of the Workshop in Bioinformatics*, number 3691 in LNBI, pages 27–38. Springer Verlag, 2005.

[31] J. P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences of the United States of America*, 101(12):4164–4169, 2004.

[32] L. Busino, M. Donzelli, M. Chiesa, D. Guardavaccaro, D. Ganoth, N. V. Dorrello, A. Hershko, M. Pagano, and G. F. Draetta. Degradation of Cdc25A by beta-TrCP during S phase and in response to DNA damage. *Nature*, 426(6962):87–91, 2003.

[33] A. J. Butte and I. S. Kohane. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In *PSB 2005: Proceedings of the Pacific Symposium on Biocomputing*, pages 418–429, 2000.

[34] L. Bystrykh, E. Weersing, B. Dontje, S. Sutton, M. T. Pletcher, T. Wiltshire, A. I. Su, E. Vellenga, J. Wang, K. F. Manly, L. Lu, E. J. Chesler, R. Alberts, R. C. Jansen, R. W. Williams, M. P. Cooke, and G. de Haan. Uncovering regulatory pathways that affect hematopoietic stem cell function using 'genetical genomics'. *Nature Genetics*, 37(3):225–232, 2005.

[35] G. A. Calin, C. Liu, C. Sevignani, M. Ferracin, N. Felli, C. D. Dumitru, M. Shimizu, A. Cimmino, S. Zupo, M. Dono, M. L. Dell'Aquila, H. Alder, L. Rassenti, T. J. Kipps, F. Bullrich, M. Negrini, and C. M. Croce. MicroRNA profiling reveals distinct signatures in B cell chronic lymphocytic leukemias. *Proceedings of the National Academy of Sciences of the United States of America*, 101(32):11755–11760, 2004.

[36] B. P. Carlin and T. A. Louis. *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall/CRC, 2000.

[37] G. Celeux and G. Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28:781–793, 1995.

[38] G. Celeux and G. Soromenho. An entropy criterion for assessing the number of clusters in a mixture model. *Journal of Classification*, 13(2):195–212, 1996.

[39] O. Chapelle, B. Schoelkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.

[40] S. Chaudhuri, M. Drton, and T. S. Richardson. Estimation of a covariance matrix with zeros. *Biometrika*, 94(1):199–216, 2007.

[41] C. Z. Chen, L. Li, H. F. Lodish, and D. P. Bartel. MicroRNAs modulate hematopoietic lineage differentiation. *Science*, 303(5654):83–86, 2004.

[42] 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 the mitotic cell cycle. *Molecular Cell*, 2(1):65–73, 1998.

[43] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[44] A. Cimmino, G. A. Calin, M. Fabbri, M. V. Iorio, M. Ferracin, M. Shimizu, S. E. Wojcik, R. I. Aqeilan, S. Zupo, M. Dono, L. Rassenti, H. Alder, S. Volinia, C. Liu, T. J. Kipps, M. Negrini, and C. M. Croce. miR-15 and miR-16 induce apoptosis by targeting BCL2. *Proceedings of the National Academy of Sciences of the United States of America*, 102(39):13944–13949, 2005.

[45] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang. Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1553–1567, 2004.

[46] T. H. Cormen, C. E. Leiserson, and C. Rivest, Ronald L. amd Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, second edition, 2001.

[47] I. G. Costa, M. C. P. de Souto, and A. Schliep. Validating gene clusterings by selecting informative gene ontology terms with mutual information. In *Advances in Bioinformatics and Computational Biology, Proceedings of the Brazilian Symposium on Bioinformatics*, LNBI, pages 81–92. Springer Verlag, 2007.

[48] I. G. Costa, R. Krause, L. Optiz, and A. Schliep. Semi-supervised learning for the identification of syn-expressed genes from fused microarray and in situ image data. *BMC Bioinformatics*, 8(Suppl 10):S3, 2007.

[49] I. G. Costa, S. Roepcke, C. Hafemeister, and A. Schliep. Inferring differentiation pathways from gene expression. *Bioinformatics*, 2008. Accepted.

[50] I. G. Costa, S. Roepcke, and A. Schliep. Gene expression trees in lymphoid development. *BMC Immunology*, 8(1):25, 2007.

[51] I. G. Costa and A. Schliep. On external indices for mixtures: validating mixtures of genes. In M. Spiliopoulou, R. Kruse, C. Borgelt, A. Nurnberger, and W. Gaul, editors, *From Data and Information Analysis to Knowledge Engineering, Proceedings of the 29th Annual Conference of the Gesellschaft fur Klassifikation*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 662–669. Springer, 2005.

[52] I. G. Costa and A. Schliep. On the feasibility of heterogeneous analysis of large scale biological data. In *Proceedings of ECML/PKDD 2006 Workshop on Data and Text Mining for Integrative Biology*, pages 55–60, 2006.

[53] I. G. Costa, A. Schönhuth, and A. Schliep. The Graphical Query Language: a tool for analysis of gene expression time-courses. *Bioinformatics*, 21(10):2544–2545, 2005.

[54] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., New York, N.Y., 1991.

[55] C. M. Croce and G. A. Calin. miRNAs, cancer, and stem cell division. *Cell*, 122(1):6–7, 2005.

[56] S. Datta and S. Datta. Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, 19:459–466, 2003.

[57] I. Davidson and S. S. Ravi. Intractability and clustering with constraints. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 201–208, New York, NY, USA, 2007. ACM.

[58] U. de Lichtenberg, L. J. Jensen, A. Fausbull, T. S. Jensen, P. Bork, and S. Brunak. Comparison of computational methods for the identification of cell cycle-regulated genes. *Bioinformatics*, 21(7):1164–1171, 2005.

[59] M. C. P. de Souto, D. A. S. Araujo, I. G. Costa, R. G. F. Soares, T. B. Ludermir, and A. Schliep. Comparative study on normalization procedures for cluster analysis of gene expression datasets. In *Proceedings of the International Joint Conference on Neural Networks*. IEEE Computer Society, 2008. Accepted.

