THE UNIVERSITY *of* EDINBURGH

Inference Dynamics in Transcriptional Regulation

Hafiz Muhammad Shahzad Asif

Doctor of Philosophy

Institute for Adaptive and Neural Computation

School of Informatics

University of Edinburgh

2012

Abstract

Computational systems biology is an emerging area of research that focuses on understanding the holistic view of complex biological systems with the help of statistical, mathematical and computational techniques. The regulation of gene expression in gene regulatory network is a fundamental task performed by all known forms of life. In this subsystem, modelling the behaviour of the components and their interactions can provide useful biological insights. Statistical approaches for understanding biological phenomena such as gene regulation are proving to be useful for understanding the biological processes that are otherwise not comprehensible due to multitude of information and experimental difficulties. A combination of both the experimental and computational biology can potentially lead to system level understanding of biological systems.

This thesis focuses on the problem of inferring the dynamics of gene regulation from the observed output of gene expression. Understanding of the dynamics of regulatory proteins in regulating the gene expression is a fundamental task in elucidating the hidden regulatory mechanisms. For this task, an initial fixed structure of the network is obtained using experimental biology techniques. Given this network structure, the proposed inference algorithms make use of the expression data to predict the latent dynamics of transcription factor proteins.

The thesis starts with an introductory chapter that familiarises the reader with the physical entities in biological systems; then we present the basic framework for inference in transcriptional regulation and highlight the main features of our approach. Then we introduce the methods and techniques that we use for inference in biological networks in chapter 2; it sets the foundation for the remaining chapters of the thesis. Chapter 3 describes four well-known methods for inference in transcriptional regulation with pros and cons of each method.

Main contributions of the thesis are presented in the following three chapters. Chapter 4 describes a model for inference in transcriptional regulation using state space models. We extend this method to cope with the expression data obtained from multiple independent experiments where time dynamics are not present. We believe that the time has arrived to package methods like these into customised software packages tailored for biologists for analysing the expression data. So, we developed an open-sources, platform independent implementation of this method (TFInfer) that can process expression measurements with biological replicates to predict the activities of proteins and their influence on gene expression in gene regulatory network.

The proteins in the regulatory network are known to interact with one another in regulating the expression of their downstream target genes. To take this into account, we propose a novel method to infer combinatorial effect of the proteins on gene expression using a variant of fac-

torial hidden Markov model. We describe the inference mechanism in *combinatorial factorial hidden model* (cFHMM) using an efficient variational Bayesian expectation maximisation algorithm. We study the performance of the proposed model using simulated data analysis and identify its limitation in different noise conditions; then we use three real expression datasets to find the extent of combinatorial transcriptional regulation present in these datasets. This constitutes chapter 5 of the thesis.

In chapter 6, we focus on problem of inferring the groups of proteins that are under the influence of same external signals and thus have similar effects on their downstream targets. Main objectives for this work are two fold: firstly, identifying the clusters of proteins with similar dynamics indicate their role is specific biological mechanisms and therefore potentially useful for novel biological insights; secondly, clustering naturally leads to better estimation of the transition rates of activity profiles of the regulatory proteins. The method we propose uses Dirichlet process mixtures to cluster the latent activity profiles of regulatory proteins that are modelled as latent Markov chain of a factorial hidden Markov model; we refer to this method as DPM-FHMM. We extensively test our methods using simulated and real datasets and show that our model shows better results for inference in transcriptional regulation compared to a standard factorial hidden Markov model.

In the last chapter, we present conclusions about the work presented in this thesis and propose future directions for extending this work.

Acknowledgements

First of all, I would like to thank my research supervisor, Dr. Guido Sanguinetti, for his continuous support and guidance in the past three and a half years of PhD study. I am also thankful to Prof. Visakan Kadirkamanathan and Dr. Dawn Walker for useful suggestions and comments during my stay at University of Sheffield. I would like to thank Prof. Dirk Husmeier and Prof. J. Douglas Armstrong for their valuable feedback during the later half of my PhD.

I would like to thank my friends and former lab members Maurizio Filippone and Greg Skolidis for their support and encouragement during the times I needed it. I would also like to thank Andrea Ocone for accompanying me to coffee breaks during the last year.

I would like to thank University of Engineering and Technology, Lahore and School of Informatics, University of Edinburgh for their financial support.

Last but not the least, I am thankful for the great support I received from my parents, family members and anonymous friends during the ups and downs of my PhD study.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Hafiz Muhammad Shahzad Asif*)

Dedicated to my parents. . .

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

This chapter provides the background on biological systems and introduces the terminologies used throughout this thesis. It starts with a discussion about the importance of system level understanding of the biological systems. Then it introduces the biology of gene regulation while identifying key components of basic biological systems. Experimental techniques used to obtain the quantitative measurements of these biological systems are briefly discussed while identifying the potential sources of noises in these measurements. Then it describes the approach followed in this thesis to analyse the data obtained from biological systems. Finally, this chapter provides a summary of the rest of the chapters of the thesis and highlights the main contributions of the thesis.

1.1 Systems Biology

Biological systems are comprised of large sub-systems that interact selectively and nonlinearly to produce coherent behaviour. The sub-systems in complex biological systems are often diverse and multi-functional in nature. This behaviour heavily depends on combination of elements and the specific elements in the sub-systems. Neither the sub-systems nor the elements of the sub-systems can produce the same functionality in isolation due to the symbiotic nature of the underlying system. To understand the behaviour of biological systems, experimental and computational research is combined to get system-level view of these complex systems. This approach is often referred to as *Systems biology*. Systems biology is an emerging field that can potentially unveil the basic functionality of living organisms and can lead to breakthroughs in medical science and engineering.

Molecular biology, on the other hand, focuses on the individual elements of complex biological systems. It states that the complex behaviour of biological systems is the result of the

interaction of these simple elements. Molecular biology has produced a large volume of information related to genome sequence and protein properties. This information, alone, can not help to understand the basic functionalities of biological systems as the interactions between the components of these complex biological systems are poorly understood. Also, these biological systems are the result of evolution so focusing on the system-level understanding can help to solve the mysteries of complex biological processes. This holistic view is the main driving force behind the approach advocated in systems biology.

Computational approaches in systems biology (usually referred to as computational systems biology (Kitano, 2002)) are necessary to tackle the multitude of information. Even in the simplest of living organisms such as unicellular bacteria, the amount of experimental measurements and related biological information is so vast that it is not possible to analyse all that without efficient computational techniques. Also, poorly understood biological phenomena can be modelled in computational models that have proven to provide useful biological insights. Due to the intrinsic complexity of biological systems and vast amount of experimental data, a combination of experimental and computational approaches promises to provide deeper understanding of biological systems.

1.2 Biological Systems

All living organisms consist of one or more cells. The cells have a membrane that separates the internal components of the cellular machinery from the external environment. Among other components of the cellular machinery such as organelles that are required for various cellular functions, the most important one is the genetic material that is responsible for producing various types of proteins and enzymes required for the important cellular functions and for the survival of cells. The genetic material is compartmentalised within nucleus in case of *eukaryotes* (including multi-cellular organisms) whereas the *prokaryotes* (bacteria and archaea) lack a defined boundary to separate the genetic material from the rest of the cellular machinery. The genetic material consists of double stranded *deoxyribonucleic acid* (DNA) which is mainly used to store the genetic information for development and functioning of the cells. DNA is one of the three types of *biopolymer* that is produced by living organisms; other two types are *ribonucleic acid* (RNA) and proteins.

The DNA in the cell is organised into long structures called *chromosomes*. DNA consists of two strands of *nucleotide* joined together to form a helix. These nucleotides are nucleic acid units that serve as the basic building blocks of DNA. It is the sequence of these nucleotides that stores the genetic information. Four nucleotides are present in a DNA strand: adenine(A),

guanine (G), cytosine (C) and thymine (T). A *gene* is a segment of DNA that contains long sequence of nucleotides encoding the instructions for the production of a particular type of protein.

A *genome* consists of the collection of all the chromosomes inside the cell. The information encoded in the form of chromosomes contains the blueprint required for the synthesis of proteins that are of vital importance. The process of synthesising proteins from the information stored in DNA is called *gene expression*. Understanding of gene expression is of paramount importance as this process is the core function performed by all known forms of life. Gene expression process serves the basis for cellular differentiation and mainly controls function and behaviour of cells. The genetic code stored in the genetic material is interpreted by gene expression which gives rise to organism's *phenotype*.

1.2.1 The Regulation of Gene Expression

Biological cells are made up of several thousand proteins that interact with one another. Each cell produces different proteins while sensing different environmental conditions e.g., when sugar molecules are sensed, the cells react by producing enzymes that can transport the sugar into the cell. Gene expression is the process that produces all the proteins required for the survival and functioning of living cells.

The production of proteins based on the encoded instructions in the gene requires other components of the regulatory machinery to work in an orchestral manner. Generally, all the genes contain a regulatory region called *promoter* (Fig. 1.1). An enzyme called RNA polymerase (RNAp) binds to the promoter region of a gene and open the DNA double helix to start reading (transcribing) the encoded sequence to generate messenger RNA (mRNA) which is a complementary copy of the nucleotide sequence encoded by the gene. This is the first step of gene expression and is called *transcription*. The direct interaction between genes and TFs is the simplest form of *transcriptional regulation*. The mRNA produced at this stage of gene expression is not in the mature form and needs processing to become mature mRNA. Next major step in gene regulation (excluding the post-transcriptional modification of the mRNA produced in case of eukaryotes) is the translation of mRNA to functional products called proteins.

During transcription, the RNAp binds to the promoter region of almost all the genes. The rate of transcription is, however, mainly governed by special proteins called transcription factor (TF) proteins. TFs are synthesised as the result of transcription of genes in a cell which are in turn regulated by other TFs. TFs change the transcription rate of their target genes by binding to the specific sites in promoter region of their target genes (cis-regulatory elements, Fig. 1.1).

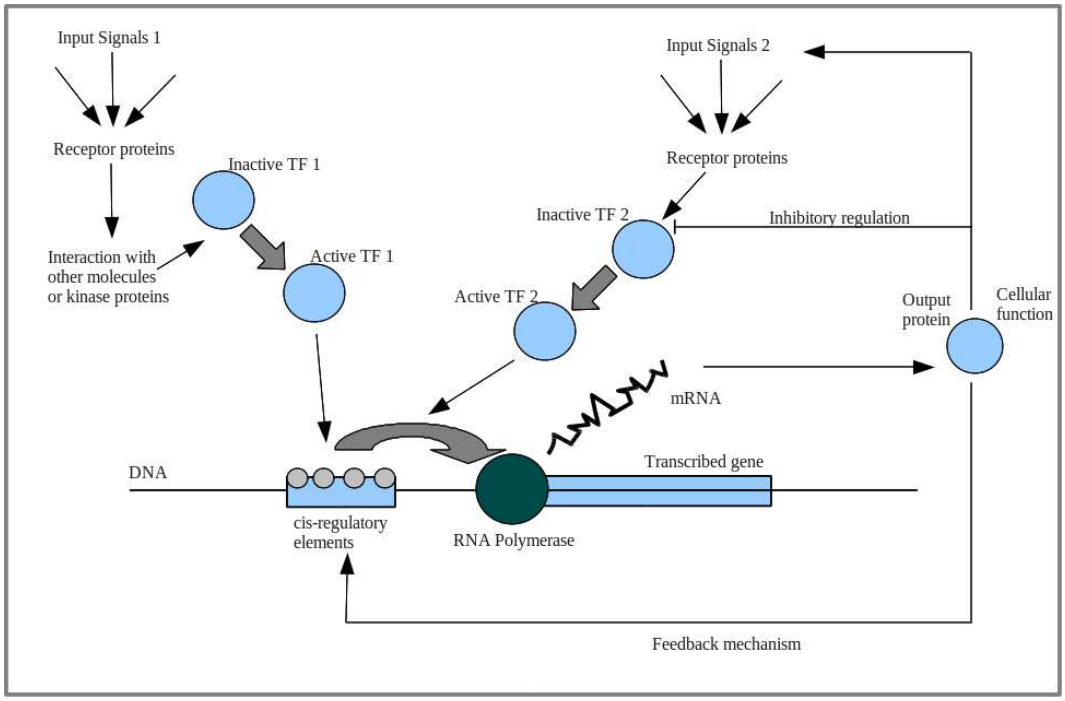


Figure 1.1: The regulation of gene expression.

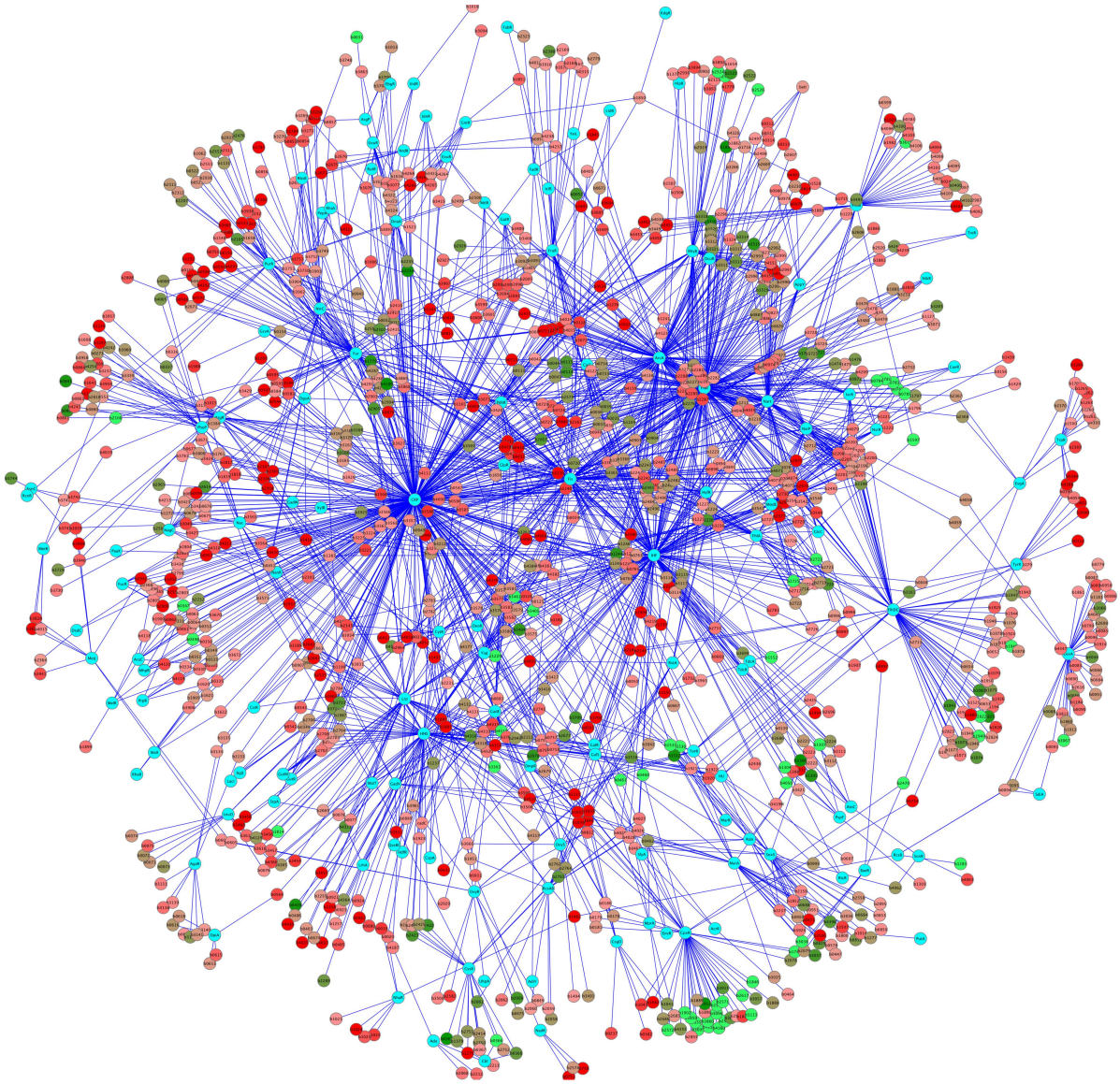


Figure 1.2: The regulatory network of *E.coli* with expression data measurements taken from (Graham et al., 2011); red colour of the nodes in the network shows under-expressed genes while green color shows over-expressed genes. The nodes with blue colour are the TFs that are regulating these genes.

When a TF is bound to a target gene, it changes the affinity (probability per unit time) that RNAp also binds the promoter to produce a mRNA molecule.

TFs can increase or decrease the rate of transcription of their target genes based on which they are categorised as *activators* or *repressors*. Depending on a specific environmental change, these transcription factors usually change from active to inactive state. Active transcription controls the rate at which specific target genes are transcribed into mRNA and translated into proteins. This set of interactions among TF proteins, genes and other cellular components form a network called *gene regulatory network* (GRN, figure 1.2). GRN is a dynamical system that determines the rate of production of different proteins.

Generally, each mRNA molecule is translated to a protein which may serve a wide range of purposes. In some situations, protein will accumulate at the cell-wall to serve the structural need. In some other cases, these proteins are enzymes that are used to speedup a chemical reaction. The rest of them carry out other functions of living cells such as repairs within the cell.

1.2.2 Combinatorial Transcriptional Regulation

It is understood that the process of transcription for a particular gene is under the control of multiple TFs where the interactions between TFs regulating the target gene play an important role. The combinatorial control of multiple TFs over the expression of a gene have different biological functions: this can result in differential expression of the target gene; it can also act as a step in transcription whereby multiple signals from different environmental stimuli are integrated. The interactions between TFs can be in different forms too: TFs form protein complexes that regulate the target gene; multiple TFs bound to the promoter region of the target gene at the same time and contribute towards the expression of the gene at different rates; all the TFs having combinatorial control over the expression of the gene are only required to be bound during transcription. It is due to the combinatorial transcriptional regulation that two interacting TFs with low concentrations are more likely to transcribe the target gene compared to when only one TF with low concentration is bound to the target gene (in which case transcription will not be initiated due to the low concentration of the single regulator). In case when two TFs are bound to the target gene simultaneously, and the binding sites of the regulators are not adjacent, the combinatorial control requires the intervening DNA to be looped to facilitate the interactions.

There are many regulatory proteins that have combinatorial control over the expression of their target genes in yeast regulatory network and in higher-level organisms in particular.

In yeast regulatory network, TF Pho2 is known to act in cooperation with other TFs in the network. It requires Pho4 to activate the transcription of Pho5 and Swi5 for the transcription of HO respectively (Bhoite et al., 2002). Another example of the combinatorial regulation is human interferon-β gene which is only regulated when all three of its regulators are bound to it in the active form. This shows the powerful role played by combinatorial transcription regulation in integrating the physiological signals as the three activator of interferon-β gene are actually driven by three signal transduction pathways (Ptashne and Gann, 2002).

1.3 Experimental Methods

To study the regulation of gene expression, we need to measure the mRNA expression levels of the genes experimentally in response to different environmental signals. The changes in the expression profile of a gene indicate that the gene is playing an important role under the experimental conditions by altering the rate of production of the encoded proteins under the influence of TFs. Measuring the proteins produced during gene expression would be ideal to analyse the gene regulation; however, experimental difficulties make it very hard to measure it. The mRNA expression levels of genes are relatively easier to measure owing to technological advancements such as DNA microarrays.

Chromatin Immunoprecipitation (ChIP) with Microarray (chip) or ChIP-on-chip is microarray based technology that is used to analyse the binding of specific proteins to DNA sequences on a genome-wide scale. These type of proteins are more commonly found in the *chromatin* of the nucleus. The chromatin is the collection of DNA and proteins that comprise the nucleus of the cell. Using ChIP-on-chip, the interactions of proteins of interest such as TFs with gene sequences can be obtained; this set of interactions can be viewed as a static picture (or *wiring*) of the GRN. This architectural information proves to be useful in statistical modelling of regulatory interactions. We will describe these methods in next sections.

1.3.1 Microarray Technology

DNA microarray technology has made it possible to measure the expression profiles of large number of genes in a genome. A DNA microarray is a solid surface with thousands of microscopic DNA spots. Each DNA spot on the microarray, called *probe*, contains a small amount of a particular DNA sequence which is used to attract the complementary DNA (cDNA) sequence of the sample. The main idea behind DNA microarray is *hybridisation* of complementary DNA strands (figure 1.3). Complementary DNA sequences have the property that the complemen-

tary strands of DNA will pair with each other due to the complementary nucleotide base pairs. DNA strands with higher number of complementary base pairs will have stronger bonds and thus will remain hybridised after washing off. The sample whose expression level is to be measured is fluorescently labelled and after binding to its cDNA generates a signal that depends on the strength of the hybridisation. Total strength of this signal from the spot on the microarray depends on the amount of the sample bound to the probes at that spot. Then the intensity of the microarray spot (under the influence of experimental conditions or query sample) is compared to the intensity of the reference microarray spot to assess what are changes in the expression level due to the changes in environmental/experimental conditions.

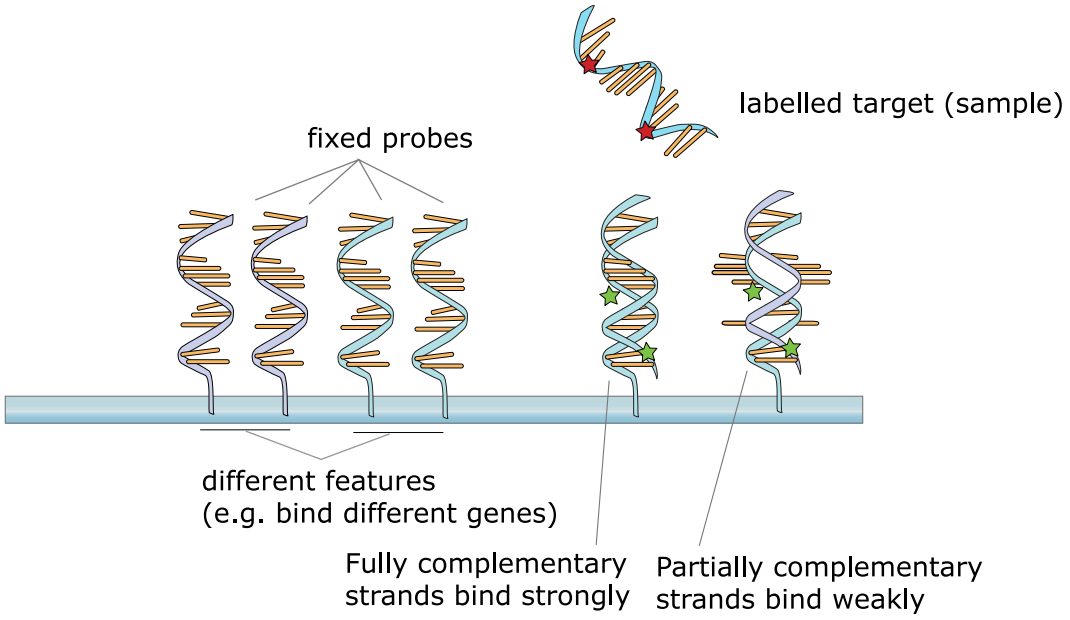


Figure 1.3: Hybridisation of the target to the probe in DNA microarray (Wikipedia, 2012b).

The underlying assumption of microarray data analysis is that the strength of the signal from microarray represents its relative expression. In order to compare the measured levels (or intensity of the signal), normalisation of the measured intensities is required to make meaningful comparison. In order to find those genes which significantly over-expressed or under-expressed given the query and reference sample (say Q and R respectively), then the relative expression level of gene i can be computed as

$$G_i = \frac{Q_i}{R_i} \tag{1.1}$$

This ratio provides a measure for characterising the genes based on their expression levels. These ratios are also termed as *fold changes*. Using this measure, gene with fold change of two

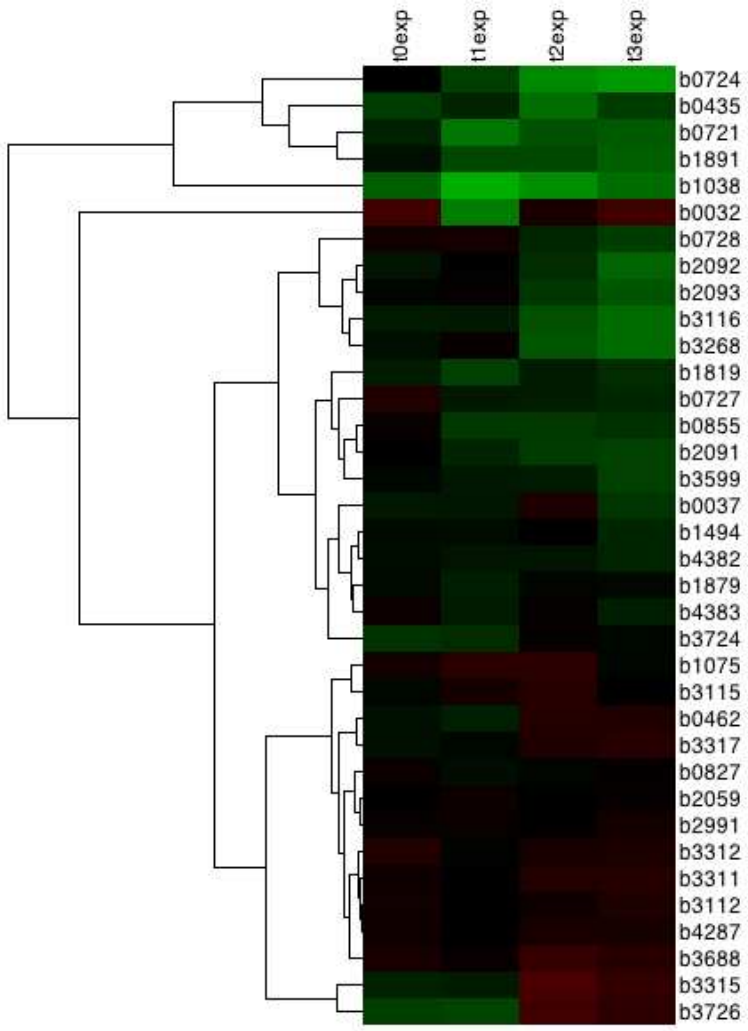


Figure 1.4: Gene expression values from microarray experiments can be represented as heat maps to visualise the result of data analysis. This heat map shows the expression values of a subset of genes from Graham et al. (2011); genes expression measurements are clustered using hierarchical clustering algorithm.

can be considered as up-regulated by factor of two; however, genes that are down-regulated by a factor of 2 have fold change which is 0.5. To overcome this problem, fold change ratios are usually analysed after taking the logarithm (base 2) which produces a continuous spectrum of values. Figure 1.4 shows the expression measurements of genes ($\log_2$ fold changes) in the form of a heat map where the these expression measurements are also clustered using hierarchical clustering in Cytoscape (Smoot et al., 2011; Morris et al., 2011). More transformation and normalisation techniques for microarray data are described in Quackenbush et al. (2002).

1.3.1.1 Sources of Noise in Microarray Experiments

Primarily, there are two sources of noise in gene expression measurements: biological and technical.

The process of gene regulation is intrinsically stochastic in nature (McAdams and Arkin, 1997; Nachman, 2004). All the events in gene regulation such as transcription, post-transcriptional modification and decay of mRNA are subjected to variability and hence this process cannot be described deterministically. Due to this, statistical models using gene expression data to describe the hidden biological phenomena should take this variability into account.

While conducting microarray experiments, there are many factors that can influence the outcome of the experiment such as hybridisation efficiency of different probes, temperature conditions, amount of sample per probe, sample solution properties. Another major source of noise could be due to samples taken from different cultures. These potential sources of noise should be taken into account before making predictions about the expression patterns of genes.

1.3.2 Chromatin Immunoprecipitation with Microarray

Also known as ChIP-on-chip, it combines the chromatin immunoprecipitation with microarrays to find the interactions between proteins and DNA *in vivo* on a genome-wide scale. Using this technique, experiments can be conducted for an organism to find all the protein-DNA interactions that provide a overall picture of the genome under consideration. Lee et al. (2002) conducted a ChIP-on-chip experiment on yeast to find the regulatory interactions that have been used as the fixed structure of yeast regulatory network in statistical models where such information is required.

A ChIP-on-chip experiment can be divided into two major phases. The first phase starts with *cross-linking* in which a protein of interest (POI) is cross-linked to a DNA sequence. Then the cells are broken down to obtain cross-linked POI-DNA complexes using immuno-

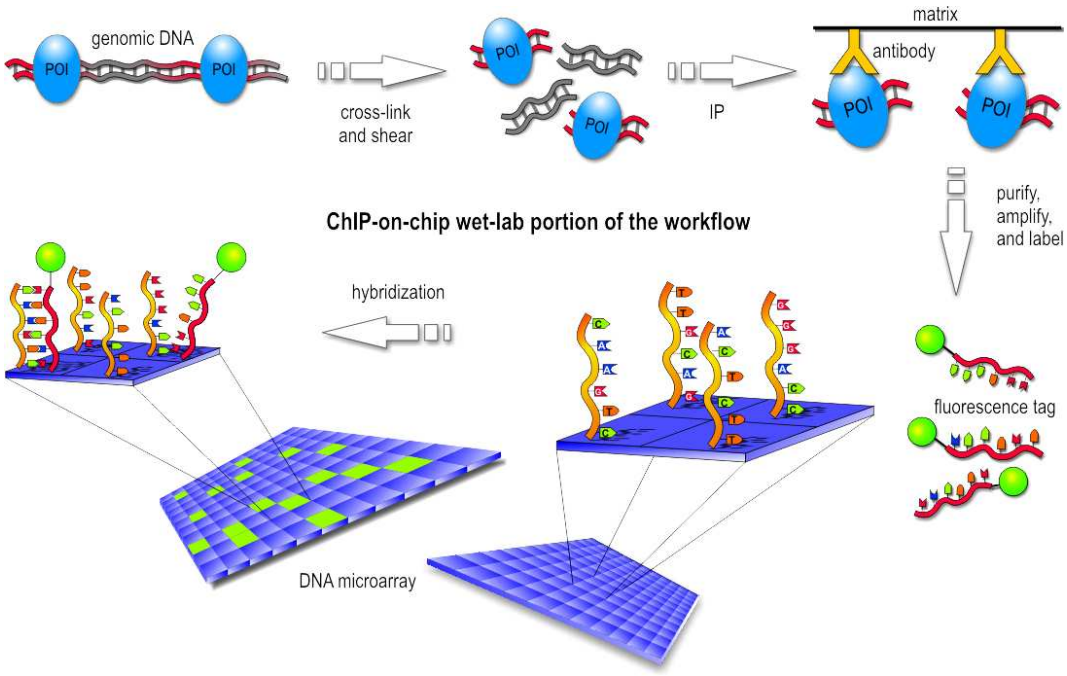


Figure 1.5: Overview of the workflow of ChIP-on-chip (in wet-lab) experiment (Wikipedia, 2012a).

precipitation (IP). After this, the cross-linking of protein-DNA sequences is reversed and the single stranded DNA obtained are labelled with fluorescent tags. The DNA segments are then poured into a microarray for hybridisation to form double stranded DNA fragments. Finally, the microarray is illuminated with fluorescent light and those probes on microarray that are hybridised to labelled segments emit light signals with is captured with the help a camera. This phase is the wet-lab portion ChIP-on-chip experiments and is summarised in figure 1.5. In the second phase, the raw data in the image captured by the camera is then used to obtain numerical values that are used in statistical analysis. This constitute the dry-lab phase of a ChIP-on-chip experiment.