[60] M. C. P. de Souto, R. B. C. Prudencio, R. G. F. Soares, D. A. S. Araujo, I. G. Costa, T. B. Ludermir, and A. Schliep. Ranking and selecting clustering algorithms using a meta-learning approach. In *Proceedings of the International Joint Conference on Neural Networks*. IEEE Computer Society, 2008. Accepted.

[61] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

[62] R. Desper, F. Jiang, O. P. Kallioniemi, H. Moch, C. H. Papadimitriou, and A. A. Schaffer. Inferring tree models for oncogenesis from comparative genome hybridization data. *Journal of Computational Biology*, 6(1):37–51, 1999.

[63] R. Desper, F. Jiang, O. P. Kallioniemi, H. Moch, C. H. Papadimitriou, and A. A. Schaffer. Distance-based reconstruction of tree models for oncogenesis. *Journal of Computational Biology*, 7(6):789–803, 2000.

[64] P. D'haeseleer. How does gene expression clustering work? *Nature Biotechnology*, 23(12):1499–1501, 2005.

[65] J. Diebolt and C. P. Robert. Estimation of finite mixture distributions through bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(2):363–375, 1994.

[66] S. Dudoit and J. Fridlyand. A prediction-based resampling method to estimate the number of clusters in a dataset. *Genome Biology*, 3(7):R36, 2002.

[67] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: probabilistic models of proteins and nucleic acids*. Cambridge Press, 1998.

[68] B. Edgar. Diversification of cell cycle controls in developing embryos. *Current Opinion in Cell Biology*, 7(6):815–824, 1995.

[69] R. Edgar, M. Domrachev, and A. E. Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002.

[70] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap.* Chapman and Hall, 1997.

[71] M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95:14863–8, 1998.

[72] A. J. Enright, B. John, U. Gaul, T. Tuschl, C. Sander, and D. S. Marks. MicroRNA targets in Drosophila. *Genome biology*, 5(1):R1, 2003.

[73] J. Ernst, G. J. Nau, and Z. Bar-Joseph. Clustering short time series gene expression data. *Bioinformatics*, 21 Suppl 1:i159–i168, 2005.

[74] J. Ernst, O. Vainas, C. T. Harbison, I. Simon, and Z. Bar-Joseph. Reconstructing dynamic regulatory maps. *Molecular systems biology*, 3:74, 2007.

[75] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[76] N. Felli, L. Fontana, E. Pelosi, R. Botta, D. Bonci, F. Facchiano, F. Liuzzi, V. Lulli, O. Morsilli, S. Santoro, M. Valtieri, G. A. Calin, C. G. Liu, A. Sorrentino, C. M. Croce, and C. Peschle. MicroRNAs 221 and 222 inhibit normal erythropoiesis and erythroleukemic cell growth via kit receptor down-modulation. *Proceedings of the National Academy of Sciences of the United States of America*, 102(50):18081–18086, 2005.

[77] M. Figueiredo and A. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

[78] V. Filkov, S. Skiena, and J. Zhi. Analysis techniques for microarray time-series data. *Journal of Computational Biology*, 9(2):317–30, 2002.

[79] C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

[80] C. Fraley and A. E. Raftery. Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification*, 24(2):155–181, 2007.

[81] M. Futschik and H. Herzel. Are we overestimating the number of cell-cycling genes? the impact of background models. In *GCB 2007: Proceedings of the German Conference on Bioinformatics*, volume 115 of *Lecture Notes in Informatics*, pages 2–14, 2007.

[82] A. Gasch, P. Spellman, C. Kao, O. Carmel-Harel, M. Eisen, G. Storz, D. Botstein, and P. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11:4241–57, 2000.

[83] A. C. Gavin, M. Busche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, A. Michon, C. M. Cruciat, M. Remor, C. Hufert, M. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M. Heurtier, R. R. Copley, A. Edelmann, E. Querfurth, V. Rybin, G. Drewes, M. Raida, T. Bouwmeester, P. Bork, B. Seraphin, B. Kuster, G. Neubauer, and G. S. Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–147, 2002.

[84] A. Gelman and D. Rubin. Markov chain monte carlo methods in biostatistics. *Statistical Methods in Medical Research*, 5:339–355, 1996.

[85] J. Gilthorpe, M. Vandromme, T. Brend, A. Gutman, D. Summerbell, N. Totty, and P. W. J. Rigby. Spatially specific expression of Hoxb4 is dependent on the ubiquitous transcription factor NFY. *Development*, 129(16):3887–3899, 2002.

[86] R. Glynne, G. Ghandour, J. Rayner, D. H. Mack, and C. C. Goodnow. B-lymphocyte quiescence, tolerance and activation as viewed by global gene expression profiling on microarrays. *Immunological reviews*, 176:216–246, 2000.

[87] R. Gonzalez and P. Wintz. *Digital image processing*. Addison-Wesley, 1991.

[88] A. D. Gordon. *Classification: methods for the exploratory analysis of multivariate data*. Chapman & Hall, 1981.

[89] A. D. Gordon. *Classification*. Chapman & Hall, New York, 1999.

[90] GQL. Graphical query language, http://www.ghmm.org/gql.

[91] S. GriffithsJones, R. J. Grocock, S. van Dongen, A. Bateman, and A. J. Enright. miRBase: microRNA sequences, targets and gene nomenclature. *Nucleic Acids Research*, 34(Database issue):D140–D144, 2006.

[92] J. M. Hammersley and P. Clifford. Markov field on finite graphs and lattices. Unpublished, 1971.

[93] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[94] S. D. Hatfield, H. R. Shcherbata, K. A. Fischer, K. Nakahara, R. W. Carthew, and H. Ruoho-laBaker. Stem cell division is regulated by the microRNA pathway. *Nature*, 435(7044):974–978, 2005.

[95] R. J. Hathaway. A constrained formulation of maximum-likelihood estimation for normal mixture distributions. *The Annals of Statistics*, 13(2):795–800, 1985.

[96] C. L. L. Hendriks, S. V. E. Keränen, C. C. Fowlkes, L. Simirenko, G. H. Weber, A. H. DePace, C. Henriquez, D. W. Kaszuba, B. Hamann, M. B. Eisen, J. Malik, D. Sudar, M. D. Biggin, and D. W. Knowles. Three-dimensional morphology and gene expression in the drosophila blastoderm at cellular resolution i: data acquisition pipeline. *Genome biology*, 7(12):R123, 2006.

[97] A. Hoffmann, A. Levchenko, M. L. Scott, and D. Baltimore. The IkappaB-NF-kappaB signaling module: temporal control and selective gene activation. *Science*, 298(5596):1241–1245, 2002.

[98] R. Hoffmann, L. Bruno, T. Seidl, A. Rolink, and F. Melchers. Rules for gene usage inferred from a comparison of large-scale gene expression profiles of T and B lymphocyte development. *Journal of Immunology*, 170(3):1339–1353, 2003.

[99] R. Hoffmann and F. Melchers. A genomic view of lymphocyte development. *Current Opinion in Immunology*, 15(3):239–245, 2003.

[100] R. Hoffmann, T. Seidl, M. Neeb, A. Rolink, and F. Melchers. Changes in gene expression profiles in developing B cells of murine bone marrow. *Genome Research*, 12(1):98–111, 2002.

[101] I. Holmes and W. J. Bruno. Finding regulatory elements using joint likelihoods for sequence and expression profile data. In *ISMB 2000: Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, volume 8, pages 202–10, 2000.

[102] D. Huang and W. Pan. Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. *Bioinformatics*, 22(10):1259–1268, 2006.

[103] L. J. Hubbert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:63–76, 1985.

[104] W. Huber, A. V. Heydebreck, H. Sultmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18 Suppl 1:i96–i104, 2002.

[105] G. Hyatt, R. Melamed, R. Park, R. Seguritan, C. Laplace, L. Poirot, S. Zucchelli, R. Obst, M. Matos, E. Venanzi, A. Goldrath, L. Nguyen, J. Luckey, T. Yamagata, A. Herman, J. Jacobs, D. Mathis, and C. Benoist. Gene expression microarrays: glimpses of the immunological genome. *Nature Immunology*, 7(7):686–691, 2006.

[106] 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(2):249–264, 2003.

[107] R. A. Irizarry, D. Warren, F. Spencer, I. F. Kim, S. Biswal, B. C. Frank, E. Gabrielson, J. G. N. Garcia, J. Geoghegan, G. Germino, C. Griffin, S. C. Hilmer, E. Hoffman, A. E. Jedlicka, E. Kawasaki, F. Martinez-Murillo, L. Morsberger, H. Lee, D. Petersen, J. Quackenbush,

A. Scott, M. Wilson, Y. Yang, S. Q. Ye, and W. Yu. Multiple-laboratory comparison of microarray platforms. *Nature Methods*, 2(5):345–350, 2005.

[108] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences of the United States of America*, 98(8):4569–4574, 2001.

[109] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1988.

[110] A. K. Jain and J. Moreau. Bootstrap techniques in cluster analysis. *Pattern Recognition*, 20:547–568, 1987.

[111] K. K. Jain. Biochips for gene spotting. *Science*, 294(5542):621–623, 2001.

[112] T. Joachims. *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms*. Kluwer/Springer, 2002.

[113] N. Kaminski and Z. Bar-Joseph. A patient-gene model for temporal expression profiles in clinical studies. *Journal of Computational Biology*, 14(3):324–338, 2007.

[114] M. Kanehisa, S. Goto, M. Hattori, K. F. AokiKinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34(Database issue):D354–D357, 2006.

[115] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 1990.

[116] A. Kazanjian, E. A. Gross, and H. L. Grimes. The growth factor independence-1 transcription factor: New functions and new insights. *Critical reviews in oncology/hematology*, 2006.

[117] S. V. E. Keränen, C. C. Fowlkes, C. L. L. Hendriks, D. Sudar, D. W. Knowles, J. Malik, and M. D. Biggin. Three-dimensional morphology and gene expression in the drosophila blastoderm at cellular resolution ii: dynamics. *Genome biology*, 7(12):R124, 2006.

[118] D. Klein, S. Kamvar, and C. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *ICML '02: Proceedings of the 19th international conference on Machine learning*, 2002.

[119] I. S. Kohane, A. J. Butte, and A. Kho. *Microarrays for an Integrative Genomics*. MIT Press, Cambridge, MA, USA, 2002.

[120] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[121] T. Koski. *Hidden Markov Models for Bioinformatics*. Kluwer Academic Publishers, 2001.

[122] R. Krause, C. von Mering, P. Bork, and T. Dandekar. Shared components of protein complexes–versatile building blocks or biochemical artefacts? *BioEssays : news and reviews in molecular, cellular and developmental biology*, 26(12):1333–1343, 2004.

[123] T. Lange, M. H. Law, A. K. Jain, and J. M. Buhmann. Learning with constrained and unlabelled data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 731–738, 2005.

[124] T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann. Stability-based validation of clustering solutions. *Neural Computing*, 16(6):1299–1323, 2004.

[125] S. L. Lauritzen. *Graphical Models*. Oxford University Press, USA, 1996.

[126] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B*, 50:157–224, 1988.

[127] E. Lecuyer, H. Yoshida, N. Parthasarathy, C. Alm, T. Babak, T. Cerovina, T. R. Hughes, P. Tomancak, and H. M. Krause. Global analysis of mrna localization reveals a prominent role in organizing cellular architecture and function. *Cell*, 131(1):174–187, 2007.

[128] T. Lee. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298(5594):799–804, 2002.

[129] M. Legendre, W. Ritchie, F. Lopez, and D. Gautheret. Differential Repression of Alternative Transcripts: A Screen for miRNA Targets. *PLoS Computational Biology*, 2(5):e43, 2006.

[130] M. Leptin. Gastrulation in drosophila: the logic and the cellular mechanisms. *The EMBO Journal*, 18:3187–3192, 1999.

[131] S. Z. Li. *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[132] X. Li, A. Schliep, and A. Schoenhuth. The Viterbi decomposition. Technical report, Center for Applied Computer Science, University of Cologne, 2004.

[133] L. P. Lim, N. C. Lau, P. GarrettEngele, A. Grimson, J. M. Schelter, J. Castle, D. P. Bartel, P. S. Linsley, and J. M. Johnson. Microarray analysis shows that some microRNAs downregulate large numbers of target mRNAs. *Nature*, 433(7027):769–773, 2005.