1.4 Inference in Transcriptional Regulation

Inferring the quantitative relationship between TFs and genes within the GRN is an area of intensive research (Lawrence et al., 2010). Most of the methods for this task use gene expression measurements to analyse the operation of GRNs. A major problem with the use of the expression data generated from high-throughput techniques is that the output signal is affected by the modulation of TFs as well as by the intra- and inter-cellular signalling mechanism and many

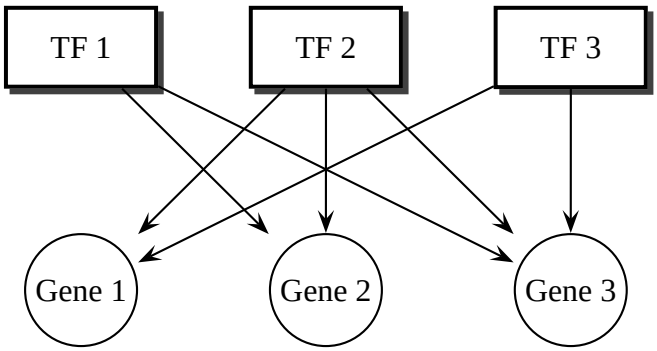


Figure 1.6: A bipartite network of genes and TFs

other cellular processes. Inference of the hidden mechanism governing the regulation of genes only from gene-expression is a challenging task given these interactions. Problems associated when modelling these data are: TF expression is often noisy and low; while post-transcriptional regulation makes the task of modelling more difficult. The task of extracting the structure and dynamics of cellular processes is difficult because of the stochastic nature of the underlying dynamical system involving many hidden factors.

Gene regulatory network can be viewed as a network of proteins and genes where TFs are regulating the production of proteins by controlling the expression rate of their downstream targets (figure 1.6). In this setting, genes and TFs are the nodes of this bipartite network and the edges between the TFs and genes are the regulatory interactions between the nodes of the network. Only the expression measurement of genes are available with a certain degree of noise; the task of modelling is to infer the latent profiles of TFs that are mainly driving the regulation of genes; these TF are in turn under the influence of known experimental/environmental conditions.

1.4.1 Our Approach

Owing to recent advancements in high-throughput techniques (Lee et al., 2002; Boyer et al., 2005; Harbison et al., 2004), a lot of connectivity information is available about GRN, but there is a need to analyse this qualitative connectivity information to generate quantitative network structures. Many statistical techniques are available for gene transcription analysis that are reviewed in detail in chapter 3. We propose to use latent variable models for inferring the relationship among latent TF activities with the observed gene expression measurements. We have used factorial hidden Markov models (FHMM) to model the regulation of gene expression

under the influence of TFs using both linear and non-linear models. The FHMM provides a natural way to model the regulation of genes by multiple TFs as we describe later in this thesis.

The structure of the regulatory network in terms of the interactions between genes and TFs is presumed known in the methods proposed in this thesis. There are two primary methods of obtaining this structural information. One is to determine the architecture of the GRN experimentally by techniques such as ChIP-on-chip that provides a static picture of the regulatory interactions between all the TFs and genes on genome-wide scale. The other source of information about the architecture of GRN is biological literature. Biological databases such as ecocyc (Karp et al., 2002; Keseler et al., 2011) or biocyc (Caspi et al., 2008) provide enormous information about the regulatory interactions so the regulatory network architecture can be compiled from these database. It is important to note that both these sources of network architectural data are known to include false positives and false negatives. Our probabilistic approach towards inference is able to identify the these and therefore provides a means of generating new biological hypotheses.

The methods proposed in this thesis are primarily focused on analysing expression data from time-course microarray experiments. However, we also propose an extension of Sanguinetti et al. (2006) where time-independent version of the model is derived (in chapter 4). The model presented in chapter 5 for combinatorial transcriptional regulation is also derived for non time-series data in appendix A.

One of the highlights of the proposed models in this thesis is the probabilistic nature of the models. The probabilistic approach towards inference provides a principled way to handle the noise in the expression measurements as well as to handle false positives/negatives in the network architecture data. It is also important to associate credibility intervals with the results obtained using gene transcription analysis. As the methods we propose are fully probabilistic in nature, our methods are able to infer confidence measures associated with the inference results.

1.5 Outline of the Thesis

The rest of the thesis is organised as follows:

Chapter 2: This chapter introduces the methodologies that are used throughout this thesis. It starts with a brief introduction to the Bayesian inference framework; then it introduces different classes of latent variables models such as linear dynamical systems and hidden Markov models. Then we describe Bayesian nonparametric methods with focus on Dirichlet process and Dirichlet process mixtures. Finally, we introduce approximate inference techniques such as variational Bayesian inference and Markov chain Monte Carlo (MCMC) sampling.

Chapter 3: This chapter provides a review of prominent statistical inference techniques for transcriptional regulation. We review four of these methods at depth and describe the advantages and disadvantages of these methods.

Chapter 4: This chapter describes a model for inference of TF activities using *state space model* (SSM) and extend it to analyse the expression data with independent experimental condition possibly with replicate. It also discusses a novel, open source and platform independent implementation of this method with an intuitive user interface. The work presented in this chapter is published in *PRIB2009* (Asif and Sanguinetti, 2009) and *Bioinformatics* (Asif et al., 2010) and used for modelling of transcriptional regulation in Rolfe et al. (2011).

Chapter 5: This chapter includes a statistical method for inference of combinatorial interactions of TFs in GRN on genome-wide scale. It describes a novel method based on factorial hidden Markov models to explore the combinatorial nature of transcription regulation. An efficient variational Bayesian expectation maximisation approach is proposed for posterior inference in the model with a detailed analysis on real and simulated data. This work is published in *Bioinformatics* (Asif and Sanguinetti, 2011).

Chapter 6: This chapter introduces an approach for simultaneous inference and clustering of TF profiles from gene expression data. The proposed method infers the latent chains (TF profiles) of the FHMM and also clusters the latents chains using nonparametric mixture modelling. We propose a collapsed Gibbs sampling approach for the nonparametric mixture modelling in this model and perform the detailed analyses of the model using simulated and real datasets.

Chapter 7: This chapter discusses possible future directions for extending the work presented in this thesis.

Chapter 2

Methodologies

This chapter introduces the basic framework for Bayesian inference with an introduction to different classes of Bayesian networks. It then provides a brief introduction to approximate inference using variational inference and MCMC sampling. Towards the end, an introduction to Bayesian nonparametric methods is presented while focusing on nonparametric mixture modelling using Dirichlet process mixture models.

2.1 The Framework of Bayesian Inference

Bayesian inference is a branch of statistics in which all forms of uncertainty about the system under consideration are expressed in terms of probabilities. As an initial step for Bayesian inference, a model is used to characterise the system that closely represents the system that we want to model. This mathematical model contains some unknown parameters that we want to infer. The unknown parameters of the model are treated as *random variables* to account for the uncertainty associated with these parameters. Random variables can be thought of as quantities whose values are not fixed but subject to variations by chance; a *probability distribution* describes the probability of a random variable taking on different values. We use *prior distributions* to reflect our prior belief about the values of these unknown parameters. After seeing the data, the unknown parameters of the model are updated using Bayes' rule to obtain *posterior distributions* for the unknown parameters of the system. The posterior distributions over unknown parameters of the system represent our posterior belief after seeing the observed behaviour of the system.

Bayes' rule defines the logic of uncertainty in the observed behaviour of a system (Jaynes et al., 2003). To understand the Bayes' rule, let us consider an example system that we want to model; the set of unknown parameters of the model for this system are denoted by Θ and

the data generated by this system is denoted by $\mathcal{D}$. We collect our prior knowledge about the unknown parameters of the model in the form of prior distribution for Θ. In most simple form, Bayes' rule is given by

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{normalising factor}} \tag{2.1}$$

or

$$p(\Theta|\mathcal{D}) = \frac{p(\mathcal{D}|\Theta)\, p(\Theta)}{p(\mathcal{D})} \tag{2.2}$$

The above equation can be interpreted as "degree of belief" in Θ before and after observing $\mathcal{D}$. $p(\Theta)$ is the *prior* belief about Θ before observing the data; $p(\mathcal{D}|\Theta)$ represent the *likelihood* function of the observed data; likelihood function represents how probable the data is for a given setting of the parameters Θ. $p(\Theta|\mathcal{D})$ is the *posterior* belief after observing the data. $p(\mathcal{D})$ is the *marginal* probability of data. Our belief about the outcome of the system is subject to the observed behaviour ($\mathcal{D}$) of the system so we define it in terms of *conditional* probabilities. Conditional probabilities reduce the set of possible outcomes based on the condition that some event have already occurred or known to occure *e.g.*, the probability of a certain range of values for the parameters Θ is increased based on the condition that $\mathcal{D}$ is observed; similarly, the probability of a certain range of values for the parameters Θ is decreased after observing $\mathcal{D}$.

An important aspect of Bayesian inference is that the unknown parameters and the observed data are all treated as random random variables. *Hidden* or *latent variables* are random variables that are not observed directly; but they can be inferred from the observed variables with the help of inference. These variables are sometimes referring to physical quantities in the system under consideration such as TF concentrations in the context of GRN which can not be measured for practical reasons. In some other situations these variables refer to an abstract concept such as cluster membership in the context of clustering. The main advantage in using random variables is the reduction in the dimensionality of the data. This is achieved by accumulating many observed variables into one abstract entity that helps to understand the data better. The reduction of dimensionality in case of clustering can be seen in fewer number of clusters compared to the number of observations.

One of the main advantages in using Bayesian inference is the reduced complexity of the model obtained by the use of *marginalisation*. This method automatically prefers simple models that sufficiently explain the observed data without increasing the complexity of the model. This is true even when prior over the unknown parameters are completely uninformative (Tip-

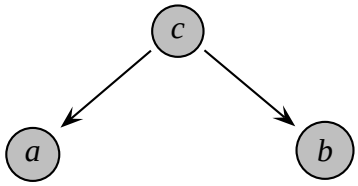


Figure 2.1: Bayesian network for joint distribution $p(a,b,c)$

ping, 2004). However in practice, this approach requires integration over the variables and in complex systems sometimes these computations are analytically intractable. Then, approximation techniques *e.g.,* MCMC sampling and variational approximation are used which are described later in this chapter.

Conditional independence is a widely used concept in Bayesian inference. In case of three random variables *a*, *b* and *c* such that the conditional distribution of *a* is independent of the value of *b* given the value of *c*,

$$p(a|b,c) = p(a|c) \tag{2.3}$$

then *a* is said to be statistically independent of *b* given the value of *c*. The conditional independence can also be derived from the joint distribution of *a* and *b* as follows:

$$\begin{aligned} p(a,b|c) &= p(a|b,c)p(b|c) \\ &= p(a|c)p(b|c) \end{aligned}$$

by using product rule of probabilities with equation (2.3). For two random variables *a* and *b* to conditionally independent of a third variable *c*, one of the above two conditions must be true for all possible of the variable *c*. This independence relationship plays a very important rule in probabilistic modelling. Using conditional independence relation, the structure of the model and the computations needed for inference and learning are simplified to a significant deal.

2.2 Bayesian Networks

Bayesian networks are graphical representation of the conditional independencies between random variables of a model in a form of directed acyclic graph (DAG). Conditional independence in Bayesian networks implies that the random variables (nodes in DAG) are only dependent on its parents and independent of other nodes in DAG given its parent.

The nodes in the Bayesian network are random variables and directed links represent the probabilistic relationships among the nodes of the network. The following joint distribution over random variables *a*, *b* and *c*

$$p(a,b,c) = p(a|c)p(b|c)p(c) \tag{2.4}$$

can be represented as a Bayesian network as shown in figure 2.1. The joint distribution is factored into simpler probability distributions by the application of product rules of probabilities and this factorisation holds for any choice of the joint distribution. The arrows in this figure represent the probabilistic relation between two random variables that can be observed or latent. The node *c* in the graphical model is the *parent* of nodes *a* and *b* as there are directed edges from *a* to *b* and *c*. In general, the joint distribution for a Bayesian network can be written as a product of the individual probability distributions. It can be written as

$$p(\mathbf{x}) = \prod_{v \in V} p(x_v | x_{pa(v)}) \tag{2.5}$$

where $pa(v)$ is the set of parents of node *v* in the graphical model and *x* represents the random variables in the Bayesian network.

2.2.1 Dynamic Bayesian Networks

To model the time dynamics of the sequential data, Bayesian networks are adapted to represent the sequence of variables over time to form dynamic Bayesian network (DBN). In this case, the observed data can not be treated as *independent and identically distributed* (i.i.d), so we need to model the sequence of observation under the assumption that the sequence follows a *Markov process*. Markov process is a stochastic process with *Markov property*; it implies that the conditional probability of the observation at present state only depends on the previous state.

$$p(x_t | x_1, \ldots, x_{t-1}) = p(x_t | x_{t-1})$$

In this case, equation (2.5) becomes

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t | x_{t-1}) \tag{2.6}$$

where *T* is the total number of observations. Equation (2.6) is also known as *first order Markov chain* (figure 2.2).

A simple example of DBN is the HMM which is shown in figure 2.3. The shaded nodes in Bayesian networks are considered observed variables while the other nodes are latent variables.

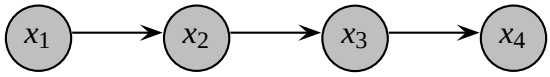


Figure 2.2: A first order Markov chain

2.2.2 Hidden Markov Models

Let $\mathbf{z}_t$ be a latent variable for each of the observations $\mathbf{x}_t$ in a sequence of T observations in equation (2.6) where $\mathbf{z}_t$ can have different dimensionality than $\mathbf{x}_t$. If we move the Markov property assumption to the latent variables $\mathbf{z}_t$ instead of $\mathbf{x}_t$, then the resultant graphical representation can be shown as figure 2.3. Based on the discrete or continuous choices for latent variable $\mathbf{z}_t$, we can get two different types of models. If both the latent and observed variables are Gaussians with a linear dependence of the conditional distributions on their parent nodes then we get *linear dynamical systems*; whereas if the latent variables are discrete then we obtain *hidden Markov models* (HMM). The general class of these models is called *state space model* (SSM).

In a HMM, latent variable $\mathbf{z}_t$ is a multinomial random variable that describes which state of the latent variable is responsible for generating observation $\mathbf{x}_t$. These variables can be thought of as K dimensional vectors where only one entry of the vector is non-zero (1-of-K representation). The joint probability of a HMM can be written as

$$p(\mathbf{X},\mathbf{Z}) = p(\mathbf{z}_1)\left[\prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{z}_{t-1})\right].\prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{z}_t) \tag{2.7}$$

where $\mathbf{X} = \{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_T\}$ and $\mathbf{Z} = \{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_T\}$. In the HMM jargon, $p(z_t|z_{t-1})$ is called *transition probability* or *transition rate* while $p(x_t|z_t)$ is called *emission probability*. The initial transition probability at $t = 1$ has a special meaning; it specifies the initial value of latent

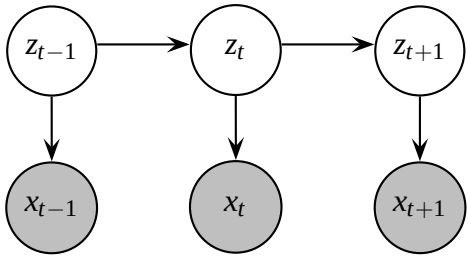


Figure 2.3: A hidden Markov model

variable z_1 and is usually denoted by π. The transition probabilities are usually denoted by $\mathbf{A}$ with $K(K-1)$ independent parameters encoding the probabilities

$$A_{jk} = p(z_{t,k} = 1 | z_{t-1,j} = 1); \quad 0 \leq A_{jk} \leq 1, \sum_k A_{jk} = 1 \tag{2.8}$$

The emission probability vector, $p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{B})$, with $\mathbf{B}$ as the parameter for this distribution consists of K values corresponding to K possible states of the latent variable $\mathbf{z}_t$. Now the joint probability of all the variables can be specified as

$$p(\mathbf{X}, \mathbf{Z}|\Theta) = p(\mathbf{z}_1|\pi) \left[\prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{A}) \right] . \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{B}) \tag{2.9}$$

where $\Theta = \{\mathbf{A}, \mathbf{B}, \pi\}$ is the set of model parameters. The basic HMM has been extended to various different forms (Rabiner, 1989; Bishop, 2006). One variant of HMM is the factorial hidden Markov model (FHMM) in which the latent state representation is distributed to multiple state variables; the observed sequence is then conditioned on a set of K independent Markov chains instead of a single Markov chain. The FHMM provides a natural way to model the regulation of genes in GRNs as we describe in chapter 5 and 6.

2.2.2.1 Forward Backward Algorithm

An important problem of a HMM given its parameters is that of finding the posterior marginal probabilities of hidden states $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T$ given an observed sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$. This *training* of a HMM is achieved by *forward backward algorithm*.

In forward backward algorithm, $\alpha_t(i)$ denotes the probability of partial observation sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t$ until time t and the state of latent variable $z_t = i$ at time t given the parameters Θ; whereas $\beta_t(i)$ denotes the probability of partial observation sequence $\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \ldots, \mathbf{x}_T$ given the state of latent variable $z_t = i$ at time t and the parameters Θ

$$\begin{aligned} \alpha_t(i) &= p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, \mathbf{z}_t = i|\Theta) \\ \beta_t(i) &= p(\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \ldots, \mathbf{x}_T, |\mathbf{z}_t = i, \Theta). \end{aligned}$$

where $\alpha_t(i)$ and $\beta_t(i)$ are called forward and backward variables respectively. The algorithm then computes the forward probabilities for all the time slices and states of the latent variable $\mathbf{z}_t$ as follows,

$$\alpha_1(j) = \pi_k p(\mathbf{x}_1|\mathbf{z}_1 = j) \qquad 1 \leq j \leq K \tag{2.10}$$

$$\alpha_{t+1}(j) = \left[\sum_{k=1}^{K} \alpha_t(k) A_{jk} \right] p(\mathbf{x}_{t+1}|\mathbf{z}_{t+1} = j) \qquad 1 \leq j \leq K, 1 \leq t \leq T-1. \tag{2.11}$$

For backward probabilities

$$\beta_T(j) = 1, \qquad 1 \leq j \leq K \tag{2.12}$$

$$\beta_t(j) = \sum_{k=1}^{K} A_{jk} p(\mathbf{x}_{t+1}|\mathbf{z}_{t+1} = j)\beta_{t+1}(k) \qquad t = T-1, T-2, \ldots, 1 \qquad 1 \leq j \leq K. \tag{2.13}$$

Having computed these probabilities, the task of finding the posterior marginal probabilities can be achieved by defining

$$\gamma_t(k) = \frac{\alpha_t(k)\beta_t(k)}{\sum_{k=1}^{K} \alpha_t(k)\beta_t(k)} \qquad t = 1, 2 \ldots, T,\ 1 \leq k \leq K. \tag{2.14}$$

Equation (2.14) specifies the probability of being in state k at time t given $\mathbf{X}$ and Θ. These probabilities also called *marginal state probabilities*. With long observation sequences, the forward backward algorithm needs to compute extremely small conditional probabilities that sometimes can result in arithmetic underflow. This situation may also arise if multiple observed sequences *e.g.,* multiple gene expression profiles are used to estimate the posterior marginal probabilities.

The solution to numerical instability of forward backward algorithm is to use *log space* for calculating the conditional probabilities of equations (2.11)-(2.14) (Mann, 2006). Another approach to circumvent this problem is to rescale these conditional probabilities by using a scaling factor that keeps these probabilities within the range of standard floating point arithmetics (Rabiner, 1989).

As the number of genes in the analysis we perform are in the order of hundreds or sometimes thousands, we also face this numerical instability problems due to the multiplication of large number of small emission probabilites. We use log space for the calculations of forward backward algorithm with gene expression profiles as observed sequences to avoid numerical instabilities.

2.2.3 Linear Dynamical Systems

Figure 2.3 shows the general class of models where sequence of latent variables are used to model the sequential data. Lets assume that the latent variables are now continuous. In this case, each pair of node $\{\mathbf{x}_t, \mathbf{z}_t\}$ represents a linear-Gaussian latent variable model. This implies that the joint distribution, conditional distributions and marginal distributions all will be Gaussians. So we can write the transition and emission probabilities as

$$p(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t|\mathbf{A}\mathbf{z}_{t-1}, \Gamma)$$

$$p(\mathbf{x}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{x}_t|\mathbf{C}\mathbf{z}_t, \Sigma)$$

or equivalently in the form of linear equations as

$$\begin{aligned} \mathbf{z}_t &= \mathbf{A}\mathbf{z}_{t-1} + \mathbf{w}_t \\ \mathbf{x}_t &= \mathbf{C}\mathbf{z}_t + \mathbf{v}_t \end{aligned} \tag{2.15}$$

with noise terms given by

$$\begin{aligned} \mathbf{w} &\sim \mathcal{N}(w|0,\Gamma) \\ \mathbf{v} &\sim \mathcal{N}(v|0,\Sigma) \end{aligned}$$

where $\Theta = \{\mathbf{A},\Gamma,\mathbf{C},\Sigma\}$ are called the parameters of the linear dynamical systems (LDS) and can be determined using maximum likelihood through expectation maximisation algorithm. In chapter 4, we derive the inference algorithm for LDS using an approximate inference technique where approximate inference is used due to the intractability of the posterior distribution. Note that special attention needs to paid for the distributions of the first sample in the sequence as in case of HMM.

2.3 Expectation Maximisation Algorithm

Expectation maximisation (EM) algorithm is a general technique for finding the maximum likelihood estimates for model with latent variables (Dempster et al., 1977; McLachlan and Krishnan, 2008). It computes the expected values of the latent variables and parameters of the model iteratively in two steps: the *expectation* or E step and *maximisation* or M step.

Let $\mathbf{X}$ denote the set of observations with each row containing one observation. Similarly, $\mathbf{Z}$ denote the set of latent variables with one row for each observation with 1-of-K encoding. If Θ denote the set of model parameters, then the log of the marginal likelihood of the data is given by

$$\ln p(\mathbf{X}|\Theta) = \ln \left(\sum_{\mathbf{Z}} p(\mathbf{X},\mathbf{Z}) \right) \tag{2.16}$$

where the summation replaces the integration if the latent variables, $\mathbf{Z}$, are continuous variables. However, this equation leaves us with one problem; the summation in this equation appears inside the logarithm which results in complicated expression when estimating the maximum likelihood solutions. The solution to this problem is to consider the *complete* data which includes $\{\mathbf{X},\mathbf{Z}\}$ instead of just $\mathbf{X}$.

Most of times, we do not know the values for latent variables $\mathbf{Z}$ but we can calculate posterior probability for $\mathbf{Z}$ given observations ($\mathbf{X}$, which we call *incomplete* data) and Θ, $p(\mathbf{Z}|\mathbf{X},\Theta)$.

Algorithm 1 Expectation Maximisation Algorithm

1: Initialise $\Theta^{old} = \Theta^0$

2: **repeat**

3: E step: Evaluate $p(\mathbf{Z}|\mathbf{X}, \Theta^{old})$

4: M step: Evaluate Θ^{new} as

$$\Theta^{new} = \arg\max_{\Theta} Q(\Theta, \Theta^{old})$$

where

$$Q(\Theta, \Theta^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \Theta^{old}) \ln p(\mathbf{X}, \mathbf{Z}|\Theta)$$

5: $\Theta^{old} = \Theta^{new}$

6: **until** convergence criterion is not satisfied.

At the start, the EM algorithm initialises the model parameters Θ by choosing some starting values Θ^0. Then it repeat the following two steps:

E step: During the E step, the current values of the parameter Θ^{old} are used to find the posterior distribution of the latent variables $\mathbf{Z}$. Having computed this, we can use this posterior probability distribution to compute the expectation of the log likelihood of complete data evaluated for some general parameter value Θ as

$$Q(\Theta, \Theta^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \Theta^{old}) \ln p(\mathbf{X}, \mathbf{Z}|\Theta)$$

Note that the logarithm directly acts on the joint distribution $p(\mathbf{X}, \mathbf{Z})$ in this case.

M step: In the M step, we maximise our estimates of the parameters Θ to obtain Θ^{new} as

$$\Theta^{new} = \arg\max_{\Theta} Q(\Theta, \Theta^{old}) \tag{2.17}$$

After one iteration of the EM algorithm we get the revised values for Θ which are then used in the next iteration as Θ^{old}; Θ^{old} is also used to compute the posterior distribution, $p(\mathbf{Z}|\mathbf{X}, \Theta)$ in the next iteration of the EM algorithm. This posterior distribution is used to compute the expectation of the log likelihood of the complete data. The convergence of the algorithm can be monitored based on the increase in the expectation of the log likelihood; the algorithm iterates until the increase is less than a predefined threshold.

2.4 Approximate Inference

Fundamental task of probabilistic modelling is the estimation of the posterior distribution of the latent variables $\mathbf{Z}$ given the observed data $\mathbf{X}$ *i.e.,* $p(\mathbf{Z}|\mathbf{X})$ and expectations with respect to these distributions. In a fully Bayesian approach, all the unknown parameters are given prior distributions and treated as latent variables $\mathbf{Z}$. Then, using the EM algorithm, we can compute the expectations of the log likelihood of complete data *w.r.t.* the posterior distributions of the latent variables (Dempster et al., 1977). In many practical applications, this is not feasible due to various reasons such as dimensionality of latent variable space or the form of the posterior distributions. For these modelling problems, approximation techniques are used; these techniques can be categorised as *stochastic* or *deterministic*. Variational inference falls under the category of deterministic approximation techniques (Jordan et al., 1999; Bishop, 2006). Variational methods are used for finding an approximate solution by restricting the range of functions over which the approximation is applied. This restriction may also be in the form of factorisation in case of the factorized variational approach as we describe later. Markov chain Monte Carlo (MCMC) techniques fall under the category of stochastic approximation techniques. We will briefly describe these two approximations next.

2.4.1 Variational Bayesian Inference

Variational Bayesian inference is an approximation technique based on the calculus of variations. The basic idea in variational inference is to approximate the posterior distribution over the latent variables and parameters with a simpler distribution. Variational techniques convert a complex problem into a simpler problem by making use of the decoupling of the degree of freedom in the original problem (Jordan et al., 1999). This decoupling is obtained by expanding the problem to include additional parameters also known as *variational* parameters that are optimised according to the problem under consideration.

In a fully Bayesian framework, a model with a set of latent variables $\mathbf{Z}$ and a set of observed variables $\mathbf{X}$ with joint distribution $p(\mathbf{X},\mathbf{Z})$, our goal is to find an approximate posterior distribution for $p(\mathbf{Z}|\mathbf{X})$ and $p(\mathbf{X})$. Decomposing the log marginal probability, we get

$$\ln p(\mathbf{X}) = \mathcal{L}(\mathbf{q}) \; + \; \mathbf{KL}(\mathbf{q} \parallel \mathbf{p}) \tag{2.18}$$

where:

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left[\frac{p(\mathbf{X},\mathbf{Z})}{q(\mathbf{Z})} \right] d\mathbf{Z} \tag{2.19}$$

$$\mathrm{KL}(q \parallel p) = -\int q(\mathbf{Z}) \ln \left[\frac{p(\mathbf{Z} \mid \mathbf{X})}{q(\mathbf{Z})} \right] d\mathbf{Z} \tag{2.20}$$

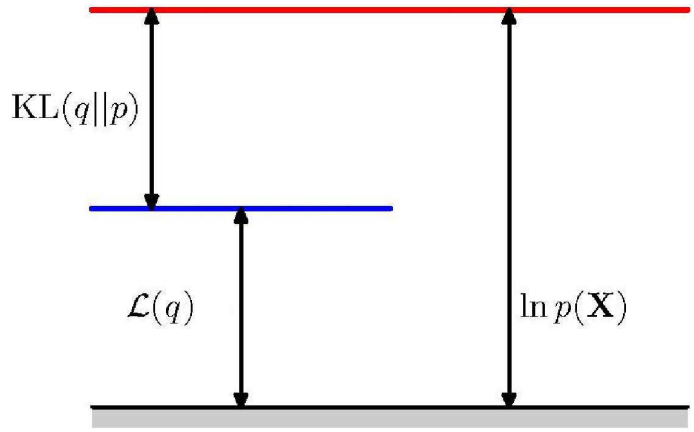


Figure 2.4: Illustration of the decomposition given in equation (2.18) which holds for any choice of distribution $q(\mathbf{Z})$ (image taken from Bishop (2006))

Here, $\mathcal{L}(q)$ is a functional and equation (2.20) characterises the Kullback-Leibler divergence between approximating distribution $q(\mathbf{Z})$ and the posterior distribution $p(\mathbf{Z}|\mathbf{X})$. Equation (2.19) and (2.20) differ in sign and $\mathcal{L}(q)$ have joint distribution of $\mathbf{X}$ and $\mathbf{Z}$ while $\mathrm{KL}(q \parallel p)$ contains conditional distribution of $\mathbf{Z}$ given $\mathbf{X}$. Using the product rule

$$\ln p(\mathbf{Z},\mathbf{X}) = \ln p(\mathbf{Z}|\mathbf{X}) + \ln p(\mathbf{X}) \tag{2.21}$$

in equation (2.19) and substituting this value in equation (2.18) gives the required log likelihood given in equation (2.18) which proves the basis for this decomposition.

Note that KL divergence is always positive or zero. If KL divergence is zero, then approximating distribution $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X})$. Therefore, looking at equation (2.18), it follows that $\mathcal{L}(q)$ is a lower-bound on $\ln p(\mathbf{X})$ i.e. $\mathcal{L}(q) \leq \ln p(\mathbf{X})$. Figure 2.4 shows the decomposition shown in equation (2.18).

We can minimise the KL divergence by maximising the lower bound specified in the equation (2.18) using optimisation *w.r.t.* the distribution $q(\mathbf{Z})$. The KL divergence vanishes when the $q(\mathbf{Z})$ is equal to $p(\mathbf{Z}|\mathbf{X})$. However, in many cases, it is difficult to work with the form of true posterior distribution. So, we restrict family of distributions $q(\mathbf{Z})$ that can be used; a member of this family for which the KL divergence is minimised is selected as the approximating posterior distribution. The goal here is usually to restrict the family of distributions by choosing a flexible distribution that can best approximate the true posterior distribution. The restriction imposed is usually for the purpose of tractability. Standard nonlinear optimisation techniques can then be used to obtain the optimal values of the parameters. One approach for restricting the family of distributions is to use factorised distributions for approximating the posterior distributions which is discussed next.

2.4.1.1 Factorized Variational Approach

One way to restrict the family of approximating distributions is to factorize the distribution. In this approach, the set of latent variables $\mathbf{Z}$ is partitioned into disjoint groups as follows

$$q(\mathbf{Z}) = \prod_{i=1}^{M} q_i(\mathbf{Z}_i); \tag{2.22}$$

assuming that the distribution q factorizes with respect to these groups. The objective is to select a distribution for which the lower bound $\mathcal{L}(q)$ is largest. To achieve this, $\mathcal{L}(q)$ is to be optimised *w.r.t* all the distributions $q_i(\mathbf{Z}_i)$; this is done by (variational) optimisation of $\mathcal{L}(q)$ *w.r.t.* each of the factors given in equation (2.22). For this purpose, substituting equation (2.22) in equation (2.19) and simplifying we obtain

$$\begin{aligned}\mathcal{L}(q) &= \int \prod_i q_i(\mathbf{Z}_i) \left[\log p(\mathbf{X},\mathbf{Z}) - \sum_i \log q_i(\mathbf{Z}_i) \right] d\mathbf{Z} \\ &= \int q_j(\mathbf{Z}_j) \mathbb{E}_{i\neq j}[\log p(\mathbf{X},\mathbf{Z})] d\mathbf{Z}_j - \int q_j(\mathbf{Z}_j) \log q_j(\mathbf{Z}_j) d\mathbf{Z}_j + \text{const}\end{aligned} \tag{2.23}$$

where

$$\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})] = \int \log p(\mathbf{X},\mathbf{Z}) \prod_{i\neq j} q_i(\mathbf{Z}_i) d\mathbf{Z}_i \tag{2.24}$$

and all the terms that do not depend on $q_j(\mathbf{Z}_j)$ are absorbed into the constant. After this, $q_{i\neq j}$ is kept fixed and $\mathcal{L}(q)$ in equation (2.23) is maximised with respect to all possible forms for the distribution $q_j(\mathbf{Z}_j)$. Another important fact is that equation (2.23) is negative KL divergence and thus maximising the equation (2.23) is equivalent to minimising KL divergence and the minimum occurs when $q_j(\mathbf{Z}_j) \propto \exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})])$. The general expression for the optimum solution is given by

$$\log \hat{q_j}(\mathbf{Z}_j) = \mathbb{E}_{i\neq j}[\log p(\mathbf{X},\mathbf{Z})] + const \tag{2.25}$$

This above framework provides the basis for variational methods. The last equation says that the log of the solution for q_j is obtained by taking the expectation of the log of the joint distribution over hidden and observed variables with respect to all other factors q_i with $i \neq j$. We can write the above equation as

$$\hat{q}_j(\mathbf{Z}_j) = \frac{\exp(\mathbb{E}_{i\neq j}[\log p(\mathbf{X},\mathbf{Z})])}{\int \exp(\mathbb{E}_{i\neq j}[\log p(\mathbf{X},\mathbf{Z})]) d\mathbf{Z}_j} \tag{2.26}$$

Equation (2.26), for $j = 1,\ldots,M$ where M is the total number of factors, represent a set of consistency conditions for the maximum of the lower bound. It is important to emphasise that this equation does not represent an explicit solution as the expression on the right-hand

side of equation (2.26) for the optimum factor $\hat{q_j}(\mathbf{Z}_j)$ depends on the expectations computed with respect to other factor $q_i(\mathbf{Z}_j)$ with $i \neq j$. So, the solution to this can be computed by first initialising all the factors $q_i(\mathbf{Z}_i)$ and then calculating factors in a cyclic order with revised estimates given by equation (2.26) until the convergence is achieved.