[134] R. J. Lipshutz, D. Morris, M. Chee, E. Hubbell, M. J. Kozal, N. Shah, N. Shen, R. Yang, and S. P. Fodor. Using oligonucleotide probe arrays to access genetic diversity. *BioTechniques*, 19(3):442–447, 1995.

[135] D. J. Lockhart and E. A. Winzeler. Genomics, gene expression and dna arrays. *Nature*, 405(6788):827–836, 2000.

[136] H. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, and J. E. Darnell. *Molecular Cell Biology*. W. H. Freeman & Co., 4th edition edition, 2000.

[137] Z. Lu and T. Leen. Semi-supervised learning with penalized probabilistic clustering. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 849–856. MIT Press, 2005.

[138] Y. Luan and H. Li. Clustering of time-course gene expression data using a mixed-effects model with B-splines. *Bioinformatics*, 19(4):474–482, 2003.

[139] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[140] P. Matthias and A. G. Rolink. Transcriptional networks in developing and mature B cells. *Nature Reviews Immunology*, 5(6):497–508, 2005.

[141] R. M. McIntyre and R. K. Blashfield. A nearest-centroid technique for evaluating the minimum-variance clustering procedure. *Multivariate Behavior Research*, 15:225–238, 1980.

[142] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, Inc., New York, Basel, 1988.

[143] 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.

[144] G. J. McLachlan and D. Peel. Robust cluster analysis via mixtures of multivariate t-distributions. In *SSPR '98/SPR '98: Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 658–666, London, UK, 1998. Springer-Verlag.

[145] G. J. McLachlan and D. Peel. *Finite Mixture Models.* Wiley Series in Probability and Statistics. Wiley, New York, 2000.

[146] J. McQueen. Some methods of classification and analysis of multivariate observations. In *5th Berkeley Symposium in Mathematics, Statistics and Probability*, pages 281–297, 1967.

[147] M. Medvedovic, K. Yeung, and R. Bumgarner. Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20(8):1222–1232, 2004.

[148] M. Meila and M. I. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48, 2001.

[149] G. W. Milligan and M. C. Cooper. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavior Research*, 21:441–458, 1986.

[150] C. Moller-Levet, F. Klawonn, K. Cho, and O. Wolkenhauer. Fuzzy clustering of short time-series and unevenly distributed sampling points. In *Advances in Intelligent Data Analysis V*, volume 2810 of *LNCS*, pages 330–340. Springer Verlag, 2003.

[151] S. Monticelli, K. M. Ansel, C. Xiao, N. D. Socci, A. M. Krichevsky, T. Thai, N. Rajewsky, D. S. Marks, C. Sander, K. Rajewsky, A. Rao, and K. S. Kosik. MicroRNA profiling of the murine hematopoietic system. *Genome biology*, 6(8):R71, 2005.

[152] H. Nakano, S. Sakon, H. Koseki, T. Takemori, K. Tada, M. Matsumoto, E. Munechika, T. Sakai, T. Shirasawa, H. Akiba, T. Kobata, S. M. Santee, C. F. Ware, P. D. Rennert, M. Taniguchi, H. Yagita, and K. Okumura. Targeted disruption of Traf5 gene causes defects in CD40- and CD27-mediated lymphocyte activation. *Proceedings of the National Academy of Sciences of the United States of America*, 96(17):9803–9808, 1999.

[153] B. Nelson and I. Cohen. Revisiting probabilistic models for clustering with pair-wise constraints. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 673–680, New York, NY, USA, 2007. ACM.

[154] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 13*, pages 849–856. MIT Press, 2001.

[155] S. K. Ng, G. J. McLachlan, K. Wang, L. BenTovim Jones, and S. Ng. A Mixture model with random-effects components for clustering correlated gene-expression profiles. *Bioinformatics*, 22(14):1745–1752, 2006.

[156] N. Niederberger, L. K. Buehler, J. Ampudia, and N. R. J. Gascoigne. Thymocyte stimulation by anti-TCR-beta, but not by anti-TCR-alpha, leads to induction of developmental transcription program. *Journal of leukocyte biology*, 77(5):830–841, 2005.

[157] C. Niehrs and N. Pollet. Synexpression groups in eukaryotes. *Nature*, 402(6761):483–487, 1999.

[158] OMIM. Online mendelian inheritance in man, http://www.ncbi.nlm.nih.gov/omim/.

[159] L. Opitz, A. Schliep, and S. Posch. Analysis of fused in-situ hybridization and gene expression data. In R. Decker and H. J. Lenz, editors, *Advances in Data Analysis, Proceedings of the 30th Annual Conference of the Gesellschaft fur Klassifikation*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 157–166, Heidelberg, Germany, 2006. Springer.

[160] J. Y. Pan, A. Guilherme, R. Balan, E. P. Xing, A. J. M. Traina, and C. Faloutsos. Automatic mining of fruit fly embryo images. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 693–698, New York, NY, USA, 2006. ACM Press.

[161] W. Pan. Incorporating gene functions as priors in model-based clustering of microarray gene expression data. *Bioinformatics*, 22(7):795–801, 2006.

[162] H. Peng, F. Long, M. B. Eisen, and E. W. Myers. Clustering gene expression patterns of fly embryos. In *Proceeding of the 3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro*, pages 1144–1147. IEEE, 2006.

[163] H. Peng and E. W. Myers. Comparing in situ mrna expression patterns of drosophila embryos. In *RECOMB '04: Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*, pages 157–166, New York, NY, USA, 2004. ACM.

[164] O. D. Perez, S. Kinoshita, Y. Hitoshi, D. G. Payan, T. Kitamura, G. P. Nolan, and J. B. Lorens. Activation of the PKB/AKT pathway by ICAM-2. *Immunity*, 16(1):51–65, 2002.

[165] L. Poirot, C. Benoist, and D. Mathis. Natural killer cells distinguish innocuous and destructive forms of pancreatic islet autoimmunity. *Proceedings of the National Academy of Sciences of the United States of America*, 101(21):8102–8107, 2004.

[166] S. H. Powis, I. Mockridge, A. Kelly, L. A. Kerr, R. Glynne, U. Gileadi, S. Beck, and J. Trowsdale. Polymorphism in a second ABC transporter gene located within the class II region of the human major histocompatibility complex. *Proceedings of the National Academy of Sciences of the United States of America*, 89(4):1463–1467, 1992.

[167] PubMed. http://www.ncbi.nlm.nih.gov/sites/entrez/.

[168] W. Qiu and H. Joe. Generation of random clusters with specified degree of separation. *Journal of Classification*, 23(2):315–334, 2006.

[169] L. R. Rabiner. A tutorial on Hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[170] H. S. Radomska, A. B. Satterthwaite, N. Taranenko, S. Narravula, D. S. Krause, and D. G. Tenen. A nuclear factor Y (NFY) site positively regulates the human CD34 stem cell gene. *Blood*, 94(11):3772–3780, 1999.

[171] S. H. Ramkissoon, L. A. Mainwaring, Y. Ogasawara, K. Keyvanfar, J. P. McCoy, E. M. Sloand, S. Kajigaya, and N. S. Young. Hematopoietic-specific microRNA expression in human cells. *Leukemia research*, 2005.

[172] M. F. Ramoni, P. Sebastiani, and I. S. Kohane. Cluster analysis of gene expression dynamics. *Proceedings of the National Academy of Sciences of the United States of America*, 99(14):9121–9126, 2002.

[173] C. Rangel, J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D. L. Wild, and F. Falciani. Modeling T-cell activation using gene expression profiling and state-space models. *Bioinformatics*, 20(9):1361–1372, 2004.

[174] H. Redestig, D. Weicht, J. Selbig, and M. Hannah. Transcription factor target prediction using multiple short expression time series from arabidopsis thaliana. *BMC bioinformatics*, 8(1):454, 2007.

[175] A. Reiner, D. Yekutieli, and Y. Benjamini. Identifying differentially expressed genes using false discovery rate controlling procedures. *Bioinformatics*, 19(3):368–375, 2003.

[176] K. Roeder. Density estimation with confidence sets exemplified by superclusters and voids in the galaxies. *Journal of the American Statistical Association*, 85(411):617–624, 1990.

[177] E. V. Rothenberg and T. Taghon. Molecular genetics of T cell development. *Annual Review of Immunology*, 23:601–649, 2005.

[178] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, 1987.

[179] S. Roweis and Z. Ghahramani. A Unifying Review of Linear Gaussian Models. *Neural Comp.*, 11(2):305–345, 1999.

[180] S. Sahu and G. Roberts. On convergence of the em algorithm and the gibbs sampler. *Statistics in Computing*, 9:55–64, 1998.

[181] A. A. Samsonova, M. Niranjan, S. Russell, and A. Brazma. Prediction of gene expression in embryonic structures of drosophila melanogaster. *PLoS Computational Biology*, 3(7):e144, 2007.

[182] J. Schafer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4:Article32, 2005.

[183] M. Schena. Genome analysis with gene expression microarrays. *BioEssays : news and reviews in molecular, cellular and developmental biology*, 18(5):427–431, 1996.

[184] A. Schliep. *Learning Hidden Markov Model topology*. PhD thesis, Center for Applied Computer Science, University of Cologne, 2001.

[185] A. Schliep, I. G. Costa, C. Steinhoff, and A. Schönhuth. Analyzing gene expression time-courses. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(3):179–193, 2005.

[186] A. Schliep, A. Schönhuth, and C. Steinhoff. Using Hidden Markov models to analyze gene expression time course data. *Bioinformatics*, 19 Suppl 1:i255–i263, 2003.

[187] A. Schliep, C. Steinhoff, and A. Schönhuth. Robust inference of groups in gene expression time-courses using mixtures of HMMs. *Bioinformatics*, 20 Suppl 1:i283–i289, 2004.

[188] T. Schmidt, H. Karsunky, B. Rodel, B. Zevnik, H. P. Elsasser, and T. Moroy. Evidence implicating Gfi-1 and Pim-1 in pre-T-cell differentiation steps associated with beta-selection. *EMBO Journal*, 17(18):5349–5359, 1998.

[189] A. Schönhuth, I. G. Costa, and A. Schliep. Semi-supervised clustering of yeast gene expression. In *Japanese-German Workshop on Data Analysis and Classification*. Springer, 2006.

[190] A. Schulze and J. Downward. Navigating gene expression using microarrays–a technology review. *Nature Cell Biology*, 3(8):E190–E195, 2001.

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

[192] E. Segal, M. Shapira, A. Regev, D. Peer, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):166–176, 2003.

[193] E. Segal, H. Wang, and D. Koller. Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, 19 Suppl 1:i264–i271, 2003.

[194] E. Segal, R. Yelensky, and D. Koller. Genome-wide discovery of transcriptional modules from dna sequence and gene expression. *Bioinformatics*, 19 Suppl 1:i273–i282, 2003.

[195] SGD. Saccharomyces genome database, http://www.yeastgenome.org/.

[196] N. Shental, A. BarHillel, T. Hertz, and D. Weinshall. Computing gaussian mixture models with em using equivalence constraints. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

[197] Y. Shi, T. Mitchell, and Z. Bar-Joseph. Inferring pairwise regulatory relationships from multiple time series datasets. *Bioinformatics*, 23(6):755–763, 2007.

[198] M. Shiga, I. Takigawa, and H. Mamitsuka. Annotating gene function by combining expression data with a modular gene network. *Bioinformatics*, 23(13):468–478, 2007.

[199] R. Sokal and F. Rohlf. *Biometry*. W. H. Freeman and Company, New York, 1995.

[200] P. Sood, A. Krek, M. Zavolan, G. Macino, and N. Rajewsky. Cell-type-specific signatures of microRNAs on target mRNA expression. *Proceedings of the National Academy of Sciences of the United States of America*, 103(8):2746–2751, 2006.

[201] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization. *Molecular Biology of the Cell*, 9(12):3273–3297, 1998.

[202] M. A. Steel and D. Penny. Distributions of tree comparison metrics-some new results. *Systematic Biology*, 42(2):126–141, 1993.

[203] C. Steinhoff and M. Vingron. Normalization and quantification of differential expression in gene expression microarrays. *Briefings in bioinformatics*, 7(2):166–177, 2006.