In general, the factorisation approach of variational inference usually underestimates the variance of the approximate distribution to the posterior distribution (Bishop, 2006). The estimation of the factorized approximating distributions may provide us with functional forms which are still intractable; therefore usually some simpler space for posterior distributions of the parameters is used (Beal, 2003). One advantage of variational inference approach is that any factorisation of the posterior distribution gives a lower bound on the marginal likelihood.

2.4.2 Sampling Techniques

In Bayesian inference, computation of the posterior distribution is usually intractable and we have to resort to some approximation technique like one described in the section 2.4.1. This section introduces another class of approximation techniques based on numerical sampling known as *Monte Carlo* techniques. In most inference problems, we are only interested in evaluating the expectations rather than the posterior distribution itself. In these situations, we can use sampling techniques to find the expectations of some function $f(z)$ *w.r.t.* a distribution $p(z)$. In case of discrete variables, expectation is computed as

$$\mathbb{E}[f] = \sum_i f(i)p(i) \tag{2.27}$$

In general, sampling techniques allow us to obtain a set of samples $z^{(i)}$ where $i = 1, \ldots, N$ drawn independently from the distribution $p(z)$. Then the expectation can be found as

$$\hat{f} = \frac{1}{N} \sum_{i=1}^{N} f(z^{(i)}) \tag{2.28}$$

Different sampling techniques are available for different types of graphical models. We will only briefly describe Gibbs sampling in the next section. The inference in the model proposed in chapter 6 is done via Gibbs sampling.

2.4.2.1 Gibbs Sampling

Gibbs sampling also known as *alternating conditional sampling* is defined in terms of subvectors of the parameter vector. In one trace, Gibbs sampler cycles through the subvector of the parameters and draws samples for each subset conditional on all other subsets. In each iteration

of the Gibbs sampler, k steps are required to draw samples from all the subvector of the parameter vector where k is the number of sub-vectors of the parameter vector (Gelman et al., 2004). More precisely, if $\mathbf{Z}$ denotes the parameter vector and z_j^t denotes the values of the subvector z_j at iteration/time t, then each z_j^t is drawn from the conditional distribution given all other subvectors as

$$p(z_j|z_{-j}^{t-1}) \tag{2.29}$$

where z_{-j}^{t-1} is given by

$$z_{-j}^{t-1} = (z_1^t, \ldots, z_{j-1}^t, z_{j+1}^{t-1}, \ldots, z_k^{t-1}) \tag{2.30}$$

In many cases, it is possible to sample directly from most of the conditional posterior distributions of the parameters and use of conjugate priors also provide ease in sampling.

2.5 Bayesian Nonparametric Methods

The models described in the previous sections are parameterised with a limited number of parameters. It is often desirable, for theoretical reason, to build models that have no limitation on the parameter space. These methods, called nonparametric Bayesian methods, define distribution of function space such as that of probability measures to avoid restrictive parametric assumptions (Müller and Quintana, 2004). The prior distribution for these nonparametric methods must also be a nonparametric distribution with infinite number of parameters. Nonparametric methods provide an efficient way to analyse the data where the number of latent components are not known in advance. In the following, we discuss one of these methods and then describe its use in nonparametric mixture modelling.

2.5.1 Dirichlet Process

Dirichlet process (DP) is a stochastic process that is widely used in Bayesian nonparametric modelling. A sample from a Dirichlet process is a discrete probability distribution that cannot be described by using a finite number of parameters. A DP can be thought of as a generalisation of the Dirichlet distribution (Holmes, 2010; Gibbons and Chakraborti, 2003).

Let G be a distribution over a space $\mathbb{X}$ and η be a (real) positive number. For any finite set of partitions of $\mathbb{X}$, $A_1 \cup A_2 \cup A_k = \mathbb{X}$, the vector $G(A_1), \ldots, G(A_k)$ is a random measure. $G \sim DP(G_0, \eta)$ with base measure G_0 and concentration parameter η if

$$G(A_1), \ldots, G(A_k) \sim \text{Dir}(\eta G(A_1), \ldots, \eta G(A_k)) \tag{2.31}$$

for any measurable finite partitions of $\mathbb{X}$.

A DP can also be viewed as a distribution over distributions with two parameters. Base distribution G_0 can be thought as the mean of the DP because $\mathbb{E}[G(A)] = G_0(A)$. The concentration parameter η can be interpreted as the inverse variance of the DP because $\mathbb{V}[G(A)] = \frac{G_0(A)(1-G_0(A))}{\eta+1}$ which implies that larger values of the concentration parameter will force DP to concentrate more of its mass around its mean.

Based on different construction schemes, DPs can be represented in different ways (Teh et al., 2006; Teh, 2007). Here we describe one method which is known as the *stick breaking* construction.

2.5.1.1 Stick-breaking Construction

The process for *stick breaking* construction of DP can be described as follows:

$$\beta_k \sim \text{Beta}(1,\eta) \tag{2.32}$$

$$\pi = \beta_k \prod_{j=1}^{k-1}(1-\beta_j) \quad for k = 1,2,\ldots \tag{2.33}$$

This process can be interpreted by considering a unit length stick and then breaking it according to the proportion $\pi_1 = \beta_1 \sim \text{Beta}(1,\eta)$; then the remaining stick broken according to the proportions $\beta_k \sim \text{Beta}(1,\eta)$ with the remaining proportion of the stick assigned to π_k. Collectively, this construction of DP is called *GEM* distribution (named after Griffiths, Engen, and McCloskey, (Gnedin and Kerov, 2001)).

$$\pi \sim \text{GEM}(1,\eta) \tag{2.34}$$

2.5.2 Dirichlet Process Mixture Modelling

Dirichlet process mixture model (DPMM) is an extension of finite mixture models where the number of latent components are not known *a priori*. It is easier to understand the DPMM by starting from finite mixture models. A graphical representation for finite mixture model is shown in figure 2.5 where *i* and *k* are the indices for observations and clusters respectively. The generative mechanism for finite mixtures is given by

$$\begin{aligned}
\pi &\sim \text{Dirichlet}(\eta_1,\eta_2,\ldots,\eta_K) \\
z_i|\pi &\sim \text{Multinomial}(\pi) \\
\theta_k|\lambda &\sim G(\lambda) \\
x_i|z_i,\{\theta\}_{k=1}^K &\sim F(\theta_{z_i})
\end{aligned}$$

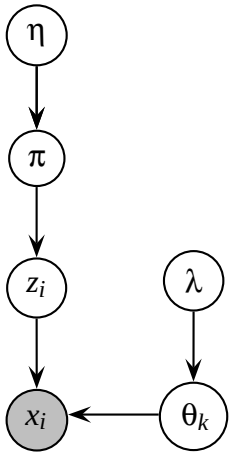


Figure 2.5: Graphical representation for finite mixture models ($i = 1,\ldots,N, k = 1,\ldots,K$). If $K \to \infty$ then it forms the graphical model for infinite mixture models (DPMM).

The above generative mechanism generates a data point x_i by selecting one of K components from a multinomial distribution; the prior distribution for this multinomial distribution is a Dirichlet distribution parameterised by η which can be taken to be uniform with $\eta/K,\ldots,\eta/K$. After selecting a component, a sample θ_k is drawn from the component distribution G to generate the data point x_i from the distribution F. For mathematical convenience, the distributions F and G are from exponential family of distributions with G as conjugate prior for F (Bishop, 2006).

In finite mixture models, the value for K is known in advance; however, this is not the case for *infinite* mixture models such as DPMM. If we change the limit of K to infinity, then the above described generative process becomes a DPMM. It is given by

$$\begin{aligned} \pi &\sim \text{GEM}(1,\eta) \\ z_i|\pi &\sim \text{Multinomial}(\pi) \\ \theta_k|\lambda &\sim G(\lambda) \\ x_i|z_i,\{\theta\}_{k=1}^{\infty} &\sim F(\theta_{z_i}) \end{aligned}$$

In case of a simple infinite mixture of Gaussians with fixed variance, G becomes the conjugate prior for the mean of the Gaussians while F is Normal distribution with mean given by G.

2.5.3 Collapsed Gibbs Sampling for DPMM

One advantage of using conjugate prior is that we can often integrate out the hyperparameters of the prior distribution; this helps to a great deal while sampling for posterior analysis. We can easily derive a Gibbs sampling scheme for DPMM if we *collapse* the Gibbs sampler by integrating out the component parameters θ_{z_i}. By doing this, we only need to take samples for z_is. The collapsed approach to sampling is justified by Rao-Blackwell theorem (Blackwell, 1947); according to this theorem, integrating out some parameters from the conditional distributions of a variable reduces the variance of the posterior estimate of that variable.

Let $F(x_i|\theta_k)$ belongs to the exponential family with $G(\theta_k|\lambda)$ as conjugate prior in the standard DPMM setting as described in section 2.5.2. The conditional posterior distribution for component indicator variable z_i, $p(z_i = k|\mathbf{z}_{-i}, \mathbf{x}, \pi, \{\theta\}_{k=1}^{K}, \eta, \lambda)$, is conditionally dependent on π and θ_k so sampling from this infinite dimensional distribution is not possible for practical reasons; here $\mathbf{z}_{-i}$ denote all other component indicators except i^{th} component. However, if we integrate out π and $\{\theta\}_{k=1}^{K}$ then it is easy to sample from the resulting conditional posterior distribution. We can write the conditional posterior distribution of components indicator variables as

$$\begin{aligned} p(z_i = k|\mathbf{z}_{-i}, \mathbf{x}, \eta, \lambda) &= p(z_i = k|x_i, \mathbf{z}_{-i}, \mathbf{x}_{-i}, \eta, \lambda) \\ &\propto p(z_i = k|\mathbf{z}_{-i}, \mathbf{x}_{-i}, \eta, \lambda) p(x_i|z_i = k, \mathbf{z}_{-i}, \mathbf{x}_{-i}, \eta, \lambda) \qquad (2.35) \\ &= p(z_i = k|\mathbf{z}_{-i}, \eta) p(x_i|z_i = k, \lambda) \qquad (2.36) \end{aligned}$$

where we have used Bayes' rule in equation (2.35) and conditional independence property of Bayesian networks in equation (2.36) (fig. 2.5). The first term in equation (2.36) can be termed as *predictive prior*; using the standard results of mixture models, it is given by

$$p(z_i = k|\mathbf{z}_{-i}, \eta) = \frac{n_{k,-i} + \eta/K}{n + \eta - 1} \qquad (2.37)$$

where $n_{k,-i}$ is the number of data items currently assigned to component k excluding the i^{th} item. The second term in equation (2.36) can be termed as *predictive likelihood*. It can be obtained as

$$p(x_i|\mathbf{x}_{k,-i}, \lambda) = \int p(x_i|\theta_k) p(\theta_k|\mathbf{x}_{k,-i}, \lambda) d\theta_k \qquad (2.38)$$

using the standard results of exponential family of distributions. For a nonparametric mixture of Gaussian with unknown mean μ_k and unit variance, the generative process for DPMM can

be written as

$$\begin{aligned} \pi &\sim \text{GEM}(1,\eta) \\ z_i|\pi &\sim \pi \\ \mu_k|\lambda &\sim \mathcal{N}(0,1) \\ x_i|z_i,\{\theta\}_{k=1}^{\infty} &\sim \mathcal{N}(x_i|\mu_k,1) \end{aligned}$$

where the conjugate prior for μ_k is taken to be $\mathcal{N}(0,1)$. In this case, the predictive likelihood comes out to be

$$p(x_i|\mathbf{x}_{k,-i},\lambda) = \frac{1}{2\sqrt{\pi}}\exp(-\frac{x_i^2}{4}) \tag{2.39}$$

where λ is the set of hyperparameters of a standard Normal distribution. The conditional posterior distribution for component indicator variables is given by

$$p(z_i = k|\mathbf{z}_{-i},\mathbf{x},\eta,\lambda) = \frac{n_{k,-i}+\eta/K}{n+\eta-1}\left[\frac{1}{2\sqrt{\pi}}\exp(-\frac{x_i^2}{4})\right];$$

so the Gibbs sampler in this case would iteratively update the component indicator variables for each of the observation using the updated component assignment for all other observations until the sampler is deemed to have converged. The sampling scheme for Dirichlet process mixture of Gaussians is summarised in algorithm 2.

2.6 Conclusion

In this chapter, we presented a brief introduction to the methodologies used in the rest of the thesis. Starting with the basic framework of Bayesian inference, we describe different classes of models that can be obtained from dynamic Bayesian networks by changing the type of latent variables. Although we do not directly use HMM in later chapters of the thesis, but we use a variant of the HMM (factorial hidden Markov model) and the inference mechanism in that model remains largely same. Then we introduce variational approximation and sampling techniques that are used for approximate inference in the models presented in this thesis. Finally, nonparametric Bayesian methods are introduced with focus on Dirichlet process mixture models that we use in chapter 6.

Algorithm 2 Gibbs sampling algorithm for DPMM

Require: $\{z_i^{t-1}\}_{i=1}^n$

Sample new $\{z_i^{t-1}\}_{i=1}^n$ in the following way:

repeat

  for $i \leftarrow 1, n$ **do**

    Remove the data item x_i given the cluster assignment z_i.

    If the cluster becomes empty, then delete this cluster and rearrange the cluster indices.

    Compute the predictive likelihood for each of K clusters (equation 2.39).

    for $k \leftarrow 1, K+1$ **do**

      Draw a sample for new z_i from

$$
\begin{aligned}
p(z_i = k, k \leq K) &\propto \frac{n_{k,-i}}{n+\eta-1}\left[\frac{1}{2\sqrt{\pi}}\exp\left(-\frac{x_i^2}{4}\right)\right] \\
p(z_i = K+1) &\propto \frac{\eta}{n+\eta-1}\left[\frac{1}{2\sqrt{\pi}}\exp\left(-\frac{x_i^2}{4}\right)\right]
\end{aligned}
$$

    end for

    If $z_i = K+1$, then instantiate a new cluster with index $K+1$.

    Update $\{n_k\}_{k=1}^K$.

  end for

until Gibbs sampler is deemed to have converged.

Chapter 3

Inference Methods for Transcriptional Regulation

3.1 Introduction

The regulation of transcription is one of the most complex processes in living organisms. Being fundamental to all biological systems, it plays a major role in governing repairs, reproduction, respiration and various other biological processes necessary for the survival of cells. The regulation of transcription determines the changes in the expression level of target genes by altering the transcription rates in the regulatory network. These transcription rates are controlled by DNA binding proteins or TFs to control transcription of genes. The expression of genes is a basic information processing mechanism whereby information stored in genes in the form of DNA is transcribed to mRNA. While the mRNA produced during transcription in prokaryotes is in ready for further processing (*i.e.,* translation), the mRNA produced in eukaryotes has to undergo further modifications to become mature mRNA.

The process of gene expression consists of several phases such as transcriptional regulation, post-transcriptional modifications, translation and post-translational modifications to produce functional gene products which are mRNA or proteins. The levels of mRNA after the transcription stage can be measured quantitatively and is usually referred to as gene expression levels. These expression levels reveal how active the genes are and any abnormality in these expression patterns indicates functional changes in the cellular behaviour.

Gene expression data is widely used as a source to reconstruct the hidden regulatory activities in the regulatory network. In order to understand the internal dynamics of the regulatory network in a quantitative manner, knowledge about the concentration of TF proteins and their downstream targets is required for all the samples in a biological experiment. While it is easy to obtain the expression measurements of genes, it is hardly possible for TFs due to various

reasons such as low concentrations of TFs, post-transcriptional modifications and rapid transition behaviour (Ptashne and Gann, 2002). Apart from this, it is known that TF interactions with genes is highly influenced by the environmental signals (Harbison et al., 2004); these reasons make the experimental measurements of TFs difficult. However, it is possible to experimentally determine the structure of the regulatory network using Chromatin Immunoprecipitation with microarray (ChIP-on-chip (Lee et al., 2002)); this information is usually helpful for statistical inference of the missing quantities in the regulatory network. The information about the structure of the regulatory network or *connectivity* reveals which TFs are responsible for regulating which genes. Although the connectivity information provides a useful insight of the regulatory network, it is prone to contain noise in the measurements in the form of false positives/negatives. So, the noise in ChIP-on-chip data needs to be accounted for in the methods that employ this information for inference of regulatory activities. The results of these methods inferring false positives/negatives in the ChIP-on-chip data can then be taken as testable hypothesis which can be tested experimentally.

Inference in transcriptional regulation has been studied with many statistical approaches. The methods proposed for understanding of transcriptional regulation reveal two different but related aspects: the response of TF proteins to environmental signals in terms of the changes in their concentrations levels or *transcription factor activity* (TFA); and the strengths of the interactions or *connectivity strength* (CS) between the TF protein and the downstream target *i.e.* gene. Depending upon the nature of expression data (time-series or static), reconstruction algorithms attempt to learn the unobserved regulatory signal (TFAs) and the unobserved connectivity strengths (CSs). All the methods discussed here assume that the regulatory strengths do not change over time; however, the nature of reconstructed regulatory signal depends on whether the expression data is time-series or not. These methods can be viewed as network inference methods for known network topology as TFs and genes can be perceived as the interconnected components of a network with TF playing a dominant role in controlling the expression patterns of connected genes. Figure 3.1 depicts the interactions between TFs and genes in a gene regulatory network. It shows that the proteins alone or sometimes in the form of complexes activate or represse the expression of genes. The activation or repression of genes indicated by positive and negative signs implies that the proteins increases or decreases the rate of production of mRNA of the target genes.

One class of these methods attempts to learn the structure of the network as well as the TFAs and CSs using gene expression data (Nachman et al., 2004; Beal et al., 2005). These methods are computationally more intensive compared to inference methods for regulatory activities with known network topology. The computational complexity arises due to either exhaustive

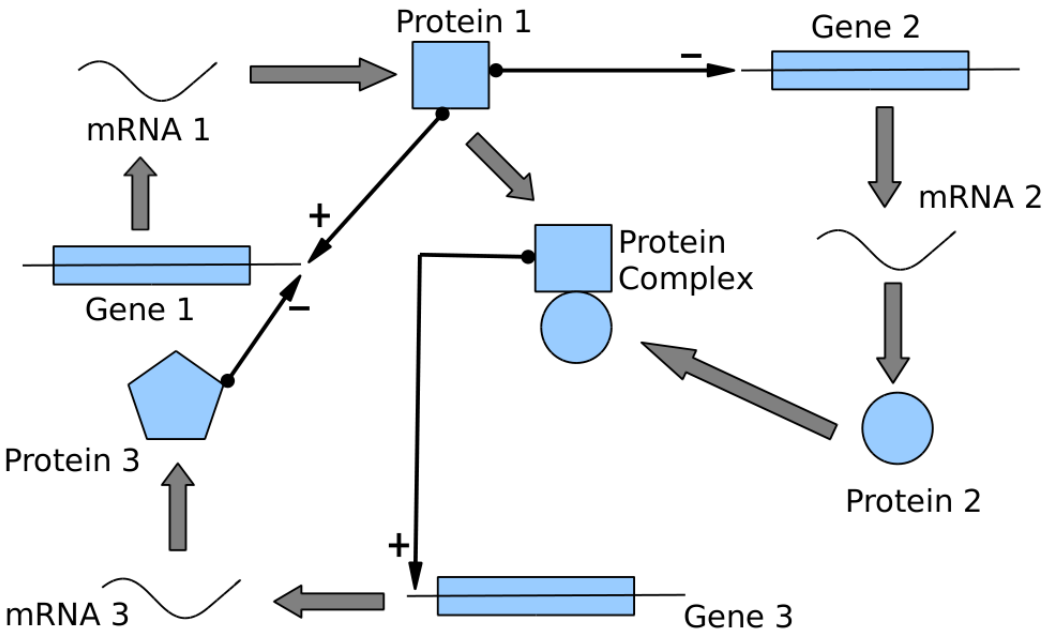


Figure 3.1: Schematic illustration of transcriptional regulation in gene regulatory network. Arrows with positive/negatives signs represent the CS with which a particular TF is effecting the target gene. The activities of TFs (TFAs) are inferred from the mRNA measurement of their target genes.

search technique (Nachman et al., 2004) and the absence of sparsity constraint (Beal et al., 2005) which implies that the method can only be applied to small networks where highly replicated data is available. Due to higher computational complexity, these methods are less feasible for genome-wide studies. Apart from this, these methods do no employ the known structure of the regulatory network made available by ChIP-on-chip. This architectural information is available for model organisms such as *E.coli* and *S.Cerevisiae* and unveils the regulators of a target gene in the regulatory network. Incorporating the prior knowledge about the regulatory interaction in an inference method has a significant advantage; it dramatically reduces the search space by exploiting the biological fact that only a few TFs in the regulatory network are regulating a particular gene. As an added benefit, the methods employing the prior knowledge about the structure of the regulatory network are more suitable for genome-wide studies due to their computational efficiency. The latter class of these methods is the subject matter of this chapter.

In the rest of this chapter, four statistical inference methods for transcriptional regulation are discussed with their merits and demerits. These methods cover a broad range of methodologies proposed for inference in transcriptional regulation employing different statistical components. One aspect common to all these methods is that they employ prior biological knowledge of the

regulatory network such as ChIP-on-chip data or sequence data to improve the results of inference. Each section of this chapter reviews one method from three different aspects: biological motivation, mathematical formulation and assessment of convergence of algorithm. At the end, a discussion is presented to conclude the chapter.

3.2 Network Component Analysis (NCA)

Network components analysis (NCA) is a data decomposition technique for reverse engineering the TF activities and the strengths by which TFs promote or repress the target genes. This method uses partial knowledge of regulatory network architecture and gene expression data to reconstruct the regulatory signals (TFA) and strengths (CS). As opposed to other data decomposition techniques such principal component analysis (PCA) or independent component analysis (ICA), this method does not ignore the biological network structure and provides the decomposition of the output signal into biologically meaningful signals. NCA utilises the prior knowledge about the connectivity of the regulatory network; it is done by subjecting the prior knowledge to certain criteria such that this connectivity information is sufficient to solve the network reconstruction problem and guarantees the uniqueness of the decomposition. NCA method is computationally efficient and well-suited for genome-wide network analysis.

3.2.1 Transcriptional Regulation Model of NCA

The gene expression data collected in matrix $\mathbf{E}$ with N genes and M time points is decomposed as

$$\mathbf{E} = \mathbf{A}\,\mathbf{P} \tag{3.1}$$

where $\mathbf{A}$ is $N \times L$ matrix composed of connectivity strengths between TFs and genes; $\mathbf{P}$ is $L \times M$ matrix that contains the TF profiles and L is number of TFs ($L \ll N$).

The solution to the inverse problem of (3.1) is not unique so this decomposition problem is constrained by using a nonsingular matrix $\mathbf{X}$ such that

$$\mathbf{E} = \mathbf{A}\,\mathbf{X}\,\mathbf{X}^{-1}\,\mathbf{P} = \bar{\mathbf{A}}\,\bar{\mathbf{P}} \tag{3.2}$$

where the matrix $\mathbf{X}$ can only be a diagonal matrix due to the constraints imposed on matrix $\mathbf{A}$ (Liao et al., 2003). To obtain a unique decomposition of $\mathbf{E}$ into $\mathbf{A}$ and $\mathbf{P}$ using NCA, the following criteria must be satisfied:

1. The matrix $\mathbf{A}$ must have full column rank.

2. The matrix $\mathbf{P}$ must have full row rank.

3. Each column of **A** must have at least $L-1$ zeros.

If all these criteria are satisfied then the decomposition is guaranteed to provide a unique solution consisting of matrix **A** that contains the CSs between all the TFs and genes and a matrix **P** that contains the TFAs for all TFs. To obtain this decomposition, an initial guess for **A** is constructed by setting all the $a_{ij} = 0$ for which there are no interactions in the regulatory network; other entries are initialised to any arbitrary number. The following constraint optimisation then provides the unique decomposition:

$$\min_{A,P} \|\mathbf{E} - \mathbf{A}\,\mathbf{P}\|^2$$

subject to

$$\mathbf{A} \in Z_0$$

$$a^l_{i,j} \leq a_{i,j} \leq a^u_{i,j}$$

$$p^l_{i,j} \leq p_{i,j} \leq p^u_{i,j}$$

where the norm is the matrix Frobenius norm and Z_0 is the topology derived from known network connectivity pattern. The constraints $a^l_{i,j}$, $a^u_{i,j}$, $p^l_{i,j}$ and $p^u_{i,j}$ are to ensure that the elements of **A** and **P** are biologically meaningful. The above constrained optimisation problem can be solved in a two-step iterative optimisation procedure by updating matrices **A** and **P** in two stages as follows:

Initialisation: Z_0 is used to initialise $\mathbf{A}_0$ with all non-zeros entries set to randomly selected non-zeros numbers.

Update for P: Using A_{k-1}, compute P_k by solving the following least-square optimisation problem

$$\min_{P_k} \|\mathbf{E} - \mathbf{A}_{k-1}\,\mathbf{P}_k\|^2$$

subject to

$$p^l_{i,j} \leq p_{i,j} \leq p^u_{i,j}.$$

Update for A: Use P_k to compute A_k

$$\min_{A_k} \|\mathbf{E} - \mathbf{A}_k\,\mathbf{P}_k\|^2$$

subject to

$$a^l_{i,j} \leq a_{i,j} \leq a^u_{i,j}$$

$$A_k \in A(Z_0).$$

NCA utilises ChIP-on-chip data for constructing the prior network topology which is known to contain false positives. As this is not a probabilistic method, it is not clear how to identify false positive. However, Liao et al. (2003) describe a small value of estimated CS for a particular TF-gene interaction as an indicator for poor likelihood and use it to identify false positives in the results of the model.

3.2.2 Convergence Criterion

The monitoring of convergence of NCA is based on the error computed between the estimates and the true values *i.e.* ($\mathbf{E} - \mathbf{A}\ \mathbf{P}$) after every cycle of the optimisation algorithm. If the difference is less than a convergence threshold, then the desired degree of optimisation has been achieved.

3.3 Bayesian Sparse Hidden Component Analysis (BNCA)

A major limitation of NCA is the non-probabilistic nature of the algorithm that cannot incorporate different sources of uncertainty in the modelling. It is always useful to be able to see confidence intervals with the estimated values that provide a gauge for certainty of results. To take this into account, Sabatti and James (2006) proposed a modified form of NCA (referred to as BNCA later) which is probabilistic in nature.

This probabilistic technique is basically a two stage process to reconstruct the transcriptional networks. First stage consists of analysing biological literature to find any known TF-gene interactions. Based on the documented biological evidence, if TF j is known to regulate gene i then $z_{ij} = 1$; all other entries of Z are set to zero. This topology of the network is refined by analysing the DNA sequence for the target genes using Vocabulon (Sabatti and Lange, 2002). Furthermore, $\pi_{ij} = P(z_{ij} = 1) < 1$; magnitude of π_{ij} encodes the prior belief that the TF j regulates the gene i which is obtained from sequence analysis. To keep this prior uninformative, one can use $\pi_{ij} = 0.5$.

3.3.1 Transcriptional Regulation model of BNCA

This network topology which provides a static picture of the regulatory network is then used as the starting point for network reconstruction using the following model:

$$\mathbf{E} = \mathbf{A}\,\mathbf{P} + \Gamma \tag{3.3}$$

where **E**, **A** and **P** have same meaning as before and $\Gamma = [\gamma_{it}], \gamma_{it} \sim \mathcal{N}(0, \sigma_i^2)$ is to account for measurement error and biological variability. During the reconstruction of the network, NCA's

identifiability criteria for the network topology are relaxed to get biologically more meaningful networks. In NCA, the position of zeros in matrix **A** encoding CS is assumed to be known a priori for the sake of identifiability and some TFs are to be removed from this matrix in order to make it consistent with NCA criteria. BNCA, however, does not assume any prior network topology and attempts to build the hidden regulatory activities of the regulatory network by employing two sources of information at two stages of the algorithm; these two sources of information are sequence data and gene expression data.

To cast the model in the Bayesian framework, prior probability distributions are specified for all the variables in the model. All p_{jt} are assumed to be a priori independent and follow a Gaussian distribution

$$p_{jt} \sim \mathcal{N}(0, \sigma_p^2).$$

Similarly, $a_{ij} = 0$ if $z_{ij} = 0$ otherwise $a_{ij} \sim \mathcal{N}(0, \sigma_a^2)$. Finally, σ_i^2 (the variance of γ_i) is taken to be an inverse gamma distribution with hyper-parameters α_i and β_i; values for which can be computed from biological replicates or calibration slides of experiments.

Let z^i denote the set of TFs that regulate gene i, π^i denote the prior probabilities with which the regulators of gene i are regulating its expression, σ represent the vector of all the variances σ_i, Σ is a diagonal matrix whose diagonal elements are elements of σ and a_i encodes the strengths with which gene i is regulated by its regulators. Then the posterior analysis can be done if the following conditional posterior distribution are sampled in an MCMC iteration,

$$\begin{aligned}
\text{sample } z^i &\sim P(z^i|\mathbf{P},\sigma,\mathbf{E}) \qquad \text{for } i = 1,\ldots,N \\
\text{sample } a_i &\sim P(a_i|\mathbf{Z},\mathbf{P},\sigma,\mathbf{E}) \qquad \text{for } i = 1,\ldots,N \\
\text{sample } p_t &\sim P(p_t|\mathbf{Z},\mathbf{A},\sigma,\mathbf{E}) \qquad \text{for } t = 1,\ldots,M \\
\text{sample } \sigma_i &\sim P(\sigma_i^2|\mathbf{Z},\mathbf{A},\mathbf{P},\mathbf{E}) \qquad \text{for } i = 1,\ldots,N.
\end{aligned}$$

The above conditional posterior distributions are specified as

$$\begin{aligned}
P(z^i|\mathbf{P},\sigma,\mathbf{E}) &\propto \pi^{i(z^i)}(1-\pi^i)^{(1-z^i)}/\sigma_a^{|z^i|} \times \det(\mathbf{P}[z^i]\mathbf{P}[z^i]'/\sigma_i^2 + \mathbf{I}_{|z^i|}/\sigma_a^2)^{-\frac{1}{2}} \\
&\quad \times \exp\{\frac{1}{2\sigma^4} e^{i'}\mathbf{P}[z^i]'(\frac{\mathbf{P}[z^i]\mathbf{P}[z^i]'}{\sigma_i^2} + \frac{\mathbf{I}_{z^i}}{\sigma_a^2})\mathbf{P}[z^i]e^i\} \\
a_i|\mathbf{P},\mathbf{Z},\sigma^2 &\sim \mathcal{N}(\Sigma_{a_i}\mathbf{P}[z_i]e^i/\sigma_i^2, \Sigma_{a_i}) \\
p_t|\mathbf{A},\mathbf{Z},\sigma^2 &\sim \mathcal{N}(\Sigma_{p_t}\mathbf{A}_Z^{'}\Sigma^{-1}e_t, \Sigma_{p_t}) \\
\frac{1}{\sigma_i^2}|\mathbf{A},\mathbf{Z},\mathbf{P} &\sim \text{Gamma}(\tilde{\alpha}_i, \tilde{\beta}_i)
\end{aligned}$$

where $x^y = \prod_{i=1}^r x_i^{y_i}$ for two vector x and y, $a[z^i]$ is a vector of elements of a corresponding to non-zero entries of z^i, $\mathbf{P}[z^i]$ is a submatrix containing selected rows of $\mathbf{P}$ for which $z^i \neq 0$,

$\mathbf{A}_Z$ is a matrix with same dimension as $\mathbf{A}$ and its elements are set to zeros corresponding to zero elements of matrix $\mathbf{Z}$, $\mathbf{I}_r$ is an identity matrix of rank r, $|z^i|$ is the number of elements in the set z^i, $\Sigma_{a_i} = (\mathbf{P}[z^i]\mathbf{P}[z^i]'/\sigma_i^2 + \mathbf{I}_{z^i}/\sigma_a^2)^{-1}$, $\Sigma_{p_t} = (\mathbf{A}_Z'\Sigma^{-1}\mathbf{A}_Z + \mathbf{I}_L/\sigma_p^2)$, $\tilde{\alpha_i} = \alpha_i + M/2$, e^i is the column vector of ith row of matrix $\mathbf{E}$, e^t represent the tth column of matrix $\mathbf{E}$, e_{it} is the expression level of gene i in the experiment t and $\tilde{\beta}_i = \beta_i + \sum_{t=1}^{M}(e_{it} - \sum_{j=1}^{L} a_{ij}p_{jt})^2/2$. Derivation of the above conditional posterior distributions can be found in the supplementary material of Sabatti and James (2006).

3.3.2 Convergence Monitoring

As the posterior estimation is based on Markov chain Monte Carlo (MCMC) simulations, the number of required iterations is not known. However, many different diagnostics are available to test the convergence of the simulated Markov chains. The authors used Cowles and Carlin (1996) for diagnosing the convergence of MCMC simulations.

Sabatti and James (2006) provide an algorithm for the reconstruction of the regulatory network where the temporal structure of the data is not taken into account. The proposed algorithm in their work can, in principle, be extended to account for time dynamics by setting

$$p^j \sim \mathcal{N}(0, \Gamma) \tag{3.4}$$

where Γ is $M \times M$ covariance matrix. However, the conditional independence structure used to derive the conditional posterior distribution before does not hold in this case and the authors propose to use a different parametrisation for incorporating time dependance in the prior and the posterior distributions for p_j. The new parametrisation involves inversion of relatively big matrices (of the order of $M \times L$) due to which genome-wide application of this method becomes less feasible but efficient inversion algorithms can be used to overcome this computational bottleneck. Another difficulty lies with their approach towards specifying a prior over binary connectivity matrix that can not be trivially extended for ChIP-on-chip data.

3.4 Probabilistic Inference of TFA using State Space Model

Sanguinetti et al. (2006) proposed to use a state space model (SSM) to infer the concentrations of TFAs and their effect on each target gene from gene expression data. SSM are a special case of dynamic Bayesian networks (DBN) with Markov chain prior on continuous-valued latent variables. Although SSMs have been previously used to learn the structur of the regulatory network interactions in Beal et al. (2005), prior knowledge about the regulatory interactions (*i.e.* ChIP-on-chip data) was not used to explicitly infer TFAs. The method proposed in Sanguinetti

et al. (2006) makes use of this prior knowledge in a probabilistic model to infer TFAs and CSs which greatly reduces the search space. An efficient variational Bayesian expectation maximisation (VBEM) algorithm is proposed for inference in this model. Owing to efficient implementation and exploitation of sparseness of the regulatory network, the proposed method is a practical tool for genome-wide analysis in transcription regulation.

3.4.1 Model for Transcriptional Regulation using SSM

This method employs a log-linear approximation to the dynamics of transcription and is based on a state space model of the following form

$$\begin{aligned} y_n(t) &= \sum_{m=1}^{q} X_{nm} b_{nm} c_m(t) + \mu_n + \varepsilon_{nt} \\ c_m(t) &= \gamma_m c_m(t-1) + \eta_{mt}. \end{aligned} \tag{3.5}$$

Here, $y_n(t)$ is the mRNA log-expression level for gene n at time t, $\mathbf{X}$ is a binary *connectivity matrix* (assumed known) encoding whether gene n is bound by TF m, b_{nm} encodes the regulatory strength with which TF m effects gene n, and $c_m(t)$ is the concentration of active TF m at time t, μ_n is the base expression level of gene n when it is not bound by any TF, ε and η are experimental and process noise respectively. The model then specifies Gaussian prior distributions over the concentrations $c_m(t)$ and strengths b_{nm} and uses a factorized variational approximation to infer posterior distributions given mRNA time course observations. Notice that the probabilistic nature of the model means that noise is treated in a natural and principled way, and estimates of the quantities of interest are always associated with a measure of the corresponding uncertainty. The details about the method and the derivations of the proposed VBEM algorithm can be seen in section 4.2.1 and 4.2.2.

3.4.2 Convergence Monitoring

After every iteration of the VBEM algorithm, update in the likelihood is calculated with the new values of model parameters and latent variables. The model is deemed to have converged if the update in the likelihood between two consecutive iterations is less than a certain threshold.

3.4.3 TFInfer - An Open-source Implementation

An open-source implementation (TFInfer) of the method proposed in Sanguinetti et al. (2006) is described in Asif et al. (2010). TFInfer is an open-source standalone software designed to infer the relative activities of transcription factor proteins based on gene-expression data.

Using gene-expression data combined with the architectural information about the regulatory network, activities of transcription factor proteins can be estimated in a computationally efficient way. TFInfer can handle time-series gene-expression data and gene-expression data from several independent conditions with or without replicates. Implementation is done using .Net framework (or equivalent on Linux), so it is a requirement that user either have Microsoft.Net on Microsoft Windows or mono[1]s on the other platforms. dnAnalytics[2], an open-source numerical library in C# and ZedGraph, an open-source plotting tool in C#, are used for the implementation of this software. This software is available on most OSes where support for either Microsoft.Net or mono is available. In chapter 3, we present the details of the methods implemented by the software and the functionalities of the software.

3.5 A Combined Expression-Interaction Model for Inferring TFAs

One of the fundamental reasons to infer TFAs from gene expression data can be attributed to the fact that biosynthesis of proteins is not only dependent on transcription of genes. The biosynthesis of proteins is also effected by post-transcriptional modifications (PTM) such as post-translational modifications, phosphorylations etc. So, inference of TFAs from expression data accounts for post-transcriptional modifications as TFAs are treated as latent variables but these methods do not explicitly incorporate PTMs in their models.

While all the methods discussed in the previous sections take post-transcriptional modifications into account by treating TFAs as unobserved, these methods only use one source of information which is expression patterns of the regulated genes. Another proxy for the activities of TFs could be the measured mRNA levels of TFs when TFs are not post-transcriptionally modified. Shi et al. (2008) proposed a method to combine both sources of information in one method. To infer TFA, they use mRNA expression levels of a TF when the TF is transcriptionally regulated and mRNA expression levels of target genes of the TF when the TF is post-transcriptionally regulated. Based on a latent indicator variable, that specifies whether the TF is transcriptionally regulated or post-transcriptionally modified, they select a model out of two models to reconstruct the hidden regulatory activity. This method is referred as Post-Transcriptional Modification Model (PTMM).

PTMM is a variant of factorial hidden Markov Model (FHMM, (Ghahramani and Jordan, 1997)) where activity of each TF is modelled as hidden Markov Chain with correlation between

[1] http://www.mono-project.com/
[2] http://dnanalytics.codeplex.com/

the hidden activity of a TF with its (observed) expression level. This correlation is embedded in FHMM by using a hidden indicator variable for each TF to designate if the TF is post-transcriptionally modified or not. In case of PTM, the hidden TFA is inferred from the activity levels of its regulators. In the other case, the hidden TFA is inferred by using the measured mRNA levels of TF.

3.5.1 Post-transcriptional Modification Model

Let m be the number of genes for which expression measurements are available under a variety of experimental conditions. Out of these m genes, n are TFs where $n < m$. So the PTMM models the joint probability distribution over multiple time-series expression levels of genes, hidden TFA and hidden post-transcriptional status of all TFs. $G_{i,d,t}$ represents the observed expression levels of gene i at time t in dataset d where first n genes are also TFs. Similarly, $T_{j,d,t}$ represents the hidden activity of TF j at time t in dataset d. For each TF j, a global binary indicator variable Z_j is used to denote if this TF is post-transcriptionally modified or not. Z_j follows a Bernoulli distribution with parameter ρ. Z_j specifies which transcriptional model TF j follows out of the following:

$$T_{j,d,t} \sim \begin{cases} \mathcal{N}(G_{j,d,t-1}, \tau_d^2) & \text{if } Z_j = 0 \\ \mathcal{N}(T_{j,d,t-1}, \gamma_d^2) & \text{if } Z_j = 1 \end{cases}$$

In case of PTM ($Z_j = 1$), activity of TF j is modelled as hidden Markov chain with γ_d^2 specifying the variability of TFA between two consecutive time-points. In case there are no PTM ($Z_j = 0$), activity of TF j is modelled as a noisy realisation of its gene's expression profile with one time-point lag. The initial time-point in this case is modelled by Gaussian distribution with zero mean and σ_d^2 variance. This dataset-specific variance allows integrating datasets with different initial condition for TFA.

PTMM models the expression profile of a gene as the linear superposition of contributions ($w_{i,j}$) of its regulators; if there are no regulators present for a particular gene in a dataset, then the gene expression for that gene is modelled as zero mean Gaussian:

$$G_{i,d,t}|T_{:,d,t} \sim \begin{cases} \mathcal{N}(\sum_{j=1}^{n} w_{i,j} T_{j,d,t}, \beta_d^2) & \text{if gene } i \text{ is regulated by at least 1 TF} \\ \mathcal{N}(0, \alpha_d^2) & \text{otherwise} \end{cases}$$

Having different variances (α_d^2 and β_d^2) encodes the intuition that the genes without any regulators may have higher variances due to the deficiencies of the model. PTMM uses different

variance for each dataset which represent the variability between noise levels of different experiments. As with the models previously discussed in this chapter, the interactions between TF and genes are assumed to be time-independent and are shared across all the datasets. PTMM parameters are learnt using approximate expectation maximisation algorithm (EM) which minimises the penalised likelihood score given by

$$\begin{aligned}\text{Score}(\mathbf{o},\mathbf{h},\mathbf{z}:\mathbf{W},\theta) = \ln(P(\mathbf{z})) + \sum_{d=1}^{D} \ln(P(\mathbf{o}_d,\mathbf{h}_d|\mathbf{z},\mathbf{W},\theta)) \\ - \lambda_1 \sum_{i=1}^{m}\sum_{j=1}^{n} |w_{i,j}| - \lambda_2 \left[\sum_{i=1}^{m}\sum_{j=1}^{n} \delta(w_{i,j} \neq 0)\{E_{i,j}\pi_1 + (1-E_{i,j})\pi_0\} \right. \\ \left. + \sum_{i=1}^{m}\sum_{j=1}^{n} \delta(w_{i,j} = 0)\{E_{i,j}\pi_0 + (1-E_{i,j})\pi_1\} \right] \qquad (3.6)\end{aligned}$$

$$\text{subject to:} \qquad \left(\left|\{w_{i,j}|w_{i,j} \neq 0, 1 \leq j \leq n\}\right| \leq C\right) \qquad \text{for all } i$$

where **o** and **h** are observed gene expression and hidden activity levels of TFs in dataset d.

This penalised likelihood score contains two regularisation terms. The first regularisation term imposes penalty on the weights ($w_{i,j}$) and forces them to be zero which has the biological notion that most TF-gene interactions should be zeros. The second regularisation term incorporates the prior network knowledge from binding experiments whereby $E_{i,j} = 1$ if gene i is (a priori) regulated by TF j and 0 otherwise. $\delta(.)$ function results in 0 or 1 if the condition is false or true respectively.

There are two penalty terms too in the penalised likelihood score of (3.6). The first penalty term π_0 is used when the model selects a regulatory link which is inconsistent with prior knowledge while the second penalty term π_1 is used when the model selects a regulatory link which is consistent with prior network structure. It is obvious to set $\pi_0 >> \pi_1$. C is the maximum number of regulators for genes which represent underlying biological notion that most of the genes in regulatory network are regulated by only a few TFs.

The purpose of regularisation and penalty terms in the penalised likelihood score of equation (3.6) is to encourage the model to selection those TF-gene interaction that are consistent the with the prior knowledge. However, the results of the model may deviate from the prior knowledge when the incurred penalty is less than the gain in the likelihood. These deviations in the results reflects the noise in the prior knowledge which can be handled efficiently by using penalised likelihood score.

3.5.2 EM Algorithm for PTMM

The EM algorithm for PTMM iteratively updates the model parameters (**W** and θ) in the M-step and hidden variables (**h** and **z**) in the E-step until the convergence is achieved.

E-step: In this step, two expectations are to be computed based on the current values of model parameters θ and **W**; expected values of hidden activity levels of TFs and hidden indicator variables for PTM. For this purpose, a generalised mean field algorithm (Xing et al., 2003) is used. Xing et al. (2003) is a generalised mean field approach for inference in graphical models where a complex distribution is approximated with a distribution that factorizes over disjoint of the graph.

Based on the current value of indicator variables **z** for TF j and the expected activity levels of all other TFs, posterior distribution for $T_{j,d,t}$ can be inferred using one of the following two ways: if $Z_j = 0$, TF j is not post-transcriptionally regulated and the posterior distribution of $T_{j,d,t}$ can be computed for each time-point independently as there is not correlation between $T_{j,d,t}$ and $T_{j,d,t-1}$. The prior in this case is a Normal distribution with mean given by the expression level of the gene corresponding to TF j at time $t-1$ and the variance given by τ_d^2. The posterior distribution in this case is dependent on the expression levels of the genes regulated by TF j as well as the activity levels of other TFs that are regulating the gene for TF j. In case of $Z_j = 1$, TF j is post-transcriptionally regulated then its activity levels can be inferred by treating it as a hidden Markov chain.

The expected values for latent indicator variables is determined by examining which model better explains the behaviour of TF j *i.e* whether the TF j is better explained by the model with PTM or without it. This is done by computing the likelihood with both types of models and selecting the value of Z_j appropriately.

M-step: The updated values of model parameters are computed, given the expected values of latent variables, by maximising the likelihood score function given in (3.6). An exact solution can be obtained for γ, σ, τ by setting the derivative of score function in (3.6) equal to zero. For α and β, maximum likelihood estimates can be obtained by fixing the TF-gene interactions weights (**W**). However, **W** cannot be computed in closed form and a greedy search method is proposed for inferring the the most likely estimates for elements of **W** in Shi et al. (2008).

3.5.3 Convergence Monitoring

The authors in Shi et al. (2008) analyse the effect of more datasets (d) on the performance of proposed EM algorithm using precision recall curve. Their results show that the results are improved for both precision and recall when more datasets are used. The convergence is

Method	Prob-abilistic?	Time Dynamics	Basic Model	Inference technique
NCA	No	No	Regression-based	Constraint Optimisation
BNCA	Yes	No	Regression-based	MCMC sampling
TFInfer	Yes	Yes	SSM	Variational EM algorithm
PTMM	Yes	Yes	FHMM	EM algorithm

Table 3.1: Comparison of different methods. This table summarises the features of the methods discussed in this chapter which are probabilistic nature (or not), support to handle time-series data, underlying model of the method and inference technique used.

monitored by evaluating the penalised likelihood score (3.6) until the desired convergence level is achieved.

3.6 Discussion and Conclusion

This chapter provides an overview of statistical methods for inference in transcriptional regulation using different statistical tools. All the methods discussed here aim to infer CSs and TFAs in gene regulatory network where the network connectivity pattern is known. While BNCA does not use the network connectivity information directly from ChIP-on-chip data, initial guess for the regulatory network architecture is obtained from biological literature and is further refined by analysing the sequence data. However, the task of building the regulatory network architecture from biological literature is cumbersome and analysis of sequence data poses further challenges. An alternative, employed by other models except BNCA, is to exploit the network connectivity pattern available from ChIP-on-chip. NCA is not a probabilistic method which means lack of confidence intervals with the results; due to this it is hard to identify false positives. Other methods, being probabilistic, are capable to identify false positives due to the availability of confidence intervals in their results. Table 3.1 summarises the main features of these methods in terms of the underlying statistical model employed, statistical approximation technique used and whether the method is probabilistic or not.

Another class of methods is available that learns the structure of the regulatory network using gene expression where no prior assumptions are made about the architectural patterns of the regulatory network (Nachman et al., 2004; Beal et al., 2005). Although these methods provides biologically meaningful results, computational cost associated with these techniques is usually quite high which hinders the applicability of these methods to genome-wides studies. Also, these methods require large amounts of data (or highly replicated data) which is usually not available from biological experiments. An important feature of the regulatory architecture data

is its sparse nature. The methods employing this information to infer the regulatory activities have significant advantage that it reduces the search space by limiting the number of parameters to be inferred based on the presence or absence of regulatory link. Due to this, these methods are more feasible for genome-wide studies.

Another criterion for selecting the appropriate model could be the nature of approximation technique used in the inference method. While MCMC sampling and variational inference provides comparable results, convergence diagnostics are required for sampling techniques; also, MCMC sampling is known to be computationally expensive compared to variational inference. On the other hand, variational methods are not considered the best approximation when uncertainty about the results is of crucial importance; however, variational methods perform well in terms of the associated computational cost and their convergence is easier to monitor.

In general, the methods reviewed in this chapter make some simplifying assumptions; mostly these methods approximate the complex biological processes such as transcriptional regulation with additive linear models. Also, the noise of the microarray is approximated by zero mean Gaussian which effects the results of the model. Another assumption is about the regulatory activities which are assumed to be constant over time. The combinatorial effect of TFs in regulating the target genes (Asif and Sanguinetti, 2011) is also ignored in all these methods. Most of these assumptions are made in order to make the model identifiable and keep it applicable to genome-wide studies.

These methods have proven to be useful in many cases and provide novel biological insights (Partridge et al., 2007; Davidge et al., 2009; Rolfe et al., 2011). The availability of architectural data about the gene regulatory network with abundance of gene expression data means that these methods can be routinely used to infer the hidden TFAs and CSs. The quantitative analysis reveals hidden regulatory relationships between TFs and genes which is otherwise unavailable due to experimental difficulties in measuring these quantities.

Chapter 4

TFInfer - A Tool for Probabilistic Inference of Transcription Factor Activities

In chapter 3, a brief description of a method based on SSM (Sanguinetti et al., 2006) was discussed without the mathematical derivation of the VBEM algorithm. In this chapter, details of this method including the derivation of the VBEM algorithm for time-series and non time-series data are given. The VBEM algorithm is implemented in an open-source implementation (TFInfer) with additional features as discussed later in this chapter. TFInfer is a novel open-access, standalone tool for genome-wide inference of transcription factor activities from gene expression data. It has been significantly optimised in terms of performance, and it was given novel functionality, by allowing the user to model both time-series and data from multiple independent conditions. With a full documentation and intuitive graphical user interface, together with an in-built database of yeast and *E. coli* transcription factors, the software does not require any mathematical or computational expertise to be used effectively.

4.1 Introduction

Transcription regulatory networks play a fundamental role in mediating external signals and coordinating the response of the cell to its changing environment. Recent technological advances in molecular biology, such as ChIP-on-chip and ChIP-seq, are uncovering an increasing amount of data about the static structure of these networks, providing us with information about interactions between promoters and specific TF. However, despite these advances, intracellular concentrations of active TF proteins remain very challenging to measure directly in a dynamic fashion, thus limiting our ability to understand the dynamics of transcriptional regulation. To obviate these problems, several research groups have proposed statistical approaches that infer TF activity levels by combining connectivity data about the structure of the regulatory network with microarray data. In this chapter, a novel implementation of one of these methods (San-

guinetti et al., 2006) along with the mathematical derivation is given which makes it freely available to the academic community in an intuitive, user-friendly platform with additional features as discussed later in the chapter.

In the following sections of this chapter, the VBEM algorithm implemented by TFInfer is derived for time-series and non time-series data followed by some results on synthetic data for two models. Then the salient features of the software are discussed with specific implementation details. At the end, the chapter concludes with a discussion.

4.2 Transcriptional Regulation Model of TFInfer

In this model, logged gene expression data from a time-series or time-independent microarray experiment is denoted by a matrix $\mathbf{Y} \in \Re^{N \times T}$, where N is the number of genes and T is the total number of time points or experimental conditions in the dataset. The underlying assumption is that the gene expression is driven by M transcription factors. The model is a log-linear approximation to the non-linear relationship between changes in TFAs and gene expression. A discrete-time SSM is used where gene expression for gene n is modelled as a linear combination of the activities of its regulators

$$y_n(t) = \sum_{m=1}^{q} X_{nm} b_{nm} c_m(t) + \mu_n + \varepsilon_{nt} \tag{4.1}$$

The matrix $\mathbf{X}$ is a binary matrix whose nm entry is one if and only if gene n is regulated by TF m. This matrix is known from biological literature or it can be obtained from ChIP technique. The activity matrix $\mathbf{B}$ encodes the CS with which TF m regulates the gene n. b_{nm} and μ_n are given zero mean Gaussian priors. To incorporate the baseline expression for each gene, vector $\mu = [\mu_n]$ is used in the SSM model of equation (4.1). The matrix $\mathbf{C}$ (encoding $c_m(t)$) represents the relative concentration of the TF m at specific experimental condition or time point t. For measurement noise, ε_{nt} is used with i.i.d. Gaussian noise assumption ($\varepsilon_{nt} \sim \mathcal{N}(0, \sigma^2)$). The matrix $\mathbf{X}$ is usually very sparse showing that very few TFs bind to a specific gene and this sparse nature of $\mathbf{X}$ is used to ensure that only required b_{nm} are estimated.

4.2.1 Model for Non Time-series Gene Expression Data

To incorporate the time-independent nature of the gene expression data, the row vector of concentrations is formalised as

$$\mathbf{c}(1) \ldots \mathbf{c}(T) \sim \mathcal{N}(0, \mathbf{K}), \tag{4.2}$$

the matrix **K** is an identity matrix in this case. The joint distribution for observed and latent variables is

$$p(\mathbf{Y},\mathbf{B},\mathbf{C},\mu) = p(\mathbf{Y}|\mathbf{B},\mathbf{C},\mu,\sigma^2)p(\mathbf{B}|\alpha^2)p(\mathbf{C}|\gamma)p(\mu). \tag{4.3}$$

$$p(\mathbf{Y},\mathbf{B},\mathbf{C},\mu) = \left[\prod_{n=1}^{N}\prod_{t=1}^{T}\mathcal{N}\left(y_n(t)|\sum_{m=1}^{q}\mathbf{X}_{nm}b_{nm}c_m(t),\sigma^2\right)\right].\left[\prod_{n=1}^{N}\prod_{m=1}^{q}\mathcal{N}\left(b_{nm}|0,\alpha^2\right)\right] .\mathcal{N}\left(\kappa|\mathbf{0},\mathbf{K}\right)\mathcal{N}\left(\mu|\mathbf{0},\mathbf{I}\right) \tag{4.4}$$

where κ is a vector obtained by concatenating the transcription factor concentrations at various time points.

Marginalization of the above equation is intractable so a VBEM algorithm is used to approximate the true posterior distribution. The VBEM algorithm is used to minimise the KL divergence between the approximating and the true posterior distribution in the following way

$$\ln(p(\mathbf{Y}|\theta)) \geq \langle \ln\ p(\mathbf{Y},\mathbf{B},\mathbf{C},\mu|\theta)\rangle_{q(\mathbf{B},\mathbf{C},\mu)} + H(q) \tag{4.5}$$

$\langle\rangle_q$ denotes the expectation under the probability distribution q; $q(\mathbf{B},\mathbf{C},\mu)$ is the approximating distribution over the variables **B**, **C** and μ; and $H(q)$ is the entropy of the distribution. The approximating distribution over the parameters factorizes as

$$q(\mathbf{B},\mathbf{C},\mu) = q_1(\mathbf{B})q_2(\mathbf{C})q_3(\mu). \tag{4.6}$$

Using this factorisation, the VBEM algorithm is initialised with prior distributions for **B**, **C** and μ. The approximating distributions are, then, updated iteratively until the convergence is achieved.

The update equations for E-step and M-step of the VBEM algorithm are described next .

E-Step: During the E-step of variational Bayesian EM algorithm, approximating distributions are updated according to the following update equations. These update equations can easily be obtained by taking the expectation of the joint likelihood in equation (4.4) w.r.t. all the variables except the variable to be approximated. For $q_1(B)$, this comes out to be

$$q_1(\mathbf{B}) = \prod_{n=1}^{N}\mathcal{N}\left(\mathbf{b}_n|\mathbf{m}_n,\Sigma_n\right) \tag{4.7}$$

where

$$\Sigma_n = \left(\alpha^2\mathbf{I} + \frac{1}{\sigma^2}\sum_{(t=1)}^{T}\chi_n\langle\mathbf{c_t}\mathbf{c_t^T}\rangle_{q_2}\chi_n\right)^{-1}$$

$$\mathbf{m}_n = \Sigma_n \left(\sum_{t=1}^{T} \frac{(y_n(t) - \langle \mu_n \rangle_{q_3})}{\sigma^2} \chi_n \langle \mathbf{c_t} \rangle_{q_2} \right).$$

χ_n is the diagonal matrix with n^{th} row of $\mathbf{X}$ on the diagonal and b_n^T is the n^{th} row of $\mathbf{B}$. Similarly, the approximating distribution for $\mathbf{C}$ is given by

$$q_2(\mathbf{C}) = \mathcal{N}(\mathbf{c}(1) \ldots \mathbf{c}(T) | \nu, \mathbf{K}^{'}) \tag{4.8}$$

with

$$\mathbf{K}^{'} = \left(\mathbf{K^{-1}} + \mathbf{I_T} \otimes \frac{1}{\sigma^2} \sum_{n=1}^{N} \chi_n \langle \mathbf{b_n b_n^T} \rangle_{q_1} \chi_n \right)^{-1}$$

$$\nu = \mathbf{K}^{'} \left(\frac{y_n - \langle \mu_\mathbf{n} \rangle_{q_3}}{\sigma^2} \chi_n \langle \mathbf{b}_n \rangle_{q_1} \right).$$

Calculating $\mathbf{K}^{'}$ is efficient in this case compared to time-series data; this can further be improved if the posterior estimation is done in the following way

$$\langle \mathbf{c}(t) \rangle = \left(\mathbf{I_q} + \frac{1}{\sigma^2} \sum_{n=1}^{N} \chi_n \langle \mathbf{b_n b_n^T} \rangle_{q_1} \chi_n \right)^{-1} \left(\frac{y_n - \langle \mu_\mathbf{n} \rangle_{q_3}}{\sigma^2} \chi_n \langle \mathbf{b}_n \rangle_{q_1} \right).$$

The approximating distribution for μ is given by

$$q_3(\mu) = \prod_{n=1}^{N} \mathcal{N}(\mu_n | \zeta_n, \beta_n^2) \tag{4.9}$$

where

$$\zeta_n = \frac{\sigma^{-2}}{1 + T\sigma^{-2}} \sum_{t=1}^{T} (y_n(t) - \mathbf{b}_n^T \chi_n \mathbf{c}_t).$$

$$\beta_n^2 = (1 + T\sigma^{-2})^{-1}$$

The set of equations (4.7) to (4.9) constitute the E-step updates for the VBEM algorithm.

M-step: Fixed point update equations are available for α^2 and σ^2. For γ, optimisation is achieved using scaled conjugate gradient algorithm. The update equations for α^2 and σ^2 are given below,

$$\alpha^{-2} = \frac{1}{N} \sum_{n=1}^{N} \text{trace} \langle \mathbf{b}_n \mathbf{b}_n^T \rangle_{q_1} \tag{4.10}$$

$$\sigma^2 = \frac{1}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T} \left[y_n^2(t) - 2y_n(t) \langle \mu_n \rangle_{q3} + \langle \mu_n^2 \rangle_{q3} - 2(y_n(t) - \langle \mu_n \rangle_{q3}) \langle \mathbf{b}_n^T \rangle_{q1} \chi_n \langle \mathbf{c}_t \rangle_{q_2} \right.$$
$$\left. + \text{trace}(\langle \mathbf{b}_n \mathbf{b}_n^T \rangle_{q_1} \chi_n \langle \mathbf{c}_t \mathbf{c}_t^T \rangle_{q_2} \chi_n) \right]. \tag{4.11}$$

4.2.2 Model for Time-series Gene Expression Data

To model the dynamics of the transcription factor concentrations, first order Markov process is used as shown in equation (4.12).

$$c_m(t) = \gamma_m c_m(t-1) + \eta_{mt}. \tag{4.12}$$

where $\eta_m \sim \mathcal{N}(0, 1-\gamma_m^2)$. The variance of the process noise $(1-\gamma_m^2)$ ensures that the Markov process governing the dynamics of the $c_m(t)$ is stationary with unit variance ($c_m(1) \sim \mathcal{N}(0,1)$). The parameter vector $\gamma = [\gamma_m], \gamma_m \in [0,1]$ determines the temporal variability of TF m. The values of γ_m close to one corresponds to less variability in the activities of TF m while the values closer to zero indicate more variability in the activities of TF m. Intermediate values for γ_m corresponds to smoothly varying temporal profile of TF m.

Using the distribution given in equation 4.12 in the joint likelihood and taking the expectation of the joint likelihood w.r.t. q_1 and q_3, one obtains that

$$q_2(\kappa) = \mathcal{N}(\kappa|\nu, \mathbf{K}') \tag{4.13}$$

with

$$\mathbf{K}' = \left(\mathbf{K^{-1}} + \mathbf{I_T} \otimes \frac{1}{\sigma^2} \sum_{n=1}^{N} \chi_n \langle \mathbf{b_n} \mathbf{b_n^T} \rangle_{q_1} \chi_n \right)^{-1}$$

$$\nu = \mathbf{K}' \left(\frac{y_n - \langle \mu_\mathbf{n} \rangle_{q_3}}{\sigma^2} \chi_n \langle \mathbf{b}_n \rangle_{q_1} \right)$$

Notice that the state space model prior implies that the prior covariance matrix $\mathbf{K}$ is banded which can be exploited in an efficient matrix inversion algorithm. For time-series data case, $\mathbf{K}$ is of size $Tq \times Tq$. For genome-wide applications, size of this matrix becomes very large while increasing the time and space complexity for inversion; an optimised inversion algorithm for banded matrix (Asif and Moura, 2005) was used for the sake of efficiency.

This is the only change required to make the VBEM algorithm work with time-series gene expression data. Apart from this, the VBEM algorithm computes the expectations for $q_1(\mathbf{B})$ and $q_3(\mu)$ as in the previous section.

It is important to mention that by using $\gamma = 0$ in equation (4.12) gives the required solution for time-independent gene expression data but it is computationally expensive due to the inversion of the large matrix $\mathbf{K}$.