[204] R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig. The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics*, 18 Suppl 2:i231–i240, 2002.

[205] A. Stolcke and S. Omohundro. Hidden Markov model induction by Bayesian model merging. In *Advances in Neural Information Processing Systems 5*. 1992.

[206] J. D. Storey, W. Z. Xiao, J. T. Leek, R. G. Tompkins, and R. W. Davis. Significance analysis of time course microarray experiments. *Proceedings of the National Academy of Sciences of the United States of America*, 102(36):12837–12842, 2005.

[207] M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems 13*. MIT Press, 2001.

[208] Y. C. Tai and T. P. Speed. A multivariate empirical bayes statistic for replicated microarray time course data. *Annals Of Statistics*, 34:2387, 2006.

[209] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2981–2986, 2004.

[210] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18 Suppl 1:i136–i144, 2002.

[211] D. Tautz and C. Pfeifle. A non-radioactive in situ hybridization method for the localization of specific rnas in drosophila embryos reveals translational control of the segmentation gene hunchback. *Chromosoma*, 98(2):81–85, 1989.

[212] S. Tavazoie, J. Hughes, M. Campbell, R. Cho, and G. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–5, 1999.

[213] B. Thiesson, C. Meek, D. Chickering, and D. Heckerman. Learning mixtures of dag models. In *UIA 98: Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pages 504–51, San Francisco, CA, 1998. Morgan Kaufmann.

[214] P. Tomancak, A. Beaton, R.Weiszmann, E. Kwan, S. Shu, E. Lewis, S. Richards, M. Ashburner, V. Hartenstein, S. Celniker, and G. Rubin. Systematic determination of patterns of gene expression during Drosophila embryogenesis. *Genome Biology*, 3(12), 2002.

[215] P. Tomancak, B. Berman, A. Beaton, R. Weiszmann, E. Kwan, V. Hartenstein, S. Celniker, and G. Rubin. Global analysis of patterns of gene expression during drosophila embryogenesis. *Genome Biology*, 8(7):R145, 2007.

[216] S. Troncale, F. Tahi, D. Campard, J. Vannier, and J. Guespin. Modeling and simulation with hybrid functional petri nets of the role of interleukin-6 in human early haematopoiesis. In *PSB 2006: Proceeding of the Pacific Symposium on Biocomputing*, volume 11, pages 427–438, 2006.

[217] O. G. Troyanskaya. Putting microarrays in a context: integrated analysis of diverse biological data. *Briefings in bioinformatics*, 6(1):34–43, 2005.

[218] O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein. A bayesian framework for combining heterogeneous data sources for gene function prediction (in saccharomyces cerevisiae). *Proceedings of the National Academy of Sciences of the United States of America*, 100(14):8348–8353, 2003.

[219] V. G. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, 98(9):5116–5121, 2001.

[220] L. E. Tze, B. R. Schram, K. Lam, K. A. Hogquist, K. L. Hippen, J. Liu, S. A. Shinton, K. L. Otipoby, P. R. Rodine, A. L. Vegoe, M. Kraus, R. R. Hardy, M. S. Schlissel, K. Rajewsky, and T. W. Behrens. Basal immunoglobulin signaling actively maintains developmental stage in immature B cells. *PLoS Biology*, 3(3):e82, 2005.

[221] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas. Som toolbox for matlab. Technical report, Department of Computer Science and Engineering at the Helsinki University of Technology., 2000.

[222] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967.

[223] A. von Heydebreck, B. Gunawan, and L. Fuzesi. Maximum likelihood estimation of oncogenetic tree models. *Biostatistics*, 5(4):545–556, 2004.

[224] L. A. Warren and E. V. Rothenberg. Regulatory coding of lymphoid lineage choice by hematopoietic transcription factors. *Current Opinion in Immunology*, 15(2):166–175, 2003.

[225] B. Weinstock-Guttman, D. Badgett, K. Patrick, L. Hartrich, R. Santos, D. Hall, M. Baier, J. Feichter, and M. Ramanathan. Genomic effects of IFN-beta in multiple sclerosis patients. *Journal of Immunology*, 171(5):2694–702, 2003.

[226] M. L. Whitfield, G. Sherlock, A. J. Saldanha, J. I. Murray, C. A. Ball, K. E. Alexander, J. C. Matese, C. M. Perou, M. M. Hurt, P. O. Brown, and D. Botstein. Identification of genes periodically expressed in the human cell cycle and their expression in tumors. *Molecular Biology of the Cell*, 13(6):1977–2000, 2002.

[227] M. P. Windham and A. Cutler. Information ratios for validating mixture analyses. *Journal of the American Statistical Association*, 87(420):1188–1192, 1992.

[228] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In S. T. S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press, 2003.

[229] T. Yamagata, C. Benoist, and D. Mathis. A shared gene-expression signature in innate-like lymphocytes. *Immunological reviews*, 210:52–66, 2006.

[230] 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 Acids Research*, 30(4):e15, 2002.

[231] C.-H. Yeang and T. Jaakkola. Time series analysis of gene expression and location data. *International Journal on Artificial Intelligence Tools*, 14(5):755–770, 2005.

[232] 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.

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

[234] K. Y. Yeung, M. Medvedovic, and R. E. Bumgarner. Clustering gene-expression data with repeated measurements. *Genome biology*, 4(5):R34, 2003.

[235] M. Yuan and C. Kendziorski. Hidden markov models for microarray time course data in multiple biological conditions. *Journal Of The American Statistical Association*, 101(476):1323–1332, 2006.

[236] G. Zhu, P. Spellman, T. Volpe, P. Brown, D. Botstein, T. Davis, and B. Futcher. Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth. *Nature*, 406(6791):90–4, 2000.

[237] X. S. Zhu, M. W. Linhoff, G. Li, K. C. Chin, S. N. Maity, and J. P. Ting. Transcriptional scaffold: CIITA interacts with NF-Y, RFX, and CREB to cause stereospecific regulation of the class II major histocompatibility complex promoter. *Molecular and Cellular Biology*, 20(16):6051–6061, 2000.