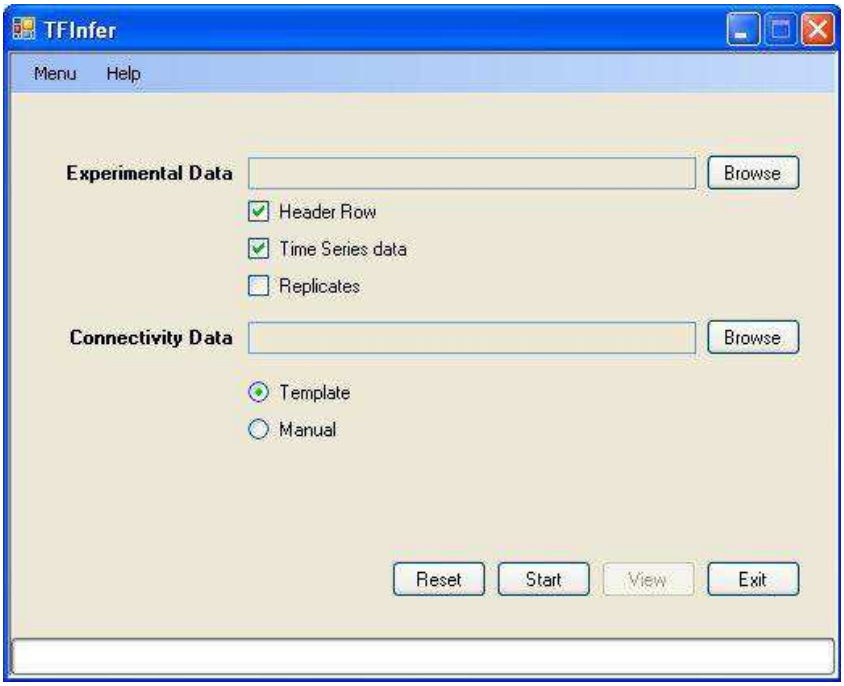


Figure 4.1: Main Interface of TFInfer

4.3 Software Overview

The model and GUI are implemented in C# which allows an efficient implementation of the variational Bayesian expectation maximisation (VBEM) algorithm. dnAnalytics, a C# open source library for scientific computing, is used for the numerical routines. ZedGraph, an open-source plotting tool, is used for displaying the results of the model in graphical format.

The main interface of TFInfer is shown in figure 4.1; the starting frame requires the user to browse for the expression data, specify its characteristics (time-series, replicates, etc) and browse for the connectivity data. If template connectivity is selected, the user is asked to select either a file for yeast (based on available ChIP-on-chip data) or a file for *E. coli* (compiled manually from the Ecocyc database[1]). Otherwise, the user can specify any binary connectivity matrix.

Once the data is selected, a summary of the data is displayed (number of genes and time points). If this is accepted, a list of all the TFs included in the connectivity matrix is displayed; the user can select a subset of TFs by clicking on the list of TFs names (figure 4.2). Once this is completed, the optimisation starts; its progress (with respect to a maximum number of iterations, default 1500) is monitored through a progress bar at the bottom of the screen.

Once the run is complete, the user can visualise TF activity profiles by clicking the box next

[1] `http://www.ecocyc.org/`

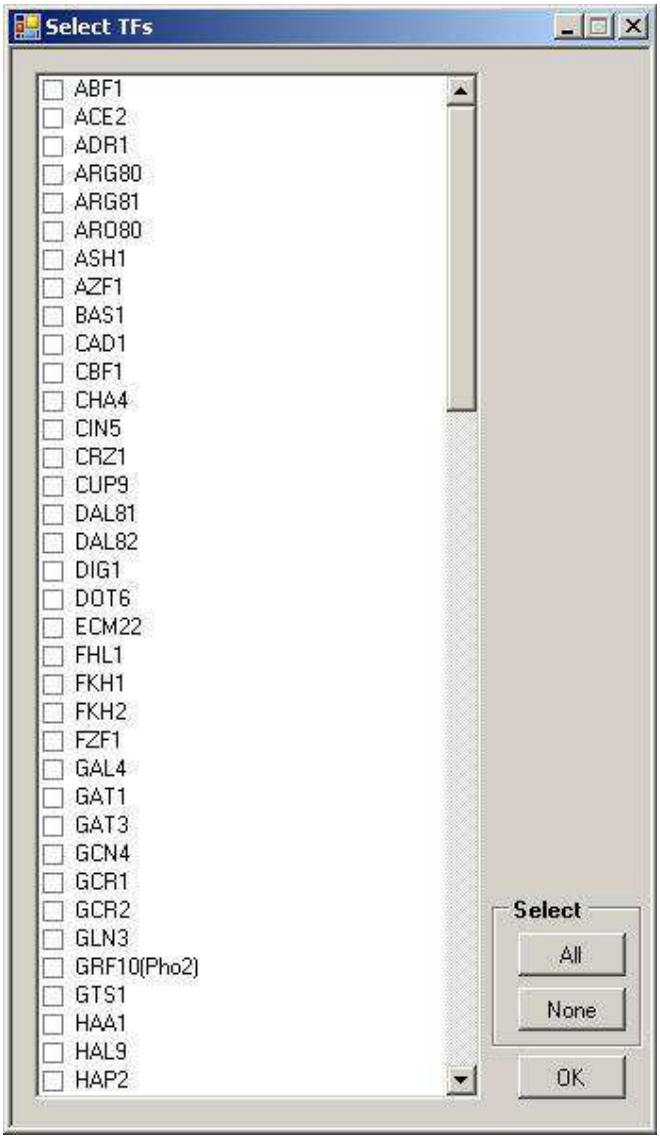


Figure 4.2: TF selection window of TFInfer. User is able to select a subset of TFs available in the connectivity file. TFInfer automatically reduces the regulatory connectivity based on the reduced set of TFs selected by the user.

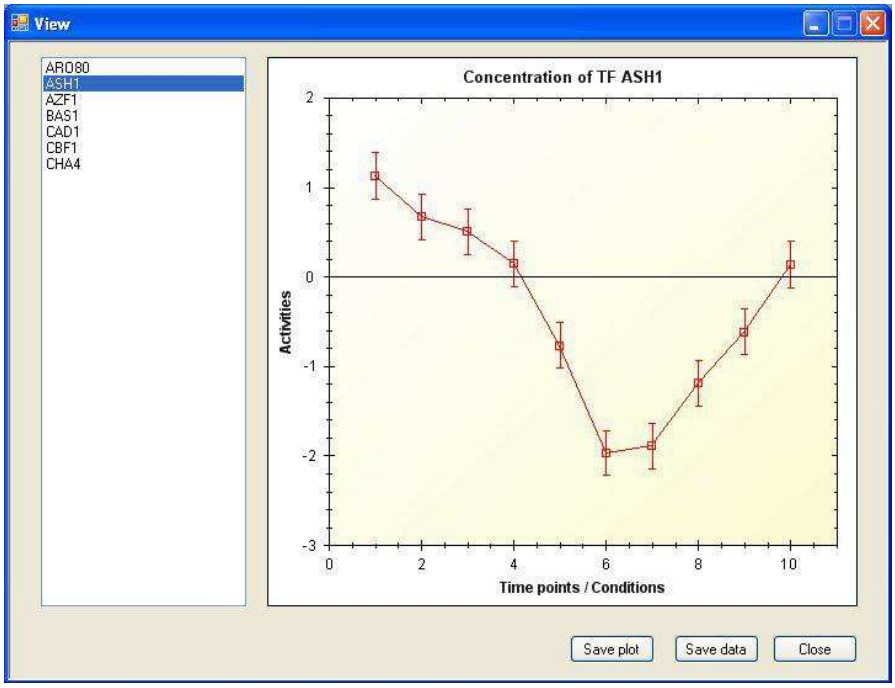


Figure 4.3: Sample results obtained using TFInfer using yeast connectivity and simulated data

to the TF name. This displays a time series activity profile with associated error bars, and by clicking the save plot button the graph can be saved in a variety of formats. An example of the output of TFInfer is given in figure 4.3 (this plot was obtained using synthetically generated data).

4.3.1 Software Features

Main features of the software are summarised below:

- It is open source, and significantly more efficient computationally;
- It is fully documented and has an intuitive Graphical User Interface (GUI);
- It contains template connectivity matrices for *Echerichia coli* and *Saccharomyces cerevisiae*;
- It has been given extra functionalities, handling both time-series data and data from several independent conditions;
- It can handle expression data with multiple biological replicates;
- The results obtained using TFInfer can be saved in different formats such as plot of the concentration profiles or all the results in a comma separate file.

4.3.2 Data Files Format and Software Requirements

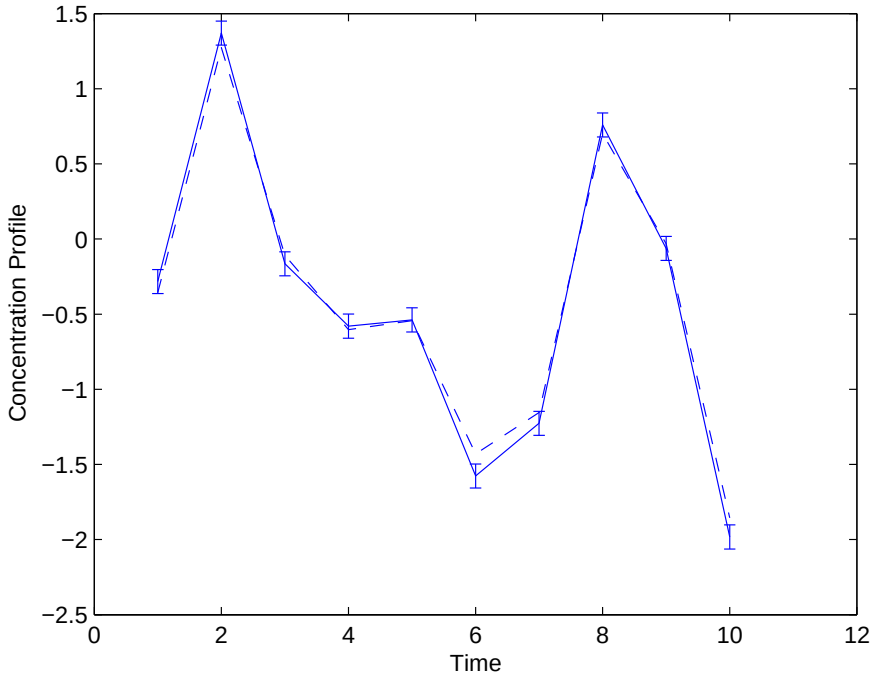
Standard file format for TFInfer is comma separated file. This is a standard format supported by many spreadsheet applications including Microsoft Excel. Two types of input file are required; a csv file containing the logged gene expression data and a file specifying the connectivity matrix (which must be a binary matrix). Replicates are handled by uploading separate data files. For logged gene expression data, the file should contain a list of genes and the corresponding expression levels in different experimental conditions. Connectivity is specified in the form of grid where every entry (zero or one) specifies the connection between the corresponding TF and the gene; the first row of the file will contain the names of the TFs, and the first column the names of the genes. For *S. cerevisiae* and *E.coli*, this connectivity information is supplied as the part of the software; the gene names used are the systematic *b* names for *E. coli* and the ORF identifiers for yeast. The software requires Microsoft .Net framework, which is freely downloadable. It runs on Windows platforms and on Linux/Mac via Mono.

4.4 Comparison of the Two Models

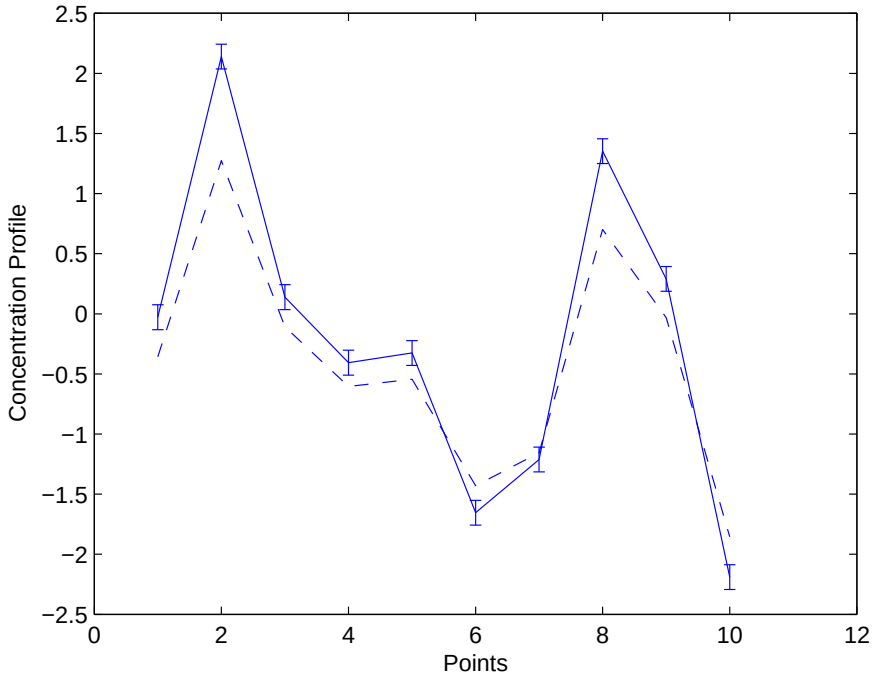
Here, we present some preliminary results comparing the time-dependent model with the time-independent model. This comparison shows that the temporal dynamics, when incorporated in posterior inference, help to reduce the uncertainty of the mean prediction of the our model. We test on a very simple synthetic data set generated using the time-dependent model. We used the time-independent model for simulated gene expression data to infer the transcription factor protein concentration and gene-specific regulatory activities from microarray data. Figure 4.4 shows the comparison of the results for both time-series and time-independent cases using artificial data. From the results, it can be seen that in both cases results are similar with slight differences in confidence intervals associated with the estimated concentration profiles of transcription factor proteins. Another measure would be to compare the ratios of variance of the expected values of a particular transcription factor protein concentration and the associated average error for both times-series and time-independent data. This come out to be 11.9185 for the time series case and 17.8964 for time-independent data. Here, figure 4.4a shows better result as the data used here is taken from a time-series experiment.

4.5 Conclusion

In this chapter, inference of transcription regulation for gene-specific activities is modelled for gene expression data containing different experimental conditions. State space model in vari-



(a)



(b)

Figure 4.4: (a) Estimated concentration profile using time-series model (b) Estimated concentration profile using time-independent model. Dashed line shows the original concentration profile while solid line is the estimated concentration profile.

ational framework is used to provide the basis for inference in transcription networks. Computational complexity is a prominent feature of this model which is better in case of time-independent data. Also, using specific structure of the regulatory network, genome-wide application are possible using time-series and time-independent gene expression data.

While the approach does rely on a simplified model of transcription, the model's results have been shown to capture important physiological effects which have led to the formulation and experimental validation of a number of hypotheses (Davidge et al., 2009; Partridge et al., 2007; Rolfe et al., 2011). Despite these successes, the model was until now only available as working code in MATLAB, requiring expert intervention to be used which resulted in significant bottlenecks in the analysis pipeline. We have now produced a new release which presents several significant advantages over the previous version.

Statistical methods for inferring TF activities are an important area of research in computational biology due to their ability to extract information which is not readily available through standard experimental practice. We believe that the time has arrived for these methods to become standard software used in biological laboratories to complement experimental work, much in the way that sequence alignment tools are now routinely used by experimentalists. By providing a simple yet powerful implementation of an already tried and tested method, we hope TFInfer will become accessible and useful to a wide community of scientists working on gene regulation.

This open-source software is fully documented to aid biologists and requires no software expertise. Full documentation is available at Asif (2010).

Chapter 5

Learning Combinatorial Transcriptional Dynamics from Gene Expression

In chapter 3, we reviewed some of the methods of inference of TFAs from gene expression data. These methods, however, neglect important features of transcriptional regulation; in particular the combinatorial nature of regulation, which is fundamental for signal integration, is not accounted for. Combinatorial regulation implies that the genes in the regulatory network are often regulated by more than one TFs that have a combinatorial control over the expression of genes. The interaction between TFs in regulating the genes is the result of different biological/environmental signals that causes the changes in the expression patterns of genes accordingly. In this chapter, we present a novel method to infer combinatorial regulation of gene expression by multiple transcription factors in large-scale transcriptional regulatory networks. The method implements a factorial hidden Markov model with a non-linear likelihood to represent the interactions between the hidden transcription factors. We explore our model's performance on artificial data sets and demonstrate the applicability of our method on genome-wide scale for three expression data sets. The results obtained using our model are biologically coherent and provide a tool to explore the nature of combinatorial transcriptional regulation.

5.1 Introduction

Understanding the control of gene expression is one of the major goals of systems biology. While gene expression is a complex process with multiple control points, perhaps the most fundamental is the control of mRNA transcription by DNA-binding proteins, transcription factors. A fundamental difficulty in elucidating this process from the experimental point of view is measuring TFAs: TFs are often expressed at low levels, and their activity state is frequently determined by fast post-translational modifications which are difficult to measure directly.

A possible solution to this impasse has arisen due to the availability of experimental tools

to determine the *connectivity* of the transcriptional regulatory network, *i.e.* which TFs bind specific target genes. In particular, the large-scale take-up of ChIP-on-chip techniques has meant that, for model organisms such as yeast and *E.coli*, this connectivity is now available on a high-throughput scale (Lee et al., 2002). As a result, several authors have recently proposed to integrate connectivity and gene expression data in an inference based approach to modelling transcription, whereby TFA is treated as a latent variable to be reconstructed from observations of target gene's expression. Broadly speaking, inferential approaches to TFA reconstruction have used one of two strategies: one approach is to use a very simplistic, typically log-linear model of transcription to infer the activity of a very large number of TFs (Liao et al., 2003; Sabatti and James, 2006; Sanguinetti et al., 2006; Asif et al., 2010). This approach is relatively well established and has already led to several novel insights in biological studies in a range of situations (Partridge et al., 2007; Davidge et al., 2009); however, the simplicity of the models, imposed by the computational constraints of working with large data sets, has meant that important features of transcriptional regulation have been neglected *e.g,* combinatorial regulation. More recently, other authors have focused on inferring TFAs in small sub-networks but employing more realistic models of transcription based on differential equations (Barenco et al., 2006; Lawrence et al., 2006). These approaches are computationally more expensive but allow to model biologically more plausible effects such as saturation (Rogers et al., 2007), rapid transitions (Sanguinetti et al., 2009) and non-linear interactions between TFs (Opper and Sanguinetti, 2010).

In the model proposed in this chapter, we aim at retaining some of the desirable features of small-scale inference approaches in a model capable of learning TFAs on a genome-wide scale. We focus on the problem of modelling interactions between multiple TFs; this is a crucial mechanism that allows cells to integrate signals (Ptashne and Gann, 2002). We present what, to our knowledge, is the first statistical method for reconstructing combinatorial interactions between TFs from target genes' expression levels. We achieve this by modelling TFAs as binary switches (which naturally allow for saturation) within a FHMM with a non-linear emission model which models combinatorial interactions between multiple TFs at a promoter.

We propose a fast structured variational approximation for inference in large scale systems. As our model includes non-linear interaction, it is relatively more parametrised than simpler models. We therefore extensively tested our model on simulated data to check its identifiability. We then applied it to three real time course datasets in *S. cerevisiae* and *E. coli*, using network architectures derived from ChIP-on-chip experiments or curated databases of biological interactions. The key purpose of our analysis of real data is to investigate the extent to which non-linear combinatorial effects are evident from expression data. Perhaps not surpris-

ingly, we find that the length of the time series is a critical factor in reducing the uncertainty of the model's predictions, and thus enabling the recovery of non-linear interactions. Despite this, specific examples of biologically meaningful combinatorial effects are recovered, showing that computational prediction of combinatorial interactions is indeed possible from analysis of mRNA time series.

5.2 A Model for Combinatorial Transcriptional Regulation

Suppose that N genes are regulated by M TFs over T conditions/time points. Throughout this chapter we will assume TFs to be binary variables who can either be on or off (Sanguinetti et al., 2009). This modelling assumption corresponds to two biological assumptions: TFs switch fast from active to inactive form and vice versa, and the number of TF molecules per cell is sufficient to saturate the downstream transcriptional machinery. Let g_i^t be the mRNA expression level of gene i at time t, and let $\{T_j\}_i \quad j \in \mathcal{J}_i \in \{1,\ldots,M\}$ be the set of TFs binding gene i. Our model for (log) gene expression is given by

$$g_i^t \;=\; \mathbf{e}_t \theta_i \;+\; \varepsilon \tag{5.1}$$

where $\theta_\mathbf{i}$ is a set of expression parameters specific for gene i and ε is measurement noise. We construct the vector $\mathbf{e}_t$ by appending all the possible pairwise interactions to the vector of TF states. For two TFs, $\mathbf{e}_t$ is constructed as follows:

$$\mathbf{e}_t \;=\; \begin{bmatrix} T_t^1 \\ T_t^2 \\ T_t^1 T_t^2 \\ 1 \end{bmatrix}.$$

It is important to note that $\mathbf{e}_t$ also encodes the connectivity information of the regulatory network *e.g.*, if the gene i is not connected to T_2, then state of T_2 and T_1T_2 are not included in the construction of the $\mathbf{e}_t$. For each gene i, θ_i contains one coefficient for each TF and for each pairwise interactions followed by the base-level expression b_i

$$\theta_i \;=\; \begin{bmatrix} A_i^1 \\ A_i^2 \\ A_i^{12} \\ b_i \end{bmatrix}$$

so equation 6.1 becomes

$$g_i^t = A_i^1 T_t^1 + A_i^2 T_t^2 + A_i^{12} T_t^1 T_t^2 + b_i + \varepsilon. \tag{5.2}$$

Gene expression is therefore quantised with four expression levels corresponding to the four possible joint states of the two regulators. This can be viewed as a steady-state approximation to the combinatorial transcription model of Opper and Sanguinetti (2010). The assumption of binary states of the TFs is mainly due to the transient behaviour of these regulators that makes it harder to measure experimentally at the sampling rate used in most of the cases.

To cast the model (6.1) in a Bayesian framework we need to specify prior distributions over the various components. The prior for the parameters θ_i is assumed to be a zero mean Gaussian with variance encoded by a hyperparameter α^2,

$$\theta_i \sim \mathcal{N}(0, \alpha^2 \mathbf{I}).$$

The choice of prior over the TFA is dictated by the experiment we are modelling. If the experimental design consists of a number of independent conditions, then a uniform prior over the TF states at each condition may be justified. While this experimental design is indeed very widely used, in this chapter we will focus on the time-course experimental design. The derivations for independent conditions experimental design are given in appendix A. In the time-course experimental design, the natural prior distribution for the TFA is given by a FHMM (Ghahramani and Jordan, 1997). Therefore, the prior probability defines a series of *a priori* independent Markov chains consisting of sequences of binary states, one for each TF

$$p\left(T_1^j, \ldots, T_T^j\right) = \prod_{t=1}^{T} p\left(T_{t+1}^j | T_t^j, \tau_j\right).$$

Each of these Markov chains depends on a matrix of hyperparameters, the *transition probabilities*, encoding the prior probability of the TF switching from active to inactive form. As the TFs are assumed to be binary, by normalisation there are only two independent hyperparameters in each transition matrix. Finally, the model is completely specified by the assumption that the observation error in equation (6.1) is zero mean Gaussian and i.i.d., so that

$$p(G|T, \Theta) = \prod_{i=1}^{N} \prod_{t=1}^{T} \mathcal{N}\left(g_i^t | \mathbf{e}_t \theta_i, \sigma^2\right)$$

here G, T and Θ are collective names for all the observations, TF states and gene specific parameters respectively.

Transition probabilities (τ_j) for transcription factors are selected such that the transitions between the on and the off states of transcription factors are not very frequent. This initialisation scheme also represent the underlying biological understanding. Other hyper-parameters (α, σ) are fixed based on the empirical analysis on different datasets.

Before discussing how inference can be performed in this model, it is important to observe that, as the parameters Θ and the TF states T only appear in the model (6.1) through their product, a basic identifiability problem exists for this model. To clarify the issue, if we take the simple case of a gene regulated by two TFs, we see that equation (5.2) is left invariant by the transformation

$$\begin{aligned} &T_t^1 \to 1 - T_t^1 \forall t \in \{1, \ldots, T\} \\ &b' \to A_1 + b, \quad A_1' \to -A_1 \\ &A_2' \to A_2 + A_{12}, \quad A_{12}' \to -A_{12}. \end{aligned} \tag{5.3}$$

This ambiguity, which is common to all statistical models involving multiplication of latent variables, cannot be resolved without prior knowledge. This is often available: for example, it may be known that a given TF activates/represses a specific target, or that the TF is on/off in a specific condition. Notice that knowledge about the sign of regulation for a *single* target gene or for a *single* condition/time point is sufficient to remove the ambiguity for all other conditions/targets of the same TF. Another important observation is that the presence or absence of a combinatorial interaction is not affected by the identifiability problem. Only the sign of the combinatorial term A_{12} changes under the transformation (6.3).

5.3 Inference in Combinatorial Factorial Hidden Markov Model

Our goal is to infer from observations of gene expression both the state of TFs and the gene-specific expression parameters θ. Bayesian inference in model (6.1) is analytically intractable so we resort to approximation techniques. The following sections provide the details of inference in the proposed model using Gibbs sampling and variational inference.

In appendix (A), we provide the details of the inference for the static case of the model where the expression data is not from a time-series microarray experiment.

5.3.1 Inference with Gibbs Sampling with Time Dynamics

Gibbs sampling is a Markov Chain Monte Carlo sampling algorithm which involves sequential sampling from the conditional posterior distribution of a latent variable given all other variables and the observations. Figure 5.1 shows the graphical model of the method presented in this

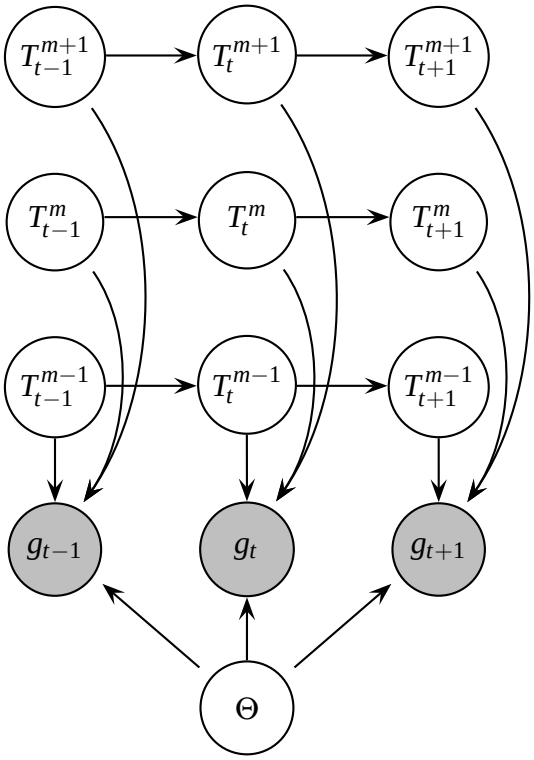


Figure 5.1: Graphical representation of the model. TF states are modelled as *a priori* independent first order Markov chains that influence the expression of gene i; pairwise interaction between all the regulators of gene i are also contributing to the changes in the expression levels of gene i. Θ is the set of gene-specific parameters that encode the strength with each a particular TF is influencing the gene i.

chapter. This model is a variant of FHMM where the pairwise interactions of latent states of Markov chains (*e.g.* $T_t^1 T_t^2$) are also effecting the observed variable (g_i^t) along with the latent states of Markov chains (*e.g.* T_t^1). We refer this form of FHMM as combinatorial factorial hidden Markov model (cFHMM) in the remainder of this chapter.

By general results on inference in graphical models (Bishop, 2006), each node is conditionally independent of all other nodes given its Markov blanket, which is defined as the set of parents, set of children and parents of its children. Using this information, the conditional posterior distribution for each TF at each time point can be written as

$$P(T_t^m|\Phi) = \frac{P(T_t^m|T_{t-1}^m)P(T_{t+1}^m|T_t^m)P(g_t^i|T_t^i)}{\sum_{T_t^m} P(T_t^m|T_{t-1}^m)P(T_{t+1}^m|T_t^m)P(g_t^i|T_t^i)} \tag{5.4}$$

where $\Phi = \{T_{t-1}^m, T_{t+1}^m, T_t^{'m}, g_t^i, \theta_i, \mathbf{X}\}$.

The conditional posterior distribution for θ_i given the TF states and observations is a multivariate Gaussian and given by

$$p(\theta_\mathbf{i}|g_i^t, TF_t, \mathbf{X}) = \frac{\prod_{t=1}^{T} \mathcal{N}(g_i^t|\mu_i(TF^t), \sigma^2).p(\theta_\mathbf{i}|\alpha^2)}{\sum_{\theta_\mathbf{i}} \left[\prod_{t=1}^{T} \mathcal{N}(g_i^t|\mu_i(TF^t), \sigma^2).p(\theta_\mathbf{i}|\alpha^2)\right]} \tag{5.5}$$

The sampling algorithm iterates sampling from each of these conditionals. Convergence of the chains can be monitored using standard heuristics (Gelman et al., 2004) and, depending on the scale of the problem, is usually achieved after a few thousand burn-in cycles.

5.3.2 Inference with Variational Bayesian Expectation Maximisation Algorithm with Time Dynamics

Stochastic inference approaches such as Gibbs sampling are often employed for analytically intractable models; unfortunately, we found that the computational costs of such an approach were too high for large scale problems. We therefore develop a fast structured mean-field approximation which is capable of performing inference in very large-scale problems.

Variational Bayesian inference is an optimisation-based approximate inference technique originally developed in statistical physics. The basic idea is to approximate the posterior distribution over the latent variables and parameters with a simpler distribution. Variational techniques convert a complex problem into a simpler problem by decoupling the degrees of freedom in the original problem (Jordan et al., 1999). This decoupling is obtained by expanding the problem to include additional parameters also know as variational parameters that are optimised

according to the problem under consideration. Compared with stochastic approximations like Gibbs sampling, this optimisation process is usually very efficient computationally, and has the advantage of allowing an unambiguous monitoring of convergence.

Variational inference relies on the following general lower bound on the log likelihood:

$$log\,[p(\mathbf{G}|\phi)] \;\geq\; \langle log\; p(\mathbf{G},\Theta,\mathbf{T}|\phi)\rangle_{q(\Theta,\mathbf{T})} + H(q) \tag{5.6}$$

which follows from Jensen's inequality (Bishop, 2006). Here $\langle\rangle$ shows the expectation of the joint likelihood under the approximating distribution q, H denotes the entropy of the distribution and ϕ collectively denote the hyperparameter α and σ. It can be shown that the lower bound (5.6) is saturated if and only if the approximating distribution q is equal to the posterior distribution $p(\Theta,\mathbf{T}|\mathbf{G},\phi)$. In our case, the approximating distribution q is assumed to be a structured mean-field approximation

$$q(\Theta,\mathbf{T}) \;=\; q(\Theta)\prod_m q(\mathbf{T}^m). \tag{5.7}$$

Therefore, we assume the approximating distribution to factor across parameters and transcription factors, but *not* across time points. The joint likelihood of the model is given by

$$p(\mathbf{G},\Theta,\mathbf{T}) = p(\mathbf{G}|\mathbf{T},\Theta)\; p(\Theta|\alpha^2)\; p(\mathbf{T}) \tag{5.8}$$

We will use a variational EM algorithm to optimise iteratively the lower bound w.r.t. Θ and each of the TFs $\mathbf{T}^i$; the reader is referred to (Beal, 2003) for a more thorough discussion of variational EM algorithms in HMMs. The lower bound (5.6) is guaranteed to increase after each step of this iterative process, and the convergence of the algorithm can be monitored through evaluation of the lower bound.

5.3.2.1 *E-step*

The log of the joint likelihood can be written as

$$\begin{aligned}\log\; p(\mathbf{G},\mathbf{T},\Theta,\alpha,\sigma) \;&=\; \sum_{t=1}^{T}\left[\sum_{m=1}^{M}\log p(T_t^m|T_{t-1}^m) + \sum_{i=1}^{N}\left\{-\frac{1}{2\sigma^2}\left(g_i^t - \mathbf{e}_t^T\theta_i\right)^2\right\}\right] \\ &\quad + \sum_{i=1}^{N}\left(-\frac{1}{2\alpha^2}\theta_i^T\theta_i\right) + \text{const.}\end{aligned} \tag{5.9}$$

Taking the expectation of the log of the joint likelihood with respect to $q(\theta_i)$ and $q(\mathbf{T}_{1:T}^{j\neq m})$, we get

$$\begin{aligned}\langle \log p(\mathbf{G},\mathbf{T},\Theta,\alpha,\sigma)\rangle_{q(\theta_\mathbf{i})q(\mathbf{T}_{1:T}^{j\neq m})} &= \sum_{t=1}^{T}\Big[\log p(T_t^m|T_{t-1}^m) \\ &+\sum_{i=1}^{N}\left\{\frac{1}{\sigma^2}\left(g_i^t\left\langle \mathbf{e}_t^T\right\rangle_{q(\mathbf{T}_{1:T}^{j\neq m})}\langle\theta_i\rangle_{q(\theta_i)}\right.\right. \\ &\left.\left.-\frac{1}{2}\left\langle \mathbf{e}_t^T\right\rangle_{q(\mathbf{T}_{1:T}^{j\neq m})}\left\langle \theta_i\theta_i^T\right\rangle_{q(\theta_i)}\left\langle \mathbf{e}_t\right\rangle_{q(\mathbf{T}_{1:T}^{j\neq m})}\right)\right\}\Big] \\ &+ \text{ const.}\end{aligned} \tag{5.10}$$

As we averaged out the θ_i and all other TFs (*i.e.* $T_t^{j\neq m}$) we are left with an expression depending only on the m^{th} TF. A closer inspection of the previous equation shows that (up to a constant) it is the log of the joint distribution of an HMM with transition probabilities given by $p(T_t^m|T_{t-1}^m)$. The emission probabilities are Gaussian with time-dependent mean and variance; their logarithm up to a constant is given by

$$\sum_{i=1}^{N}\left\{\frac{1}{\sigma^2}\left(g_i^t\left\langle \mathbf{e}_t^T\right\rangle_{q(\mathbf{T}_{1:T}^{j\neq m})}\langle\theta_i\rangle_{q(\theta_i)}-\frac{1}{2}\left\langle \mathbf{e}_t^T\right\rangle_{q(\mathbf{T}_{1:T}^{j\neq m})}\left\langle \theta_i\theta_i^T\right\rangle_{q(\theta_i)}\left\langle \mathbf{e}_t\right\rangle_{q(\mathbf{T}_{1:T}^{j\neq m})}\right)\right\}, \tag{5.11}$$

we can read off the expression 5.11 the actual emission probabilities at time t as

$$\log\prod_{i=1}^{N}p(g_i^t|\mathbf{T}_t^m)=\log\left[\prod_{i=1}^{N}\mathcal{N}\left(g_i^t|\langle\mathbf{e}_t\rangle\langle\theta_i\rangle,\sigma^2\right)\right], \tag{5.12}$$

so equation 5.10 simplifies to

$$\langle \log p(\mathbf{G},\mathbf{T},\Theta,\alpha,\sigma)\rangle_{q(\theta_\mathbf{i})q(\mathbf{T}_{1:T}^{j\neq m})}=\sum_{t=1}^{T}\left[\log p(T_t^m|T_{t-1}^m)+\sum_{i=1}^{N}\log\mathcal{N}\left(g_i^t|\langle\mathbf{e}_t\rangle\langle\theta_i\rangle,\sigma^2\right)\right] \tag{5.13}$$

which gives the transition probabilities and time-dependent emission probabilities of the HMM with m^{th} TF. The posterior distribution over each TF can be easily obtained using the standard *forward backward* (FB) algorithm (section 2.2.2.1) that provides the probabilities for both states (*i.e.* on or off) of TFs over all the time point of the gene expression measurements. Further using the factorisation across TFs given in equation (5.7), we use the FB algorithm independently for each hidden layer of FHMM (Fig. 5.1) to provide the single time state marginals of the approximate posterior distribution $q(\mathbf{T})$.

5.3.2.2 *M-step*

Taking expectations of the log of the joint likelihood (equation (5.9)) under $\mathbf{T}$, one can see that the approximate posterior distribution over the parameters of Θ_i is given by a multivariate normal

$$q(\Theta)=\prod_{i=1}^{N}\mathcal{N}\left(\theta_i|\mathbf{m}_\mathbf{i},\Sigma_i\right) \tag{5.14}$$

The mean and covariance of this multivariate normal distribution are given by

$$\Sigma_i^{-1} = \frac{1}{\sigma^2} \sum_{t=1}^{T} \mathbf{X}_i \left\langle \mathbf{e}_t \mathbf{e}_t^T \right\rangle_{q(\mathbf{T})} \mathbf{X}_i + \alpha^{-2} \mathbf{I}$$

$$\mathbf{m_i} = \frac{1}{\sigma^2} \left[\sum_{t=1}^{T} g_i^t \left\langle \mathbf{e}_t^T \right\rangle_{q(\mathbf{T})} \mathbf{X}_i \right] \Sigma_i^{-1}$$

Here $\langle\rangle_{q(\mathbf{T})}$ denotes the expectation under $q(\mathbf{T})$, $\mathbf{X}_i$ denotes a diagonal matrix with the i-th row of the connectivity matrix $\mathbf{X}$ along the diagonal. For more details about the method and implementation, refer to supplementary material.

As the length of the time series is usually very limited, we will not attempt to infer hyperparameters of the model such as the transition matrices and observation noise variance (even if point estimation of hyperparameters by Type II maximum likelihood is in principle straightforward). In chapter 6, we propose a simultaneous inference and clustering technique for transcriptional regulation that provides a possible solution for inferring the transition rates of the latent Markov chains with few time-points . In this model, these hyperparameters will be fixed heuristically: transition matrices will be set to give a prior expectation of few transitions within the time under consideration; and noise variance will be fixed after preliminary inspection of the data. Experiments on synthetic data showed that the model predictions to be fairly insensitive to the specific values of the transition matrices.

Using the EM algorithm, we iteratively update the posterior distributions for model parameters (Θ) and latent variables ($\mathbf{T}^m s$ encoded in $\mathbf{e}_t$) until the model is deemed to converge. This convergence can be monitored by evaluating the Eq. 5.9 of the likelihood of the model which is guaranteed to decrease. It is shown in figure 5.3 for a small simulated dataset ($N = 100$, $M = 15$). This process of iterative optimisation using EM algorithm is illustrated in algorithm 3.

5.4 Comparison of Approximation with Gibbs Sampling and Variational Inference

To evaluate the approximation of VBEM algorithm and Gibbs Sampling, we ran the VBEM on a smaller dataset consisting of 400 genes and 20 TFs over 20 time-points, and compared the results with those obtained using the Gibbs sampler derived in Section 5.3.1. We monitored convergence of the Gibbs sampler by mixing of the Markov chains of Θ parameters. The Gibbs sampler took almost a day to converge compared to less than an hour with variational EM. In general, both methods obtained very similar results, both in terms of mean predictions and

Algorithm 3 Variational Bayesian Expectation Maximisation Algorithm for inference in cFHMM

Require: Initialise $\mathbf{e}_{1:T}$ randomly or from expression data ($\mathbf{G}$)

Require: $\alpha^2 \leftarrow 1$

Require: $\sigma^2 \leftarrow 0.1$

Require: Initialise transition probabilities (τ) for all the TFs

$nIterations \leftarrow 1$

1: **repeat**

2: **for** $i \leftarrow 1, N$ **do**

3: Update the posterior distribution (Eq. 5.14) over θ_i for gene i

4: **end for**

5: **for** $j \leftarrow 1, M$ **do**

6: Calculate the state marginal of $T^j_{1:T}$ using FB algorithm

7: **end for**

8: Update $\mathbf{e}_{1:T}$ using the state marginals

9: Calculate the $NewLikelihood$ using the expected values of the latent variables and parameters in Eq. 5.9

Ensure: $NewLikelihood < OldLikelihood$

10: $OldLikelihood \leftarrow NewLikelihood$

11: $maxIterations \leftarrow maxIteration - 1$

12: **until** $(nIterations > 2000) \vee (NewLikelihood - OldLikelihood > 1e^{-4})$

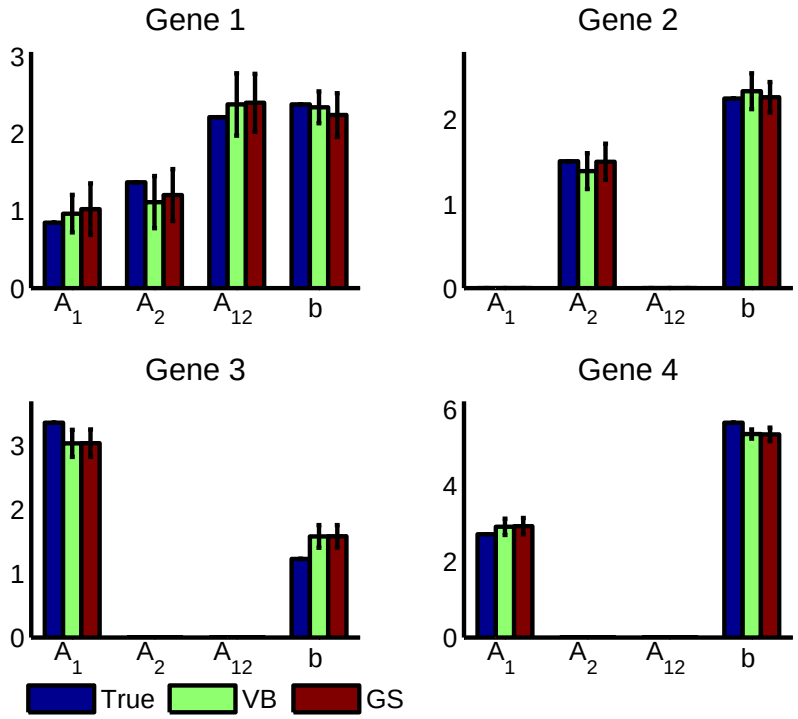


Figure 5.2: Comparison of inferred parameter using variational Bayesian inference and Gibbs sampling for four randomly selected genes. Blue bars shows the ground truth while the green and red bars shows the inferred values of the parameters (A_1, A_2, A_{12}, b) using variational Bayesian (VB) inference and Gibbs sampling (GS) respectively. Empty spaces in the plots correspond to TF not bound to target gene.

in terms of associated uncertainties. Figure 5.2 shows a comparison of the inferred values of four randomly selected genes using variational Bayesian inference and Gibbs Sampling with the true values. Variational inference are known to often underestimate uncertainties; a global comparison between MCMC and variational estimates of the uncertainties indicates that in our case this is a fairly modest effect (correlation coefficient 0.8614 at p-value of 0.0003). Due to higher computational complexity of Gibbs sampling, we, therefore, employ the variational approximation for approximating the true posterior distribution in the rest of this chapter.

5.5 Analysis using Variational Bayesian Expectation Maximisation Algorithm

While our model is still relatively simple, the addition of non-linear interaction terms means that more parameters need to be estimated. On top of that, asymptotically exact inference is computationally unfeasible in large scale examples. Therefore, as a first analysis we perform a thorough test on the proposed model using artificial data to verify its identifiability in a realistic

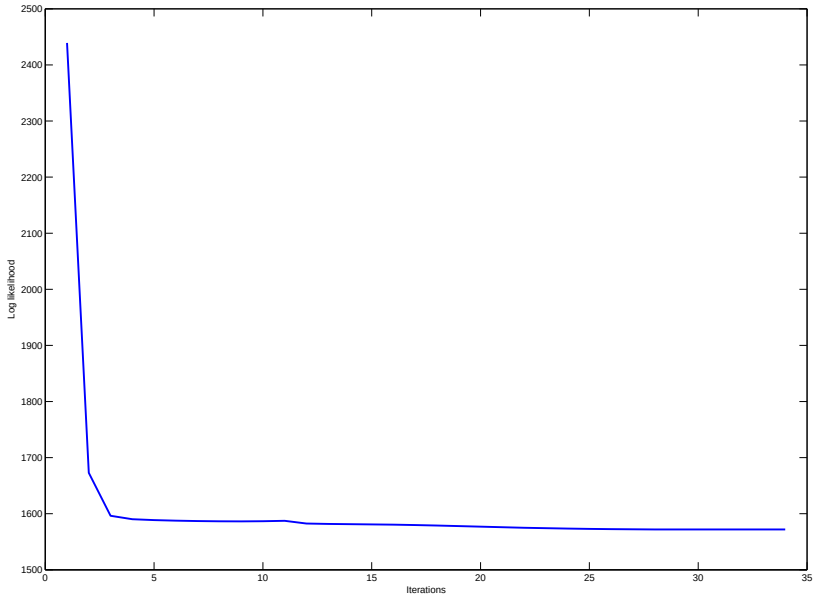


Figure 5.3: Convergence of VBEM algorithm on small simulated dataset ($N = 100$, $M = 15$).

simulated situation.

5.5.1 Analysis using Synthetic Data

We performed a series of experiments on artificial data generated with known parameters to benchmark and check the consistency of the model. Specifically, three aspects of the inferential problem need to be investigated:

1. Is the model identifiable given realistic data, *i.e.* in a large scale example with relatively few time points?

2. Does the efficient variational approximation developed in section 5.3.2 give an accurate representation of the posterior uncertainty over the random variables?

3. How does the length of the time series effect the inference of combinatorial interactions of TFs at a certain noise level?

5.5.1.1 Model Identifiability

In this sections, we present a brief comparison of cFHMM two other methods on synthetic data. Shi et al. (2008) used FHMMs with inputs to simultaneously infer TFAs and post-transcriptional regulation in TFs; in our case, we are interested only in the TF inference part

of the model, so that their model reduces to a simplified form of our model (*i.e.* a standard FHMM) without the non-linear interactions of TFs. This method is denoted as FHMM. The other method we compare to is the TFInfer model (chapter 4). This is a log-linear model using a discrete-time state space model for the TFAs. To compare the binary TF states obtained with the other two methods with the TFInfer results, we binarize the inferred TFAs using the average of the inferred temporal profile of each TF in the network (activity 0 if below average, 1 if above). We use three criteria to evaluate the performance of our method with these methods; run-time, mean squared error (MSE) in reconstructing gene expression profiles and the Hamming distance between the inferred states of the TFs.

Synthetic data was generated using the cFHMM model with two different connectivities that we take from the yeast regulatory network (Lee et al., 2002) and E.coli regulatory network[1]. Using yeast connectivity, three synthetic datasets with M TFs were generated (30 time-points $M = 25, 30, 50$). Another synthetic dataset was generated using the E.coli connectivity with 30 time points and 6 transcription factors. Results obtained using these datasets are represented in the table 5.1 where the comparison of different techniques is shown.

It is important to stress that these two connectivities have different degree of sparsity. In yeast connectivity data, average connectivity is $2-4\%$ for three datasets while in case of E.coli dataset, average connectivity is about 20%. Average number of genes/TF in three yeast datasets are 11.5, 14.2 and 22.7 respectively; while in case of E.coli connectivity, average number of genes/TF is 60.1 that implies more potential combinatorial interactions between TFs.

The Hamming distance between the inferred temporal profiles of TFs (obtained using FHMM, cFHMM and TFInfer) and the true ones are comparable in all four datasets. It is important to mention here an aspect of the TFInfer inference procedure; the optimisation of the hyper-parameters that we keep fixed in our model and in FHMM.

In case of sparse yeast connectivity, FHMM and cFHMM are closely related in terms of the Hamming distance between the inferred temporal profiles and the true profiles in the synthetic data. This is mainly due to the sparse connectivity that implies less combinatorial interactions and hence cFHMM results closely match with FHMM results in terms of MSEs and Hamming distances. This can be seen in the first three columns of table 5.1.

The last column of the table 5.1 shows the results of the experiment with a much dense connectivity (average connectivity is 20%) where cFHMM is better at reconstructing the expression profiles of genes in the network (MSE=0.0099) compared to MSE of FHMM (MSE=0.0187).

[1] `http://ecocyc.org/`

Method \ Dataset	**Yeast Connectivity** $N = 500, M = 25$	**Yeast Connectivity** $N = 500, M = 50$	**Yeast Connectivity** $N = 500, M = 75$	**E.coli Connectivity** $N = 320, M = 6$
FHMM	MSE: 0.0967 HD with True=0.0820 HD with cFHMM=0.0300 HD with TFInfer=0.0880	MSE: 0.1012 HD with True=0.1380 HD with cFHMM=0.0880 HD with TFInfer=0.1340	MSE: 0.1258 HD with True=0.1933 HD with cFHMM=0.1273 HD with TFInfer=0.1993	MSE: 0.0187 HD with True=0.0625 HD with cFHMM=0.0375 HD with TFInfer=0.0750
cFHMM	MSE: 0.0931 HD with True=0.0600 HD with FHMM=0.0300 HD with TFInfer=0.0740	MSE: 0.1065 HD with True=0.1380 HD with FHMM=0.0880 HD with TFInfer=0.1420	MSE: 0.1184 HD with True=0.2167 HD with FHMM=0.1273 HD with TFInfer=0.2173	MSE: 0.0099 HD with True=0.0667 HD with FHMM=0.0375 HD with TFInfer=0.0542
TFInfer (Asif et al., 2010)	MSE: 0.0910 HD with True=0.0780 HD with FHMM=0.0880 HD with cFHMM=0.0740	MSE: 0.0894 HD with True=0.1280 HD with FHMM=0.1340 HD with cFHMM=0.1420	MSE: 0.0858 HD with True=0.1687 HD with FHMM=0.1993 HD with cFHMM=0.2173	MSE: 0.0150 HD with True=0.0792 HD with FHMM=0.0750 HD with cFHMM=0.0542

Table 5.1: Comparison of different techniques for inference of the states of transcription factors using simulated data. The states inferred with different methods are compared using the Hamming distance (HD) between the vectors of states.

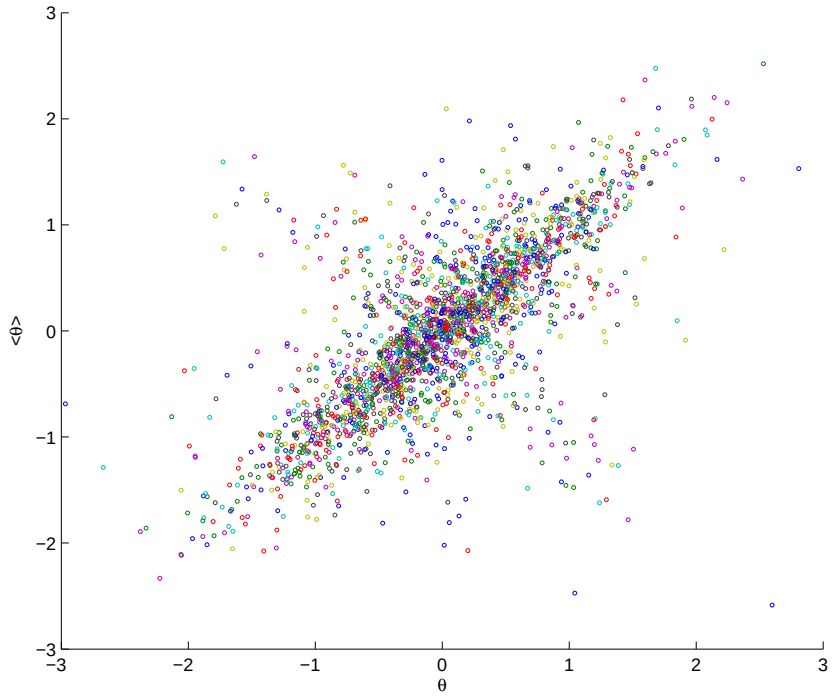


Figure 5.4: Comparison of inferred and true values for parameter Θ

5.5.1.2 Accuracy of the Posterior Estimation

We generated an artificial dataset with 1000 genes, 50 transcription factors and 20 time-points. We used the connectivity information from yeast cell regulatory network (Lee et al., 2002) with random initialisation for the gene-specific parameters. We then ran the variational EM algorithm to infer the posterior probabilities over TF states and gene specific parameters, and compared with the true parameter values/ TF states. The results for parameter estimation are given in figure 5.4, displaying true parameter values with posterior mean estimates. In most cases, it is clear that the parameters inferred using the variational EM algorithm match closely with the true values. In a few cases, the inferred parameters are anticorrelated with the true parameter values; these correspond to TFs whose activity was inferred to be the opposite of the true activity. As we noted earlier, this ambiguity is unavoidable and cannot be resolved without further knowledge.

5.5.1.3 Effects of the Length of the Time-series and Noise in Gene expression

While Figure 5.4 gives support to the identifiability of the *mean* predictions of our model, the Bayesian nature of the model means that estimates of the uncertainty of the predictions are also available. These estimates can be precious to assess the statistical significance of

T	$\sigma^2 = 0.1$		$\sigma^2 = 0.5$		$\sigma^2 = 1$	
	$A_{ij}(\%)$	Avg. Post. s.d.	$A_{ij}(\%)$	Avg. Post. s.d.	$A_{ij}(\%)$	Avg. Post. s.d.
10	18	0.2273	5	0.4009	3	0.5027
20	28	0.1655	10	0.3016	6	0.3953
30	40	0.1342	25	0.2550	8	0.3364
40	54	0.1088	33	0.2248	18	0.2993
50	54	0.0996	33	0.2003	18	0.2710

Table 5.2: Combinatorial interactions found using synthetic data with different number of time-points. A_{ij} is the percentage of combinatorial interactions recovered from the data. σ^2 stands for the noise-level in the synthetics data. Column 3, 5 and 7 shows the corresponding inferred average posterior standard deviation for each dataset.

predicted interactions: for example, we could say that two TFs regulate combinatorially a certain gene at 5% significance level if the absolute value of the posterior mean of the predicted combinatorial term in equation (5.2) is greater than twice the predicted standard deviation. We are interested in quantifying what fraction of combinatorial interactions can be recovered at a certain significance level as a function of the length of the time series and the experimental noise. To do this, we generated multiple artificial data sets with different numbers of time-points (Table 5.2, column 1) and varying corrupting noise levels ($\sigma^2 = 0.1, 0.5, 1.0$). In all cases the number of genes and transcription factors, as well as the network architecture and true parameter values, was kept fixed ($N = 200, M = 50$). It is important to note that these datasets are generated with Θ as zero mean Gaussians (with unit variance) so all the combinatorial terms used to generate the datasets are nonzero. Table 5.2 reports the fraction of combinatorial regulatory interactions which were recovered at 5% significance level for specific lengths of the time series and different values of the Gaussian noise in gene expression. Not surprisingly, this percentage increases monotonically with the length of the time series and decreases when the additive observation noise is increased. Also, it appears that the level of noise somehow determines the proportion of combinatorial interactions that can be recovered *even for long* time series. Empirically, it appears that, with this network structure, more than 40 time points do not lead to a significant change in the proportion of combinatorial interactions recovered.

5.5.2 Analysis using Real Data

We use three real datasets; in all cases, the main purpose is to probe the extent to which combinatorial regulations can be learned from expression data. These datasets are the classic and much studied yeast cell cycle data set (Spellman et al., 1998), the yeast metabolic cycle data set (Tu et al., 2005) and the *E. coli* micro-aerobic shift data set (Partridge et al., 2007). Finally,

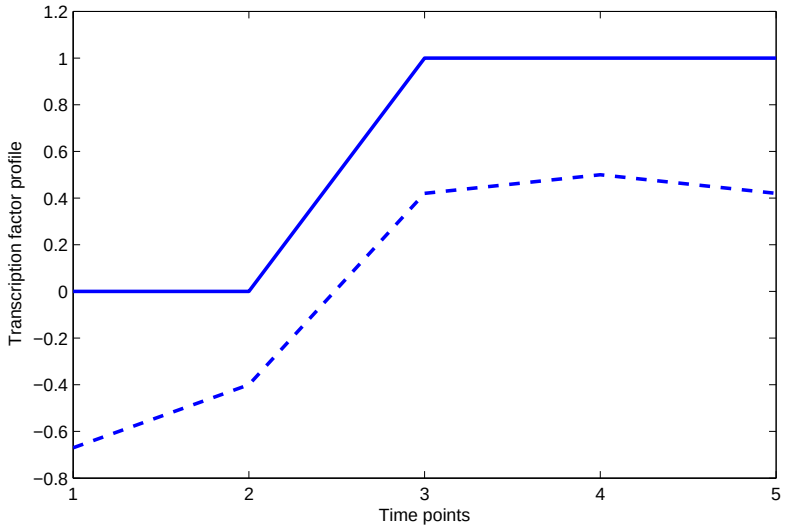


Figure 5.5: Comparison of the inferred temporal profiles of transcription factor ArcA using TFinfer (Asif et al., 2010) and cFHMM. The dotted line in the plot shows the profile of ArcA inferred using Sanguinetti et al. (2006).

we compare our results with those obtained with two different methods: a standard FHMM and the TFInfer (Sanguinetti et al., 2006; Asif et al., 2010).

5.5.2.1 Micro-aerobic Shift in *E. coli*

Partridge et al. (2007) studied the transcriptomic response of *E.coli* to the withdrawal of oxygen in a chemostat culture under controlled growth conditions. *E.coli* is a metabolically versatile bacterium and responds to changes from aerobic to micro-aerobic conditions by activating TF proteins that act as oxygen sensors. The probabilistic approach described in Sanguinetti et al. (2006) was used to infer the states of six crucial regulators of oxygen sensing and metabolism (FNR, MetE, MetJ,ArcA, CpxR,SigE) from the mRNA expression of 302 target genes. The analysis revealed insights in the dynamics of the key regulators upon oxygen withdrawal, as well as biologically interesting predictions about the timing of TFA. The data set consists of 4 time points taken at 5, 10, 15 and 60 minutes and measured relative to a sample taken immediately before the perturbation. Connectivity information about the regulatory network was obtained from the ecocyc database[2] and is available for 6 TFs and 302 genes in the supplementary material of Partridge et al. (2007).

The predictions of our model in terms of TFAs are in broad agreement with what reported in Partridge et al. (2007) (average Pearson correlation 0.9). As an example, Figure 5.5 shows the inferred temporal profile of transcription factor ArcA to be in close agreement with the pre-

[2] `http://ecocyc.org/`

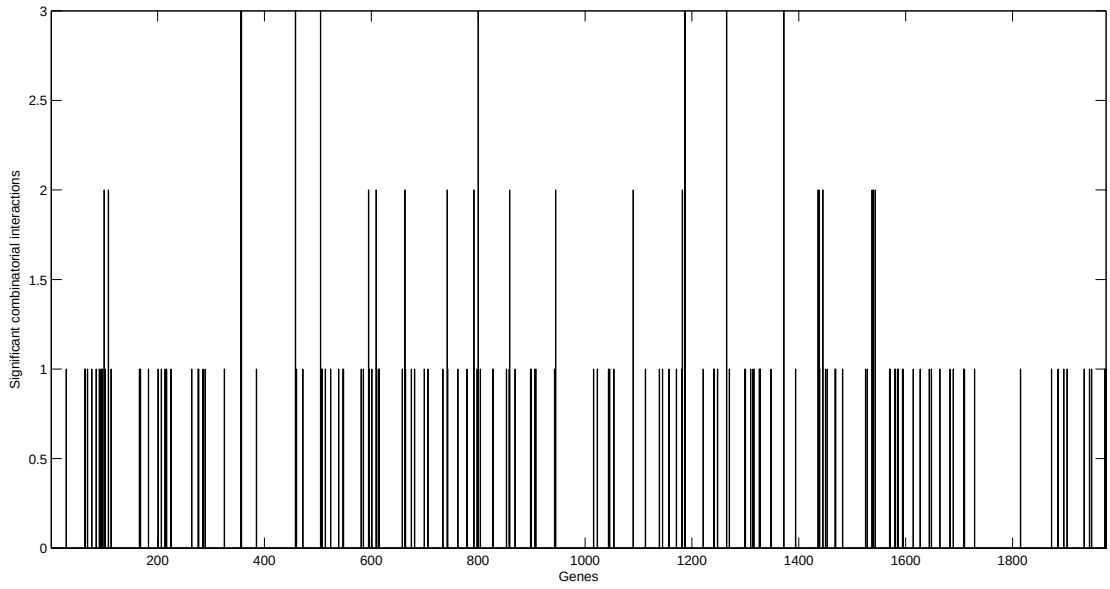


Figure 5.6: Number of $A_{ij} \geq 2$ $s.d.$ for 1975 genes of Spellman et al. (1998)

dicted profile in Partridge et al. (2007). However, no combinatorial interactions were predicted at a significance level of 5%. In the light of the analysis on synthetic data, this is probably due to the very short time series.

5.5.2.2 Yeast Cell-cycle Data

Spellman et al. (1998) used microarray hybridization to measure the expression profiles of most of the yeast genes over a complete cell cycle. Three time-series experiments were conducted on three different strains of yeast and these experiments were synchronised by three independent methods; α factor-based synchronization, size-based synchronization and cdc15-based synchronization. We use the cdc15 synchronized data, consisting of 6181 gene expression profiles over 24 time-points. The connectivity information for the yeast regulatory network was obtained in Lee et al. (2002) using ChIP-on-chip for 113 TFs measuring their binding to 6270 genes. These two datasets are relatively old but well studied and serve as the standard benchmark for validating the model described here. We preprocessed these two datasets such that all the genes are bound by at least one TF and each TF is regulating at least one gene; that gave us a network of 1975 genes and 104 TFs and expression profiles of 1975 genes. The data was analysed using the variational approximation, since the large size of this network rules out the application of the Gibbs sampling algorithm.

Once again, the predictions in terms of TFAs matched well the predictions of previous mod-

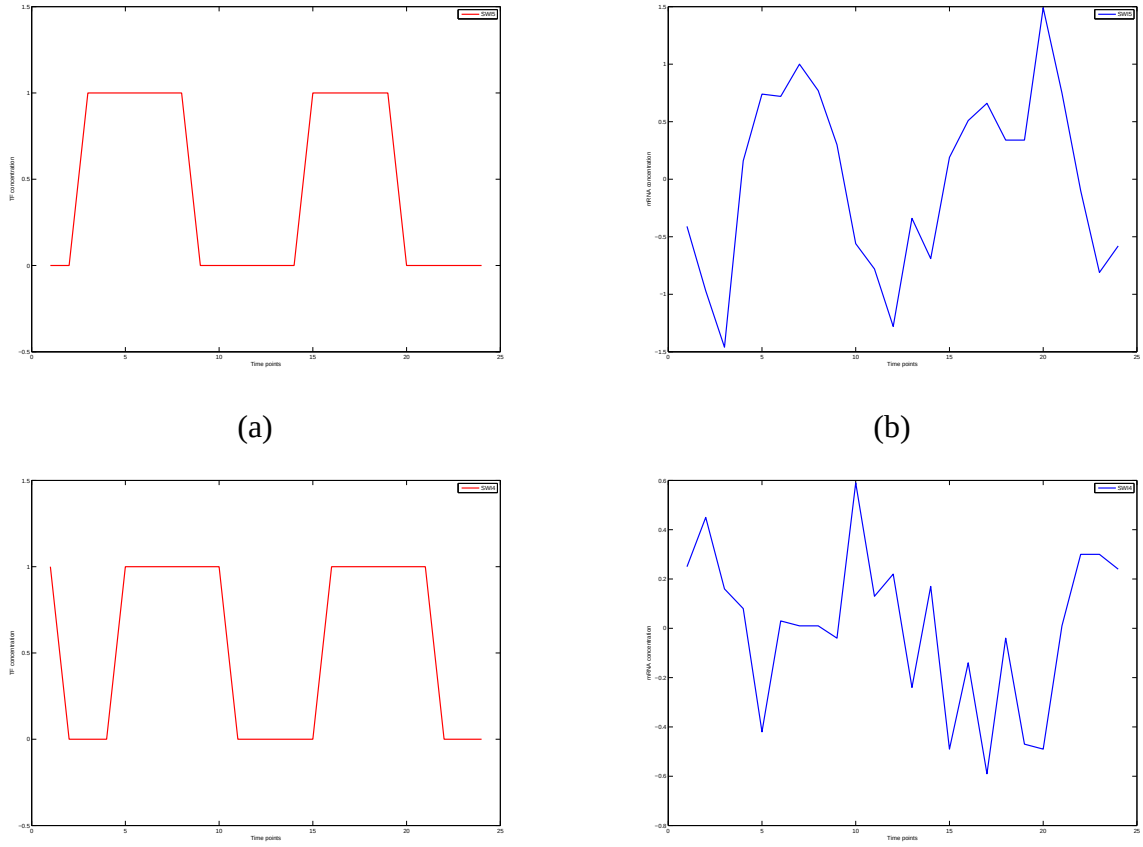


Figure 5.7: Inferred TF profiles from Spellman et al. (1998) and their corresponding mRNA expression levels. (a) Inferred TF profile for SWI5 (b) Measured mRNA expression levels for gene SWI5 (c) Inferred TF profile for SWI4 (d) Measured mRNA expression levels for gene SWI4.

els (such as Liao et al. (2003); Sanguinetti et al. (2006)), in particular recovering the periodic pattern of key cell-cycle regulators such as SWI5 and SWI4. It is shown in figure 5.7.