Appendix A

Gene Ontology enrichment

In order to find GO terms with annotations related to a given group (or cluster) of genes, one should look for annotation terms that are over-represented in this group. The probability that this over-representation is not found by chance can be measured with the use of a hyper-geometric Fisher exact test [199]. This test returns for each cluster and gene ontology term a p-value describing how statistically significant a GO term is for describing genes in a particular cluster.

Let n be the total number of annotated genes in GO (reference group), and m be the number of genes annotated with a specific GO term. This will give us m positive genes and $n - m$ negative genes. If we draw k genes from the reference group (or analogously obtain a cluster with k genes), we obtain q positive genes and $k - q$ negative genes, see Table A.1 for a 2X2 contingency table representation of these terms. We are interested in observing how unusually large this value q is, given n, m and k. This can be achieved by calculating a p-value defined by $p(X \geq q)$, where X is defined by $\{\mathbf{P}(x = i)\}_{1 \leq i \leq k}$, and $\mathbf{P}(x = i)$ is defined as below:

$$\mathbf{P}(x = i) = \frac{\binom{m}{i}\binom{n-m}{k-i}}{\binom{n}{k}}$$

In the thesis, when a particular GO term is over-represented for a given cluster, we state GO Term X is enriched in cluster Y, or we found enrichment for GO Term X in cluster Y.

A later correction of the p-values is necessary, because of the effects of multiple testing. For example, if we have 1000 GO terms, and a p-value of 0.1 is used, at least 100 false

Table A.1: *2x2 Contingency Table for genes annotated or not annotated by a given GO term*

	Annotated Genes	Non-annotated Genes	Total
in cluster	q	$k - q$	k
not in cluster	$m - q$	$(n - k) - (m - q)$	$n - k$
Total	m	$n - m$	n

positives are expected. To correct this, we apply a false positive discovery ratio proposed in [175].

Appendix B

Analysis of Gene Expression of Lymphoid Development

Table B.1: *Contingency Table comparing results from* `MixDTrees-Dev` *(columns) versus SOM (lines) for* `TCell`

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	41	24	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4
3	6	38	14	1	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
6	2	1	1	14	2	11	2	2	0	6	0	0	0	0	0	0	0	0	0	1
2	4	12	31	32	25	13	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	0	1	10	0	13	1	0	0	1	0	0	0	1	0	0	0	0	3	0	2
5	0	0	0	35	8	88	3	34	0	4	0	0	0	0	0	0	0	0	0	0
10	0	0	1	0	0	1	15	6	9	1	0	0	0	1	0	0	1	3	0	0
14	0	0	0	0	0	0	10	7	2	23	9	0	0	0	0	0	3	0	0	0
15	0	0	0	0	0	0	0	0	19	0	0	16	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	3	35	0	49	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	17	0	12	21	1	2	0	0	1	0	0	0
18	0	0	0	0	0	0	0	0	4	0	0	47	18	2	0	0	1	0	0	0
12	2	0	0	0	0	0	0	0	1	0	0	4	11	5	7	4	8	5	2	1
17	0	0	0	0	0	0	0	0	5	0	4	0	7	4	7	1	8	0	2	0
19	0	0	0	0	0	0	0	0	0	0	0	6	15	35	40	4	27	0	0	0
13	2	0	0	0	0	0	0	0	0	0	0	0	0	4	7	34	21	0	6	1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	24	23	0	0	0
4	4	0	5	2	0	0	5	0	3	0	0	1	0	0	0	0	0	11	0	0
11	1	0	2	0	3	0	0	1	0	0	1	0	0	0	0	6	2	0	3	10
7	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	12	1	0	11	13

Table B.2: *Contingency Table comparing results from* `MixDTrees-Dev` *(columns) versus SOM (lines) for* `BCell`

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	52	0	0	0	0	4	5	4	0	0	0	3	0	0	0	0	0	0	0	0
4	0	20	12	10	0	4	0	0	1	14	0	0	0	0	0	0	0	0	0	0
7	0	6	64	5	25	2	0	0	22	0	0	0	0	0	0	0	0	0	0	0
2	14	3	8	4	2	40	0	0	0	0	0	1	0	0	0	0	0	0	0	0
15	0	0	1	0	0	1	1	0	0	2	0	0	0	0	0	0	0	0	0	0
6	0	7	43	5	10	42	2	0	3	0	0	0	0	0	0	0	0	0	0	0
5	4	1	0	0	0	1	4	1	0	2	1	0	0	0	0	0	0	0	0	0
3	5	3	0	1	0	5	0	7	0	5	0	0	0	0	0	0	0	0	0	0
8	0	7	3	0	0	0	0	0	9	13	0	0	1	0	0	0	0	0	0	0
9	0	1	1	10	0	0	0	7	1	17	0	0	3	0	0	0	0	0	0	0
16	0	0	0	0	0	0	4	1	0	0	4	2	0	2	1	1	18	0	0	0
20	1	0	0	0	0	1	1	3	0	0	7	14	0	0	0	0	0	0	0	0
10	0	0	0	3	0	0	0	0	8	6	0	0	18	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	7	0	0	0	0	2	6	18	2	6	1	0	1
13	0	0	0	0	0	0	0	1	0	0	1	0	0	0	20	13	1	0	0	8
14	0	0	0	0	0	0	0	2	0	0	2	0	0	0	3	28	3	8	4	6
17	0	0	0	0	0	0	0	0	0	0	1	0	0	1	3	25	24	0	0	3
19	0	0	0	0	0	0	2	0	0	1	5	12	0	0	0	4	0	35	18	0
18	0	0	0	0	0	0	0	0	0	0	14	9	0	0	0	24	8	3	18	0
11	0	0	0	0	0	0	0	3	0	1	0	0	14	1	19	0	1	0	0	8

Table B.3: *MicroRNA enrichment per cluster for* `TCell` *for* `MixDTrees-Dev`

Cluster ID	MicroRNA	p-value
3	miR-222	0.0006906
5	miR-15a	0.0019456
	miR-26a	0.0369906
	miR-24	0.0369906
	miR-221	0.0051746
	miR-181a	0.0244306
7	miR-342	0.0200686
8	miR-26a	0.0013526
10	miR-150	0.0012176
	miR-142-3p	0.0000056
11	miR-16	0.0049776
	miR-146	0.0011936
	miR-181b	0.0049776