An analysis of the predicted interaction terms reveals that about 5% of the combinatorial interactions (A_{12} in (5.2)) are significant at 5% level as shown in figure 5.6. This accounts for 186 combinatorial interactions out of a total of 3886 possible pairwise interactions allowed by the structure of the regulatory network.

A more detailed analysis of the results obtained (across transcription factor profiles) using the model 5.2 reveals that some of the TFs in the yeast regulatory network have a much higher proportion of significant combinatorial interactions than the average. Figure 5.8 shows the percentage of significant combinatorial regulation for the all the TFs in this dataset. It can be seen from this plot that a group of TFs (DAL82, Pho2, GTS1, HAP3, HIR1, MAL13) have 15% or

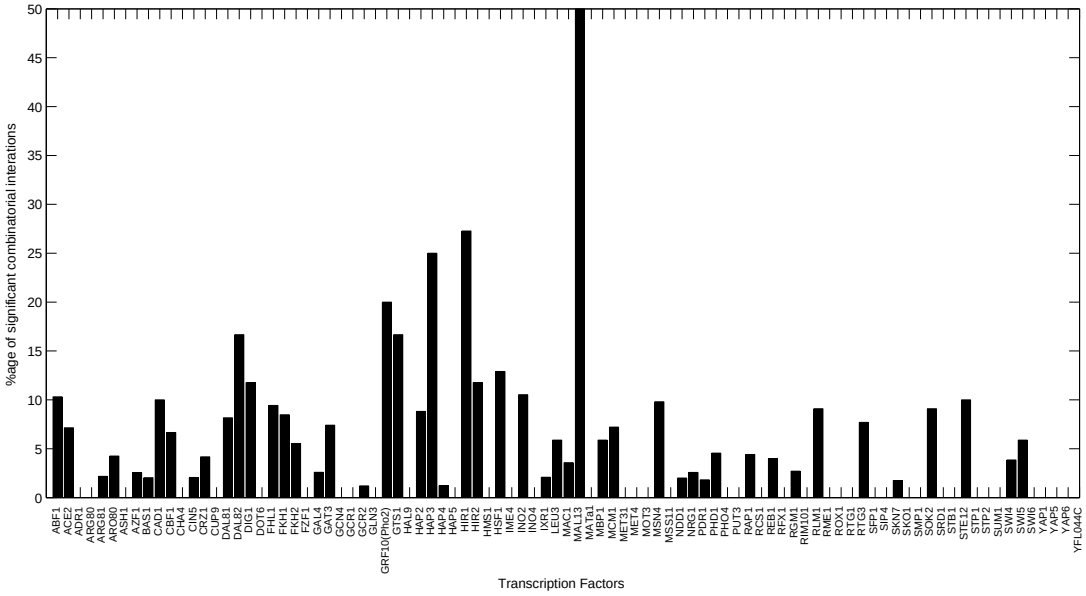


Figure 5.8: Percentage of combinatorial interactions for 104 TFs of yeast dataset (Spellman et al., 1998)

more significant combinatorial interactions compared to overall average of 5% significant combinatorial interactions. Looking at biological function of these highly interacting proteins, we found that our results are often plausible in terms of the underlying biology. The transcription factor Pho2 found to be actively involved in combinatorial regulation by our model is known to behave in a combinatorial manner (Bhoite et al. (2002)). Pho2 is functionally active in many biological processes such as histidine biosysnthesis and phosphate utilization (Daignan-Fornier and Fink, 1992). Similarly, HAP3 is a global regulator of respiratory gene expression and contains sequence contributions to both complex assembly and DNA binding (Xing et al., 1993), Hahn et al. (1988). The contributions of these transcription factors to multiple biological processes indicates that plausibly these TFs will need cofactors to achieve specificity in gene regulation.

Our model predicted that DAL82 regulatory activities contains a higher percentage of significant combinatorial regulation. DAL82 is a positive regulator of allophanate inducible genes and is one of four transcription factors that are required for this process (Scott et al., 2000). Experimental evidence in this case suggests that DAL81 protein is required for DAL82-dependent transcription activation. As shown in figure 5.8, our model also predicted the higher percentage of combinatorial activity for DAL81 (approximately 10%). GTS1 is a transcriptional co-activator for the genes that exhibits the metabolism of carbohydrates, requiring interactions with other regulators to induce gene expression (Xu and Tsurugi, 2007).

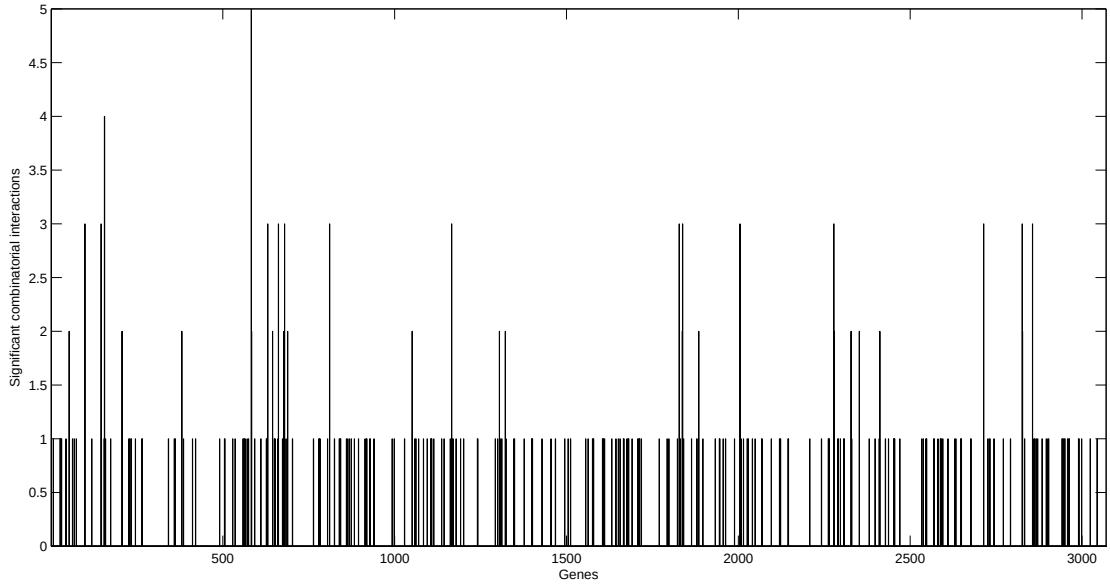


Figure 5.9: Number of $A_{ij} \geq 2\ s.d.$ for 3070 genes for yeast dataset (Tu et al., 2005)

5.5.2.3 Metabolic Cycle Data

Tu et al. (2005) studied the yeast metabolic cycle (YMC) that governs the genome-wide transcription of genes in a periodic manner. Budding yeast under nutrient-limited conditions goes through robust cycles of respiratory bursts that in turn causes almost half of the yeast genome to express periodically. In this experiment, total RNA was prepared after every 25 minutes over a period of three consecutive metabolic cycles. In order to use this dataset with our model, we fused the network connectivity available from two ChIP-on-chip experiments (Lee et al., 2002), Harbison et al. (2004) and removed the genes that were not regulated by any TFs in the connectivity information. The TFs not involved in regulating any genes were also eliminated leaving a network of 3070 genes and 177 TFs. Our probabilistic approach can handle the false positive that could arise from this dataset by assigning higher uncertainty to the regulatory interactions that are not evident from data.

Once again, the predicted activity profiles of most regulators showed a good agreement with previously reported results Sanguinetti et al. (2006) using different inference models (results not shown). In particular, our model confidently predicted a periodic behaviour for many of the regulators, which is in agreement with the experimental design. The details about the extent of the combinatorial regulation in this dataset are shown in figure 5.9 where approximately 3% of the possible combinatorial interactions are found to be statistically significant. Out of a total of 10876 possible combinatorial interactions in this data set, only 322 were predicted to have

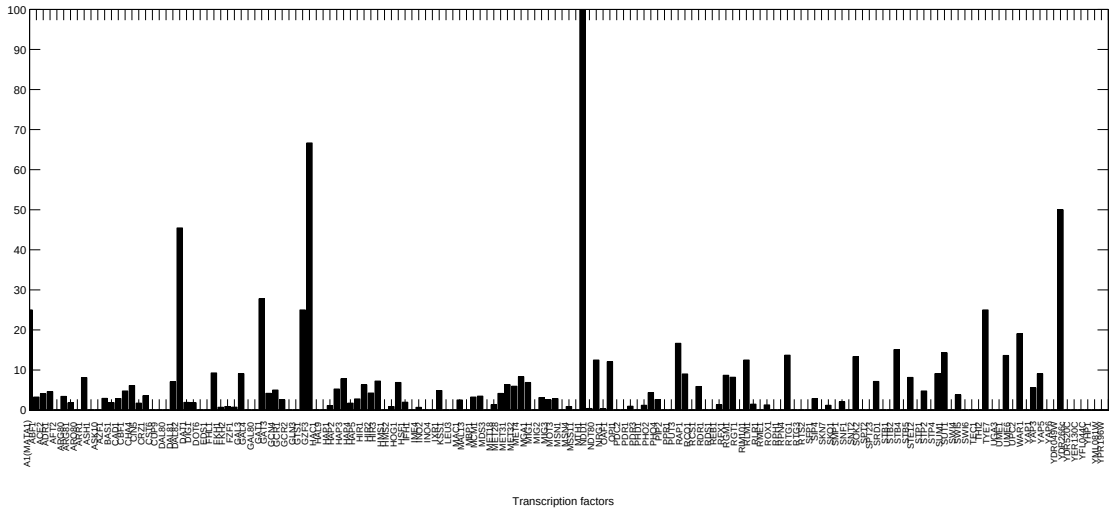


Figure 5.10: Percentage of combinatorial interactions for 177 TFs for yeast dataset (Tu et al., 2005)

posterior mean greater than 2 standard deviations.

Further analysis across the transcription factor profiles showed that a small proportion of the TFs in this dataset have significantly higher combinatorial interactions as shown in figure 5.10. The most prominent of these highly interacting TFs are: DAL82, GAT1, GTS1, GZF3, MTH1, PUT3, STB2, THI2, UPC2, VMS1. Some of these TFs appear to have consistently combinatorial behaviour between the cell cycle and the metabolic cycle; e.g. DAL82 and GTS1 could be interpreted as ”housekeeping” combinatorial TFs. GAT1, a positive regulator of nitrogen catabolite repression (NCR), is an essential regulator of the NCR-sensitive genes along with another transcription factor GLN3. The model for regulatory circuit of GAT1-GLN3 combination is discussed in Coffman et al. (1996). The majority of the other TFs predicted to have high combinatorial behaviour are clearly associated with metabolic processes: GZF3 is a catabolite repressor, MTH1 regulates glucose sensing, THI2 regulates thiamine biosynthesis, UPC2 regulates sterol biosynthesis. This is perhaps not surprising, as metabolic genes have higher expression changes within the metabolic cycle, and hence presumably a lower level of noise. However, this highlights an important feature of our model: even if the absolute fraction of combinatorial interactions recovered is rather low, predictions have higher confidence for the specific biological processes investigated in the given experiment.

5.6 Comparison with Other Methods

To assess the relative merits of our method (cFHMM), we performed an extensive comparative study with two published methods for reconstructing TF profiles. These include a standard FHMM (this is used for TF inference in Shi et al. (2008) that also account for post transcriptional modification) and TFInfer (Asif et al., 2010).

It should be stressed that the method proposed here models the non-linear interactions of the transcription factors at the promoters, something that neither of the competitor methods can do. The flip side of this extra flexibility is that more time is required to execute the algorithm.

Table 5.3 presents the comparison of the results obtained using our method with two other methods on the real data sets considered in this study. In the *E. coli* data set, the results of FHMM and cFHMM are similar in terms of TF reconstruction (average Hamming distance 0.067); this is probably due to fact that we did not find any combinatorial interactions at 5% significance level. In the other data sets, we obtained a relatively larger Hamming distances between FHMM and both cFHMM and TFInfer (0.2688 and 0.2502 respectively). These data sets contained many more time points, which allowed the recovery of a small but non-negligible number of combinatorial interactions, leading to the predictions of cFHMM (which does take these interactions into account) to be significantly different from the two linear methods.

5.7 Conclusion

We present a novel method to infer combinatorial interactions between transcriptional regulators from expression data and network connectivity data. To our knowledge, this is the first statistical method which simultaneously infers TFAs and their combinatorial interactions in large-scale networks. We model TFAs as latent binary variables with Markovian dynamics; gene expression is determined by the latent TFAs through a non-linear likelihood which allows for pairwise interactions between TFs. According to our model, gene expression is digitized; digitized levels of gene expression have recently been shown to yield computational savings and more robust predictions (Tuna and Niranjan, 2010). The principal novelty of our work in this perspective is to connect the level of discretisation with the state of underlying regulators.

We conducted experiments on simulated data (with two different connectivities, the E.coli connectivity data and the yeast connectivity with varying network sizes. The data was generated from the cFHMM model; however, we noted that both cFHMM and FHMM managed to give good reconstructions of the TF profiles (obviously FHMM could not capture the coefficients of the non-linear effects). This is essentially due to the sparsity of the connectivity; in particular,

Dataset / Method	Partridge et al. (2007)	Spellman et al. (1998)	Tu et al. (2005)
FHMM	Run time: 6 seconds MSE: 0.0189 HD with cFHMM=0.0667 HD with TFInfer=0.0667	Run time: 4.7 hours MSE: 0.1381 HD with cFHMM=0.2688 HD with TFInfer=0.2015	Run time: 5.5 hours MSE: 0.4332 HD with cFHMM=0.2502 HD with TFInfer=0.2280
cFHMM	Run time: 22 seconds MSE: 0.0423 HD with FHMM=0.0677 HD with TFInfer=0.1333	Run time: 42 hours MSE: 0.1391 HD with FHMM=0.2688 HD with TFInfer=0.2708	Run time: 335 hours MSE: 0.4125 HD with FHMM=0.2502 HD with TFInfer=0.3021
TFInfer (Asif et al., 2010)	Run time: 45 seconds MSE: 0.0399 HD with FHMM=0.0667 HD with cFHMM=0.1333	Run time: 10 hours MSE: 0.1156 HD with FHMM=0.2015 HD with cFHMM=0.2708	Run time: 115 hours MSE: 0.3811 HD with FHMM=0.2280 HD with cFHMM=0.3021

Table 5.3: Comparison of different techniques for inference of the states of transcription factors with different biological datasets. The states inferred with different methods are compared using the Hamming distance (HD) between the vectors of states.

the connectivity matrix in the yeast data is sparser, so that FHMM is a very good model for most genes. For the denser *E. coli* network, the performance of cFHMM was significantly better, particularly in terms of MSE (table 5.1). The results on real datasets show predictions that are in good agreement with existing methods. However, the length of the time-series data is a critical factor to obtain the statistically significant combinatorial interactions.

Factorial Hidden Markov Models have been previously used to model TFAs (Shi et al., 2008); in that work, further dependencies were included between TF mRNA expression levels and their predicted activities, which enabled to predict possible post-transcriptional modifications in TFs. Naturally, it should be possible to combine both our approach and their approach to give a model capable of simultaneously inferring TFAs, combinatorial interactions and post-transcriptional regulations. This would also allow to remove the assumption, hard-wired into our model as well as many other related models, that TFAs are independent of their mRNA expression levels. While in many cases this assumption is justified by the fact that measurement of TF gene expression are often poor proxies for their activity state, it is plausible that, at least in some situations, mRNA expression levels of TF genes will bear some influence on their activity.

Chapter 6

Simultaneous Inference and Clustering of Transcriptional Dynamics in Gene Regulatory Networks

In the last chapter, we presented a variant of FHMM to model the hidden TFAs in the regulatory network as binary Markov chains and used a variational approximation to find the posterior estimates. The transition rates for the latent Markov chains were not inferred in that model. One critical factor that hinders the inference of these transition rates from the experimental data is the length of the time-series which is not sufficient is most of the cases. In that model, we fixed the transition rates of the latent Markov chains of the FHMM to plausible values that were coherent with the underlying biological assumptions. One way to deal with the limited length of time-series is to pool the data together from different time-series and use it for the inference of the transition rates. This pooling scheme serves two purposes: firstly, no assumptions are required to fix the transition rates in the inference; secondly, as a consequence, pooling the data from different time-series also clusters the latent Markov chains into a priori unknown number of clusters.

In this chapter, we present a novel method for simultaneous inference and nonparametric clustering of transcriptional dynamics from gene expression data. The proposed method uses gene expression data to infer time-varying TF profiles and cluster these temporal profiles according to the dynamics they exhibit. We use the latent structure of FHMM to model the TF profiles as Markov chains and cluster these profiles using nonparametric mixture modelling. An efficient Gibbs sampling scheme is proposed for inference of latent variables and grouping of transcriptional dynamics into a priori unknown number of clusters. We test our model on simulated data and explore the effect of different noise levels of observations on the inference results with varying network size. We also analyse our model's performance on two expression datasets; *S. cerevisiae* cell cycle data and *E.coli* oxygen starvation response data and show its

applicability for genome wide analysis of expression data.

6.1 Introduction

High throughput microarray experiments generate vast amounts of data about the expression patterns genes. The abundance of gene expression data poses many mathematical and computational challenges to reverse engineer the molecular processes responsible for transcriptional regulation. Gene expression is regulated by the binding of TF proteins to the promoter regions of genes. Reconstructing the dynamics of transcriptional regulation in gene regulatory network, however, remains an open issue due to the difficulties involved in experimental measurement of TF activity levels. Experimental techniques such as ChIP-on-chip technique (Lee et al., 2002), which directly measure the binding of TFs to promoters, can provide a static picture of the wiring (connectivity) of the regulatory network. This architectural information is partially available for humans and mouse, and almost fully documented for yeast and *E.coli*. Combining this architectural information with gene expression data, it is possible to decipher the role of TF proteins in the genetic machinery using statistical tools. Over the last few years, several methods have been proposed to infer the activities of several TF proteins from the expression of (hundreds or thousands) of their target genes (Liao et al., 2003; Sabatti and James, 2006; Sanguinetti et al., 2006; Asif et al., 2010), leading frequently to useful biological insights (Partridge et al., 2007; Davidge et al., 2009).

One subcategory of these inference approaches is based on FHMMs (Ghahramani and Jordan, 1997). In FHMM-based inference methods (Shi et al., 2008; Asif and Sanguinetti, 2011), each latent Markov chain models the (binary) activity of a TF protein, assuming *a priori* independence between different TFs. The distributed latent state representation of FHMMs provides a natural way to model the regulation of genes by multiple TFs. Each TF is characterised by prior propensities to switch state (transition rates), which also have to be determined from the data in general. The states of the TFs are assumed to be either on or off that corresponds to underlying biological assumptions that the number of TF molecules per cell is sufficient to saturate the downstream transcriptional machinery and TF rapidly changes from active to inactive states and vice versa (Ptashne and Gann, 2002).

However, the length of the Markov chain plays a pivotal role in enabling reliable estimation of transition rates. Most biological datasets are of very limited length (at most a few tens of time-points), making reliable estimation of transition rates effectively very difficult. While fixing prior rates to a plausible value implying few transitions may be a practical solution in some cases (Asif and Sanguinetti, 2011), in general this will be potentially inaccurate for

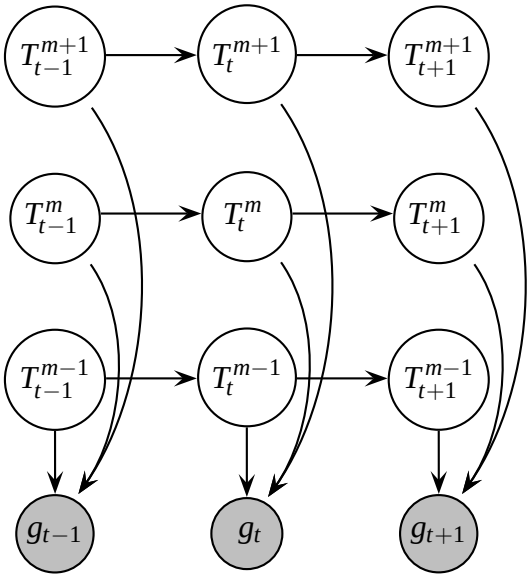


Figure 6.1: A factorial HMM with 3 chains.

large data sets. A biologically more plausible assumption could perhaps be obtained from the observation that TFs rely on few different activation mechanisms: for example, many TFs are activated by rapid conformational changes (Ptashne and Gann, 2002), while others rely on more gradual changes in concentration. Therefore, it is natural to assume that TF dynamics may be *clustered*, with several TFs sharing the same transition rates. Besides the advantages of more biologically interpretable results, this clustering approach is also attractive from the statistical point of view: by pooling data from different TFs, it allows a more reliable estimation of transition rates.

In this work, we build on the FHMM model of transcriptional regulation (Fig. 6.1) for inference of TF profiles and employ a clustering approach to group the inferred TF profiles based on their dynamics. Since specifying a number of clusters *a priori* is not possible, we propose to use Dirichlet Process Mixture (DPM) models (Ferguson, 1973; Antoniak, 1974; Rasmussen, 2000) to tackle the problem of model selection. Our proposed method does not make any assumption about the dynamics for TF profiles as we learn these dynamics by pooling the statistics from the groups of inferred TF profiles. In the following text, we present the model for inference and clustering of transcriptional dynamics and propose an efficient Gibbs sampling scheme for inference in the hierarchical model (Fig. 6.2). Then we test our model using simulated datasets and apply it to two well studied real datasets in *Saccharomyces cerevisiae* and *Escherichia coli*, showing how the model can return biologically meaningful clusterings.

6.2 Modelling Regulatory Dynamics

Suppose that N genes are regulated by M TFs over T time-points. Let g_i^t be the (log) mRNA expression level of gene i at time t, and let $\{T_j\}_i \quad j \in \mathcal{J}_i \subset \{1,\ldots,M\}$ be the set of TFs regulating gene i. We will model (log) gene expression as a linear combination of the activity of TF inputs as (Asif and Sanguinetti, 2011)

$$g_i^t \;=\; \mathbf{e}_t^T \theta_i \;+\; \varepsilon \tag{6.1}$$

where $\mathbf{e_t}$ is composed of the binary states of the TFs that bind gene i, $\theta_\mathbf{i}$ is a set of interaction strength parameters specific for gene i and ε is Gaussian distributed measurement noise with variance σ^2 (and mean 0). It is important to note that e_t is a vector of states of all the TFs that regulate gene i and thus also encodes the connectivity information of the regulatory network. For example, in the simple case of two TFs binding gene i, we obtain

$$g_i^t \;=\; A_i^1 T_t^1 + A_i^2 T_t^2 + b \;+\; \varepsilon. \tag{6.2}$$

The prior for the parameter θ_i is assumed to be a zero mean Gaussian with variance encoded by a hyper-parameter α^2,

$$\theta_i \sim \mathcal{N}(0, \alpha^2).$$

The TF states (entering the vector $\mathbf{e}_t$) are assumed to follow Markovian dynamics, with prior independence between different TFs. The basic architecture of our expression model is therefore given by a FHMM, depicted graphically in Figure 6.1. As evident from equation (6.1), the latent variable $\mathbf{T}$ and the parameter Θ only appear through their product, leading to an identifiability problem. We can take the example of equation (6.2) to elaborate on this. Equation (6.2) is invariant to following transformation

$$\begin{aligned} T_t^1 &\to 1 - T_t^1 \forall t \in \{1,\ldots,T\} \\ b' &\to A_1 + b, \quad A_1' \to -A_1. \end{aligned}$$

which we refer to as the *flipping* of TF profile. This ambiguity can easily be resolved with prior knowledge which is often available. Examples of such prior knowledge could be the experimental evidence that the TF is activating/repressing a specific downstream target or that a particular TF is in a specific state of activation at the beginning of the time course.

6.2.1 Clustering Temporal Profiles by Dynamics

In standard FHMM setting, each latent Markov chain is characterised by a transition matrix that specifies the conditional probabilities of moving from one state to another. However, we proposed to use a shared transition matrix for multiple latent Markov chains of the FHMM that have similar dynamics. This sharing of transition matrices leads to clustering of Markov chains as we show later. Since it is impossible to know about the number of clusters that govern the dynamics of latent Markov chains, a non-parametric approach is required to deal with the unknown number of clusters of Markov chains.

DPM models are nonparametric Bayesian methods that encode the natural clustering property that the prior probability of cluster membership is proportional to the size of the cluster. DPMs have been widely used for nonparametric clustering of expression data (Medvedovic and Sivaganesan, 2002; Dahl, 2006; Savage et al., 2010). DPM is characterised by a hyper-parameter η, Dirichlet distributed π that serves as the prior for indicator variables z_m and cluster specific parameters τ^k.

Clustering by dynamics implies that we estimate the dynamics exhibited by TF profiles (*i.e.,* the transition rates) and then cluster these dynamics. The estimation of the dynamics from TF profiles is based on the transitions between time points; in case of binary Markov chains, this boils down four possible transitions in a Markov chain as we describe later. It can also be understood as the clustering of the transition dynamics of TF profiles rather than TF profiles themselves.

The FHMM assigns each TF to a different Markov chain with a priori different dynamics. This may be undesirable for biological or statistical reasons. From a biological perspective, there are fewer processes that regulate the transcriptional machinery compared to the number of TFs in GRN. To take this into account, we use z_m as the indicator variable that assigns TF m to one of K clusters of the DPM model. In this way, prior over TF m can be specified as

$$p\left(T_1^m,\ldots,T_T^m|z_m,\tau^k\right)=\prod_{t=1}^{T}p_{z_m}\left(T_{t+1}^m|T_t^m,\tau^k\right)$$

where τ^k is the transition matrix for cluster k that governs the dynamic behaviour of TF m with $z_m=k$.

The individual transition probabilities of τ^k are denoted by ξ_j^k; it is useful to interpret ξ_j^k as *persistence probabilities* ($p(T_t^m=0|T_{t-1}^m=0)$ or $p(T_t^m=1|T_{t-1}^m=1)$) as these probabilities are used to construct τ^k. The probabilities of changing states (off diagonal entries of τ^k) are easily obtained by normalisation. The prior over these persistence probabilities is taken to be given by $\xi_j^k=Beta(\lambda_1,\lambda_2)$. As we normally do not have prior information over the values of λ, we

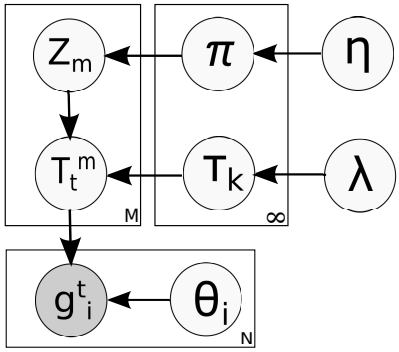


Figure 6.2: Graphical model (Static case)

will fix this hyperprior to be uniform by taking $\lambda_1 = \lambda_2 = 1$.

The graphical representation of the model (without time dynamics for simplicity) is shown in figure 6.2. Notice that the TF profiles are independent given τ^k and the cluster assignments. It is interesting to speculate what this implies in terms of which TFs will be clustered together. Naturally, TFs with very similar profiles are highly likely to be clustered together. However, clustering by dynamics implies that some clusters will also include very different profiles: for example, TFs who are mainly in one state and occasionally briefly visit the other state are also likely to be clustered together. Biologically, this would mean that TFs which are only needed at specific times during the time course are clustered together, which can be biologically meaningful.

We emphasise that the number of Markov chains in this model is fixed and we are considering one time-series/TF profile as a single entity to estimate the sufficient statistics of the Markov chain. The sufficient statistics obtained from a time-series contribute towards the inference of the number of clusters and the dynamics of clusters.

6.3 Inference using Gibbs Sampling

We aim to infer the temporal profiles of TFs, strength of the genetic interactions and cluster the dynamics exhibited by TF profiles. We use gene expression data and connectivity information of the regulatory network in our inference procedure. Due to the intractability of Bayesian inference and highly parameterised nature of our model, we resort to Gibbs sampling. Gibbs sampling requires drawing samples from the conditional posterior distribution (CPD) of one set of variables given all others. Derivations of these CPDs is greatly aided by the conditional independences implied in the model (see figure 6.2).

The CPD for θ_i given the TF profiles and expression measurements is a multivariate Gaussian distribution given by

$$\prod_{i=1}^{N} \mathcal{N}(\theta_i | \mathbf{m_i}, \Sigma_i) \tag{6.3}$$

with mean and covariance given by

$$\Sigma_i^{-1} = \frac{1}{\sigma^2} \sum_{t=1}^{T} \mathbf{X}_i \mathbf{e}_t \mathbf{e}_t^T \mathbf{X}_i + \alpha^{-2} \mathbf{I}$$

$$\mathbf{m_i} = \frac{1}{\sigma^2} \left[\sum_{t=1}^{T} g_i^t \mathbf{e}_t^T \mathbf{X}_i \right] \Sigma_i^{-1}$$

Here $\mathbf{X}_i$ denotes a diagonal matrix with the i-th row of the connectivity matrix $\mathbf{X}$ along the diagonal.

The CPD for each TF at each time point can be obtained by using the conditional independence properties of graphical models. It is given by

$$P(T_t^m | \Phi) = \frac{p_{z_m}(T_t^m | T_{t-1}^m) p_{z_m}(T_{t+1}^m | T_t^m) p(g_t^i | T_t^i)}{\sum_{T_t^m} p_{z_m}(T_t^m | T_{t-1}^m) p_{z_m}(T_{t+1}^m | T_t^m) p(g_t^i | T_t^i)}$$

where $\Phi = \{T_{t-1}^m, T_{t+1}^m, T_t^{-m}, g_t^i, \theta_i, \mathbf{X}, \mathbf{Z}, \tau\}$ and p_{z_m} is the transition matrix for cluster k of DPM such that $z_m = k$. To improve the efficiency of posterior estimation, we employed the stochastic Forward Backward algorithm (Boys et al., 2000) for simultaneous sampling of all the states of a Markov chain. For this purpose we run the Forward algorithm to obtain the forward message $\alpha_{z_m}^t(T_t^m)$ and then use it in above equation to get

$$P(T_t^m | \Phi) = \frac{\alpha_{z_m}^t(T_m) p_{z_m}(T_{t+1}^m | T_t^m)}{\sum_{T_t^m} \alpha_{z_m}^t(T_m) p_{z_m}(T_{t+1}^m | T_t^m)} \tag{6.4}$$

which is then used to sample T_t^m.