Table B.4: *MicroRNA enrichment per cluster for* `BCell` *for* `MixDTrees-Dev`

Cluster ID	MicroRNA	p-value
3	miR-26a	0.0358116
	miR-181c	0.0025866
	miR-181b	0.0358116
5	miR-15b	0.0029956
	miR-15a	0.0029956
	miR-223	0.0029956
	miR-221	0.0323296
6	miR-191	0.0486736
	miR-155	0.0271276
19	miR-342	0.0402686
	miR-142-3p	0.0088346

Appendix C

Notation

All chapters

$\mathbf{1}(e)$ indicator function, which takes value 1 iff e is true

α_k mixture coefficient of the kth mixture component

$E[X]$ expectation of a random variable X

$\mathcal{L}$ likelihood function

K number of clusters or components in a mixture model

μ_x mean value of random variable X

$p(x \mid \theta)$ a probability density function over variable X and parameterized by θ

r_{ik} posterior probability that observation x_i is assigned to the kth mixture component, i.e., $p(y_i = k \mid x_i, \Theta)$

Σ_x covariance matrix of random variable X

Θ set of parameters of a mixture model

θ_k set of parameters of the kth mixture component

X an L dimensional continuous random variable

x an observation vector $(x_1, ..., x_L)$ from X

$\mathbf{X}$ a data set represented by a $N \times L$ matrix, where entry x_{ij} denotes the values of the jth variable from the ith observation

Y an one dimensional discrete random variable

y an observation of Y, where $y \in \{1, ..., K\}$ indicates the mixture component (or cluster) the observation belongs

$\mathbf{Y}$ a set of N observations from Y, where $y_i = k$ denotes that the ith observation belongs to the kth mixture component (or mixture)

$\mathcal{Y}$ space of all possible values of $\mathbf{Y}$

Chapter 4

A transition matrix of a HMM, where a_{uv} represents the probability of going from state u to state v

d_u duration parameter representing the expected number of visits to state u

M	number of states of the HMM
μ_u	mean parameter of the emission function of the uth state
π_u	probability of visiting state u at time $t = 1$
Q	an L-dimensional discrete variable representing the sequence of visited states
q	observation from Q, where $q = (q_1, ..., q_t, ..., q_L)$ and $q_t \in \{1, ..., M\}$ represents the HMM state visited at time t.
σ_u^2	standard error parameter of the emission function from the uth state
θ_L	parameters of a linear HMM

Chapter 5

$\mathrm{D}(p\|\|p^*)$	relative entropy between the pdfs p and p^*
$\mathrm{H}(X)$	entropy of variable X
$\mathrm{I}(X_u, X_v)$	mutual information of variables X_u and X_v
$p^T(x\|\Theta)$	dependence tree pdf
$p(x_u\|x_v, \tau_u)$	conditional Gaussian pdf
pa	parent map defining the dependence tree structure
$\sigma_{u\|v}^2$	standard error of the conditional Gaussian pdf
τ_u	parameters of a conditional Gaussian pdf
$w_{u\|v}$	regression parameter of the conditional Gaussian pdf

Chapter 6

λ^+	parameter defining the penalty weights of positive constraint violations
λ^-	parameter defining the penalty weights of negative constraint violations
W	pair (W^+, W^-) representing the positive and negative constraint matrices
W^+	positive constraints matrix, where w_{ij}^+ is the positive constraint value for observations i and j
W^-	negative constraints matrix, where w_{ij}^- is the negative constraint value for observations i and j
Z	an L-dimensional continuous random variable
z	an observation $(z_{i1}, ...z_{il}, ..., z_{iL})$ of Z representing the pixel intensities of an image

Appendix D

Abbreviations

BCell	B cell development data
Bimm	immature B cells
BMC	Bayesian model collection
Bpre	pre B cells
Bpro	pro B cells
BIC	Bayesian information criteria
CL	co-location index
CLP	common lymphoid progenitor
CMP	common myeloid progenitor
CR	corrected Rand index
DAG	directed acyclic graph
DN	CD4-/CD8- double negative cells
DPL	CD4+/CD8+ double positive large cells
DPS	CD4+/CD8+ double positive small cells
DTree	dependence tree
ECR	extended corrected Rand index
ED	equal density
EM	expectation-maximization algorithm
E-Step	expectation step
FACS	fluorescence activated cell sorting
GQL	Graphical Query Language
GO	Gene Ontology
ImaGO	Image Gene Ontology
HemoMIR	hematopoiesis related microRNAs data
HMM	hidden Markov model
HMRF	hidden Markov random fields
KEGG	Kyoto encyclopedia of genes and genomes
KMC	k-means model collection
lHMM	linear hidden Markov model
MAP	maximum-a-posteriori

MCMC	Monte Carlo Markov Chain
mir	microRNA
MixDTrees	mixture of dependence trees
MixDTrees-Dev	MixDTrees with the developmental tree as structure
MixDTrees-Str	MixDTrees with estimated structure
MLE	maximum likelihood estimation
MM	probe mismatch
MoG	mixture of multivariate Gaussians
MoG Full	MoG with full covariance matrix
MoG Diag	MoG with diagonal covariance matrix
M-Step	maximization step
NK	natural killer cells
NMF	non-negative matrix factorization
PC	Pearson correlation
pdf	probability density function
pHSC	pluri-potent, self-renewing hematopoietic stem cells
PM	probe match
PPP	pluripotent progenitor
RMC	random model collection
SCC	strongly connected components
Sens	sensitivity index
SIM	simulated data
SOM	self-organizing maps
SSL	semi-supervised learning
Spec	specificity index
SP4	single positive CD4
SP8	single positive CD8
TCell	T cell development data
TCD4	cd4 T cells
TCD8	cd8 T cells
TDN	double negative T cells
TF	transcription factor
TFBS	transcription factor binding site
TNK	natural killer T cells
TR	transcription regulation data
YCC	yeast cell cycle
VD	Viterbi decomposition

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen verwendet habe.

Berlin, Mai 2007 Ivan Gesteira Costa Filho