6.3.1 Collapsed Gibbs Sampling of Cluster Memberships

For inference of z_m, we use a collapsed Gibbs sampling approach integrating out π and τ^k, so that we need to take samples from $p(z_m = k | \mathbf{z}_{-m}, \mathbf{T}_{1:T}^{1:M}, \eta, \lambda)$. To obtain the CPD of cluster assignment variables z_m, we start as follows:

$$\begin{aligned}
p(z_m = k | \mathbf{z}_{-m}, \mathbf{T}_{1:T}^{1:M}, \eta, \lambda) &= p(z_m = k | \mathbf{z}_{-m}, \mathbf{T}_{1:T}^{-m}, T_{1:T}^m, \eta, \lambda) \\
&\propto p(z_m = k | \mathbf{z}_{-m}, \mathbf{T}_{1:T}^{-m}, \eta, \lambda) p(T_{1:T}^m | z_m = k, \mathbf{z}_{-m}, \mathbf{T}_{1:T}^{-m}, \eta, \lambda) \quad (6.5) \\
&= p(z_m = k | \mathbf{z}_{-m}, \eta) p(T_{1:T}^m | \mathbf{T}_{1:T}^{k,-m}, z_m = k, \lambda) \quad (6.6)
\end{aligned}$$

Here $\mathbf{z}_{-m}$ is the set of clustering assignment for all TFs except T^m, $\mathbf{T}_{1:T}^{-m}$ is the set of all TF profiles except TF m and $\mathbf{T}_{1:T}^{k,-m}$ is the set of TFs profiles already assigned to cluster k except TF m. We use Bayes theorem in equation (6.5) and conditional independence property of graphical models in equation (6.6). The first term in equation (6.6) can be interpreted as the predictive prior and is due to the marginalization of π. Using standard results in the DPM literature, we obtain

$$p(z_m = k|\mathbf{z}_{-m}, \eta) = \frac{n_{k,-m} + \eta/K}{M + \eta - 1} \tag{6.7}$$

here $n_{k,-m}$ is the number of TFs already assigned to cluster k of DPM.

The second term in the CPD of z_m is the predictive likelihood which is calculated by integrating out the cluster specific parameters τ^k. As we see later, it depends on the count of transitions for the TF profiles that are currently assigned to cluster k excluding the TF m,

$$\begin{aligned} & p(T_{1:T}^m|\mathbf{T}_{1:T}^{k,-m}, z_m = k, \lambda) \\ &= \int p(\mathbf{T}_{1:T}^m|\tau^k, z_m = k) p(\tau^k|\mathbf{T}_{1:T}^{k,-m}, z_m = k, \lambda)\, d\tau^k \\ &= \int \prod_{j=1}^{2} \left[(\xi_j^k)^{x_j^k} (1-\xi_j^k)^{y_j^k} \right] \prod_{j=1}^{2} \left[\frac{\Gamma(a_j^k + b_j^k)}{\Gamma(a_j^k)\Gamma(b_j^k)} (\xi_j^k)^{a_j^k - 1} (1-\xi_j^k)^{b_j^k - 1} \right] d\xi_j^k \end{aligned} \tag{6.8}$$

To compute the conditional posterior for the persistence probabilities, we define

$$x_j^k = \begin{cases} \#\{T_t^m = 0, T_{t-1}^m = 0\} & \text{if } j = 1 \\ \#\{T_t^m = 1, T_{t-1}^m = 1\} & \text{if } j = 2 \end{cases}$$

$$y_j^k = \begin{cases} \#\{T_t^m = 1, T_{t-1}^m = 0\} & \text{if } j = 1 \\ \#\{T_t^m = 0, T_{t-1}^m = 1\} & \text{if } j = 2 \end{cases}$$

$$N_{j1}^k = \sum_{\kappa: z_m = k} x_j^\kappa \;\;,\;\; N_{j2}^k = \sum_{\kappa: z_m = k} y_j^\kappa$$

$$a_j^k = \lambda_1 + N_{j1}^k \;\;,\;\; b_j^k = \lambda_2 + N_{j2}^k$$

The CPD for the persistence probabilities ξ_j^k is therefore given by the following distribution

$$\xi_j^k \;\sim\; Beta(a_j^k, b_j^k) \tag{6.9}$$

Note that these transition rates are estimated by pooling the statistics of all the TFs currently assigned to cluster k of the DPM; this provides more robust estimates of transition rates. Plugging this back into equation (6.8), we obtain

$$p(T_{1:T}^m|\mathbf{T}_{1:T}^{k,-m}, \lambda) \;\; = \;\; \prod_{j=1}^{2} \frac{\Gamma(a_j^k + b_j^k)}{\Gamma(a_j^k)\Gamma(b_j^k)} \cdot \frac{\Gamma(a_j^k + x_j^k)\Gamma(b_j^k + y_j^k)}{\Gamma(a_j^k + x_j^k + b_j^k + y_j^k)} \cdot \mathbf{1} \tag{6.10}$$

which finally leads to the following CPD for latent indicator variables

$$p(z_m = k|\mathbf{z}_{-m}, \mathbf{T}_{1:T}^{1:M}, \eta, \lambda) = \frac{n_{k,-m} + \eta/K}{M + \eta - 1} \cdot \prod_{j=1}^{2} \left[\frac{\Gamma(a_j^k + b_j^k)}{\Gamma(a_j^k)\Gamma(b_j^k)} \cdot \frac{\Gamma(a_j^k + x_j^k)\Gamma(b_j^k + y_j^k)}{\Gamma(a_j^k + x_j^k + b_j^k + y_j^k)} \right]. \quad (6.11)$$

The Gibbs sampling algorithm for the inference of the $\Theta, \mathbf{T}$ and z_m is outlined in algorithm (4) where each random variable is sampled from the CPD iteratively until the sampler is deemed to have converged.

6.4 Experimental Analysis

To test our model, we check its performance on two simulated datasets. Then we perform a sensitivity analysis of the model using simulated datasets of varying sizes with different levels of noise. Finally we show the applicability of our model on two real datasets.

6.4.1 Analysis using Simulated Data

One simulated dataset is relatively small compared to the scale of most regulatory networks and consists of 20 genes, 5 TFs and two transition matrices governing the dynamics of the TF profiles ($N = 20, M = 5, K = 2, T = 20$). The other simulated dataset is larger with 100 genes, 20 TFs and 3 transition matrices to account for TFs dynamics ($N = 100, M = 20, K = 3, T = 20$). We start by generating the cluster assignments that relate each TF to one of the transition matrices; which are then used to generate TF temporal profiles $\mathbf{T}$. Using these temporal profiles with the artificial Θ parameters and the known regulatory architecture, we generate the expression profiles for all the genes in the dataset with added Gaussian noise.

It is important to mention that if the persistence probabilities in the transition matrix are low then two temporal profiles sampled from the same transition matrix can be sufficiently different. It is then possible that the nonparametric clustering approach we employ may decide to generate an extra cluster and cluster these two TFs separately. This scenario is elaborated with the help of an example in section 6.4.1.2. Similar problems may occur when the inferred TF profile is flipped. One principled approach to avoid these flips in simulated and real data analysis is by incorporating the prior knowledge about the dynamics of TFs at the initial time point. In case of simulated datasets, flipping can be avoided by assuming that all the TFs are off at the start of the experiment and base expression levels of all the genes is zeros when not bound by any TF.

Algorithm 4 Gibbs sampling algorithm for inference in DPM-FHMM

Require: Initialise **T** randomly or from expression data (**G**)

Require: Initialise $\{z_m\}_{m=1}^{M}$

Require: $\alpha^2 \leftarrow 1$

Require: $\sigma^2 \leftarrow 0.1$

Require: Initialise z_m

repeat
  for $i \leftarrow 1, N$ **do**
    Update the CPD (Eq. 6.3) over θ_i for gene i given $\{G, T_{1:T}^{1:M}\}$
  end for
  for $m \leftarrow 1, M$ **do**
    Update the CPD of $T_{1:T}^{m}$ (Eq. 6.4) given $\{z_m, \tau^k, \Theta, \mathbf{G}\}$
  end for
  Update $\mathbf{e}_{1:T}$ using the state marginals
  for $m \leftarrow 1, M$ **do**
    Update the CPD of z_m (Eq 6.11) given $\{\mathbf{T}_{1:T}^{1:M}\}$
  end for
  Remove empty cluster to get K_{active}
  for $k \leftarrow 1, K_{active}$ **do**
    for $j \leftarrow 1, 2$ **do**
      Sample $\xi_j^k \sim Beta(a_j^k, b_j^k)$ given $\{\mathbf{T}_{1:T}^{1:M}, \mathbf{Z}\}$
    end for
  end for
until Converged

Label switching is a major problem in mixture modelling and our model faces the same challenge. This reflects the possibility that same labelling may recur in a sample with clusters labelled differently. While there are approaches available in the literature for dealing with the label switching problem in the context of finite mixture models (Celeux et al., 2000; Stephens, 2000; Frühwirth-Schnatter, 2001), fewer are available in case of unbounded numbers of clusters. One possible remedy is to use an $M \times M$ *co-occurrence* matrix $\mathbf{C}$ that, for each pair of TFs, stores the sample fraction with both members of the pair falling in the same cluster. The entries in the symmetric matrix $\mathbf{C}$, for each draw of the Gibbs sampler, are 1 along the diagonal and 1 for row i and column j if TFs i and j fall in the same cluster, zero otherwise. The matrix $\mathbf{C}$ is invariant to label switching and hence identifiable. We use $\mathbf{C}$ to calculate $\hat{\mathbf{C}}$ that summarises MCMC draws of $\mathbf{z}_m$ after the burn-in period of the Gibbs sampler.

We systematically compare our approach with standard FHMM throughout our experiments; results of these comparisons are reported in table 6.1, where the proposed method is referred to as DPM-FHMM . For the sake of comparison, we use FHMM to infer the temporal profiles of TFs, regulatory interactions and transition rates via Gibbs sampling. The criteria for comparison are mean squared error (MSE) in reconstructing the temporal profiles of genes and Hamming distance (H.D.) between inferred TF profiles using our model with FHMM. In general, the two methods provide similar MSEs with our method better at inferring the TFAs (*i.e.* H.D.) where the experimental noise is high (see section 6.4.2). Obviously, our method also has the added benefit of interpretable clustering of TFs.

In order to analyse the clustering obtained from our model, we use the TF profiles inferred using FHMM and cluster them profiles using K-means algorithm with H.D. as the distance measure. The results obtained for K-means clustering for these two simulated datasets in shown in the subsequent sections.

6.4.1.1 Simulated Dataset #1

The clustering assignment in our method is unconstrained and is only bounded by the total number of TFs in the dataset. Due to this, each draw of the Gibbs sampler may have different number of clusters in it. The inferred co-occurrence matrix for small simulated dataset in shown in figure 6.3a. The information in this co-occurrence matrix lacks one critical piece of information *i.e.* the number of clusters.

To infer the number of cluster, we collect the total number of clusters present in each MCMC draw after the burn-in period. This information is shown in figure 6.3b after normalisation and can be interpreted as the posterior probability distribution over the number of clusters. This suggest that TF profiles are best explained when clustered in 2 groups which is

Datasets	MSE (DPM-FHMM)	MSE (FHMM)	HD (with ground truth)	HD (FHMM with DPM-FHMM)
Simulated dataset #1	0.0086	0.0086	0	0
Simulated dataset #2	0.0086	0.0086	0	0
Partridge et al. (2007)	0.0889	0.3404	N.A.	0.2333
Spellman et al. (1998)	0.2469	0.1607	N.A.	0.2444

Table 6.1: Comparison of the proposed method with FHMM on simulated and real datasets

consistent with the original co-occurrence matrix shown in figure 6.3c.

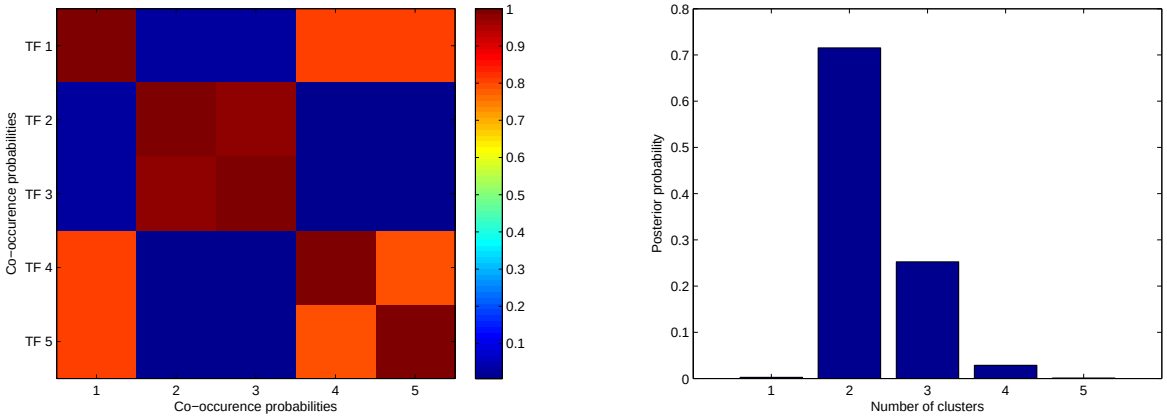
It is easy to find the clustering assignment of all TFs from the inferred co-occurrence matrix. From the co-occurrence matrix in figure 6.3a, we see that TF 2 and TF 3 are grouped together in one cluster, TF 1, 4 and 5 in another cluster. While comparing the accuracy of our model's predictions in terms of inferred TF profiles with the ground truth, we found our model is able to reconstruct the TF profiles with 100% accuracy as shown in table 6.1.

Figure 6.3d shows the co-occurrence matrix for the inferred profiles clustered using K-means algorithm (with $K = 2$). As the number of samples is very few this case ($M = 5$), K-means algorithm is unable to find the right cluster membership.

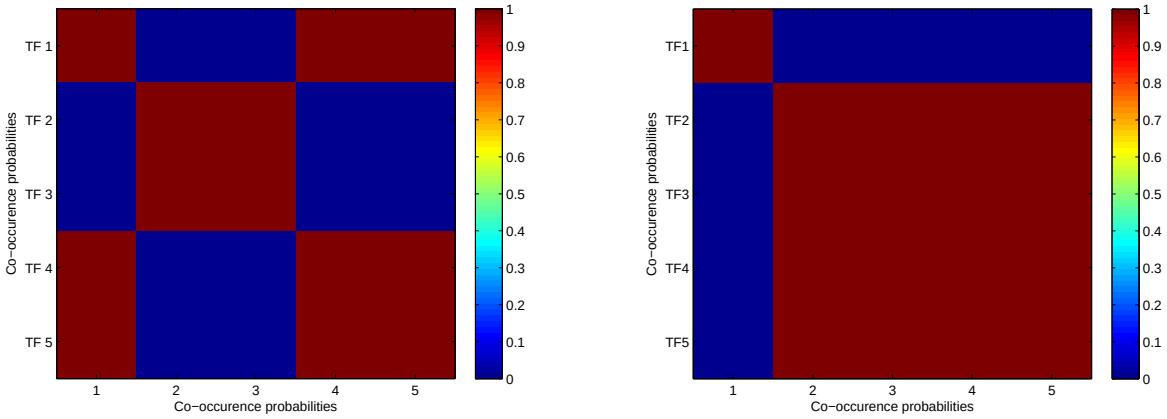
6.4.1.2 Simulated Dataset #2

The results for clustering of TFs for this dataset are summarised in figure 6.4a-b in the form of co-occurrence matrices and posterior distribution over the number of clusters. The TF profiles in this dataset are generated from 3 transition matrices. Although our method is able to reconstruct the TF profiles without any false positive or negatives (true TF profiles for simulated dataset #2 shown in figure 6.5b), the histogram in figure 6.4b suggests that there could be 4 or 5 clusters of TF profiles. This is due to considerable amount of variability in the TF profiles that are generated from the same transition matrix. An instance of this weak co-occurrence can be seen from the co-occurrence probabilities of TF 2 and TF 3 (and similarly for TF 13 and TF 15) in figure 6.4a that are not co-clustered with high co-occurrence probability; this results in the instantiation of a new cluster to accommodate relatively different dynamics of these TFs.

The *splitting* of a cluster can be seen by considering TF 2, 4, 7, 8, 10 and 13 which are co-clustered during data generation as shown figure 6.4c. A close look at figure 6.4a shows that the co-occurrence probabilities for this cluster of TFs are not comparable to other co-occurrence probabilities. It is easy to find two groups of TFs within this cluster with high co-occurrence probabilities; one group for TF 2, 7 and 13 and another group for TF 4, 8 and 10. Furthermore, the subgroup with TF 2, 4 and 7 shows weak co-occurrence between TF 2 and 13. This a



(a) (b)



(c) (d)

Figure 6.3: Results using simulated dataset 1 (a) Inferred co-occurrence matrix constructed from simulated dataset 1 (b) Posterior probability distribution over number of clusters inferred from simulated dataset 1 (c) Co-occurrence matrix constructed from known cluster assignments for simulated dataset 1 (d) Co-occurrence matrix constructed using K-means algorithm based on the inferred TF profiles from FHMM.

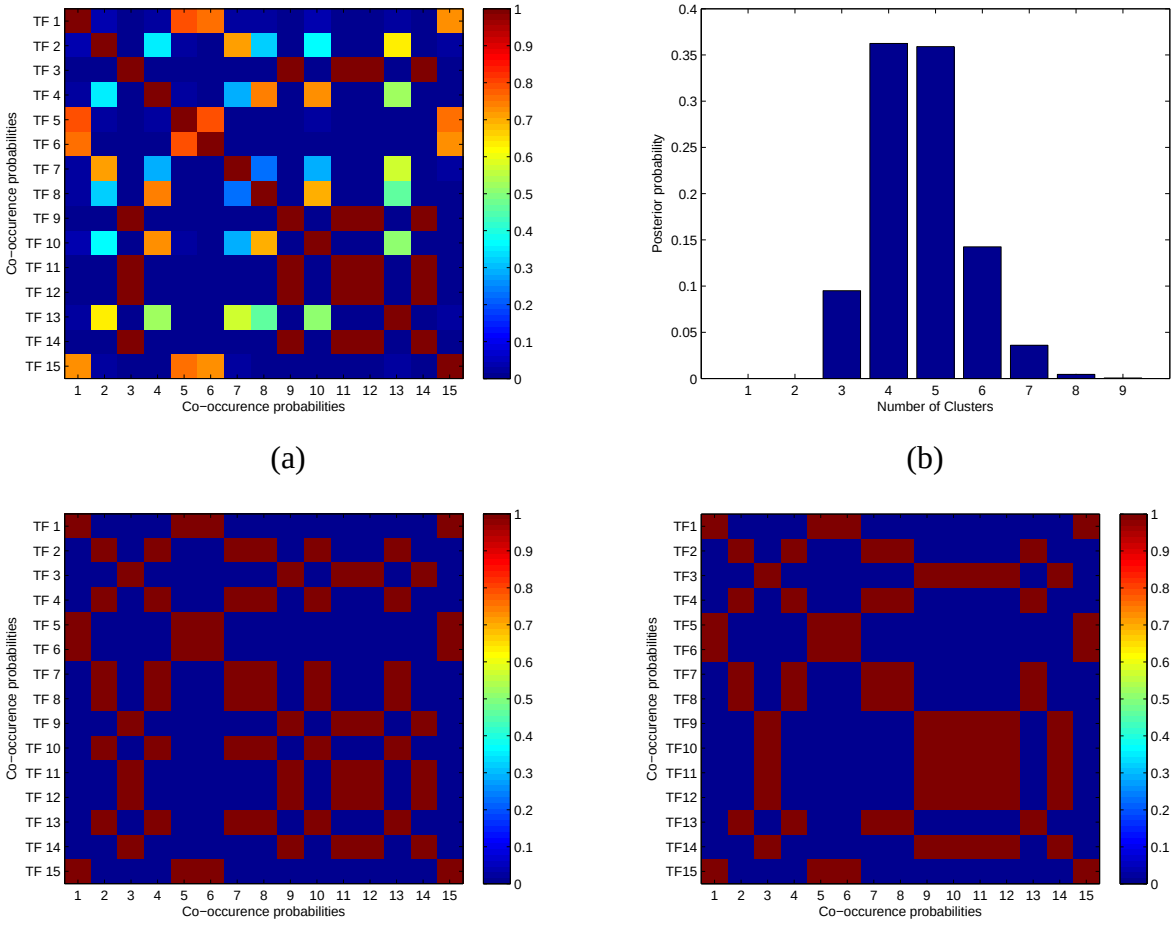
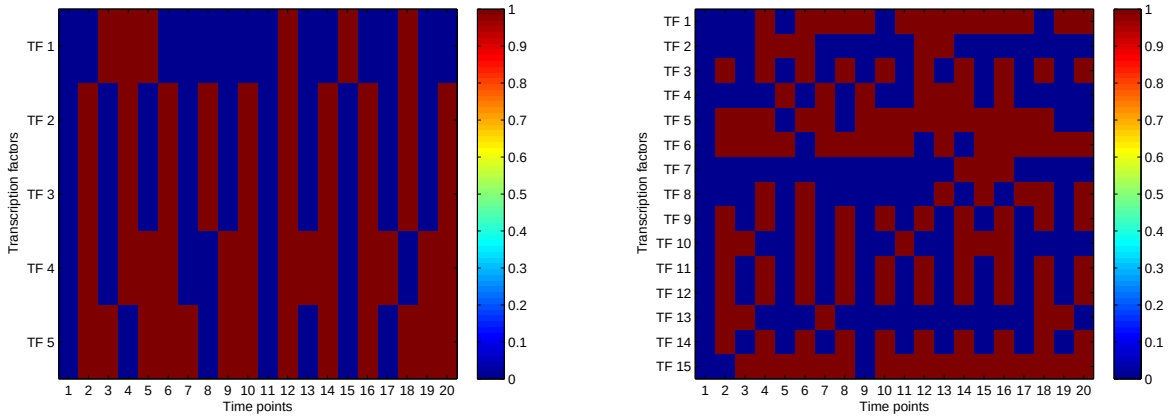


Figure 6.4: Results using simulated dataset 2 (a) Inferred co-occurrence matrix for simulated dataset 2 (b) Posterior probability distribution over number of clusters inferred from simulated dataset 2 (c) Co-occurrence matrix constructed from known cluster assignments for simulated dataset 2 (d) Co-occurrence matrix constructed using K-means algorithm based on the inferred TF profiles from FHMM.



(a) (b)

Figure 6.5: (a) True TFs profiles for simulated dataset 1 (b) True TFs profiles for simulated dataset 2

posteriori splitting of clusters explains the high probabilities for 4 or 5 clusters in the histogram in figure 6.4b.

The results of cluster membership obtained from K-means algorithm (with $K = 3$) using TF profiles inferred from FHMM are shown in figure 6.4d in the form of co-occurrence matrix. As the number of samples in this dataset is higher (*i.e.* $M = 15$ compared to simulated dataset #1 with $M = 5$) the co-occurrence matrix in figure 6.4d shows cluster membership which is in agreement with the original co-occurrence matrix in figure 6.4c except for TF 10 which is not co-clustered correctly by K-means algorithm. Our proposed method also shows weak co-occurrence probability for TF 10 as shown in figure 6.4a.

6.4.2 Sensitivity Analysis

We conducted a thorough sensitivity analysis of the proposed model to see how it responds to different levels of noise in the measurements of gene expression data. As we compare the results of the inference on the proposed model with standard FHMM, this would also allow us gauge the accuracy of inference in two models; namely DPM-FHMM and FHMM.

To achieve this, we use four simulated datasets with the statistics given below:

- Simulated dataset # 1: $N = 100, M = 15, T = 20, K = 3$ with $\sigma^2 = \{0.1, 0.5\}$.
- Simulated dataset # 2: $N = 200, M = 30, T = 20, K = 5$ with $\sigma^2 = \{0.1, 0.5\}$.

We trained three methods on these four datasets: DPM-FHMM, standard FHMM where the transition rates for latent Markov chains are also inferred and FHMM where the transition

Criteria	$\sigma^2 = 0.1$			$\sigma^2 = 0.5$		
	FHMM (with DPM)	**FHMM (with rate learning)**	**FHMM (with fixed rates)**	**FHMM (with DPM)**	**FHMM (with rate learning)**	**FHMM (with fixed rates)**
MSE*	0.0860	0.0859	0.0860	0.4367	0.4537	0.4425
HD*	0.0067	0.0167	0.0067	0.0867	0.1667	0.0433
MSE**	0.0878	0.0884	0.0887	0.4405	0.4430	0.4448
HD**	0.0200	0.0433	0.0367	0.0650	0.0933	0.0467

HD*:HD on simulated dataset #1, MSE*:MSE on simulated dataset #1
HD**:HD on simulated dataset #2, MSE**:MSE on simulated dataset #2

Table 6.2: Comparison of DPM-FHMM, FHMM with transition rate learning and FHMM with transition rates fixed to true values

rates are kept fixed to the ground truth. The inference in FHMM is done via Gibbs sampling. As before, we used MSE and HD to find the deviation between the inferred values of model parameters and latent variables with the ground truth. The results of the inference on these simulated datasets are summarised in table 6.2.

It can be seen from table 6.2 that the predictions of our model are in closer agreement with the ground truth compared to the predictions of FHMM (where the transition rates are also inferred) in terms of HD. This improvement in inferring the latent Markov chains can be seen in both datasets. From this, we can conclude that our model's ability to explain the data is better (even in the presence of relatively large measurement errors) compared to FHMM when the size of the problem is large which is the case for most of the biological system with hundreds of thousands of genes and and hundreds of TFs.

It is important to mention here that our model is not only better than FHMM in learning the temporal profiles of TFs but it also infers the cluster membership of TFs which a standard FHMM cannot do.

6.4.3 Micro-aerobic Shift in *E.coli*

Partridge et al. (2007) studied the changes in transcriptomic behaviour of *E.coli* against the oxidative stress. *E.coli* responds to changes from aerobic to micro-aerobic conditions by activating TF proteins that act as oxygen sensors such as FNR and ArcA. This study measures the mRNA expression profiles of 302 genes and employed a probabilistic technique (Sanguinetti et al., 2006) to infer the activities of the key regulators involved in oxidative stress response in *E.coli*. The analysis reveals the biologically plausible results about the activations patterns of these regulators.

The mRNA expression data consists of 4 time-points taken at 5, 10, 15 and 60 minutes and

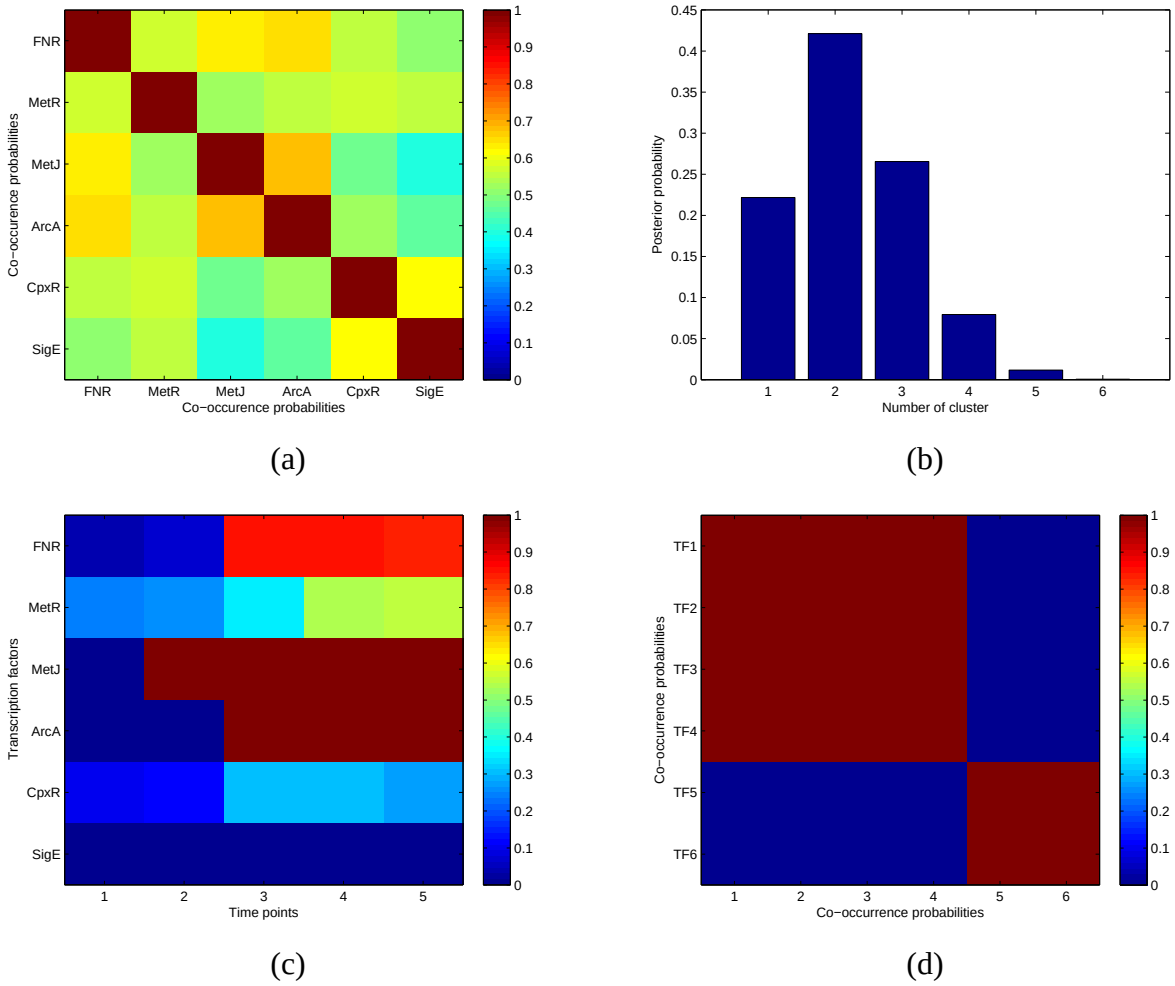


Figure 6.6: (a) Inferred co-occurrence matrix from Partridge et al. (2007) dataset (b) Posterior probability distribution over number of clusters (c) Inferred temporal profiles of six TFs (d) Co-occurrence matrix constructed using K-means algorithm based on the inferred TF profiles from FHMM.

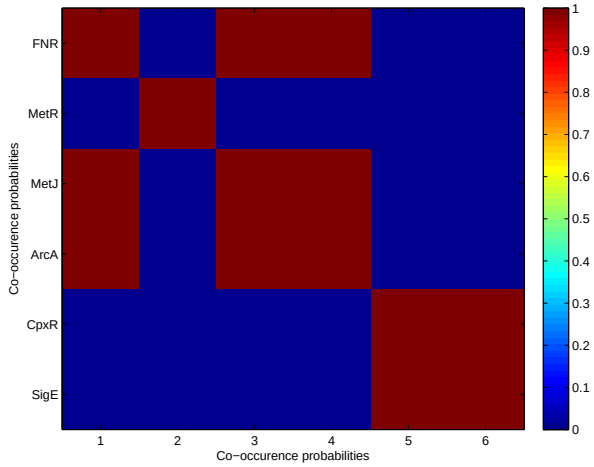


Figure 6.7: Co-occurrence matrix constructed using K-means algorithm (with $K = 3$) based on the inferred TF profiles from FHMM for Partridge et al. (2007) dataset.

measured relative to a sample taken immediately before the oxygen starvation. The connectivity information about the regulatory network of *E.coli* was obtained from ecocyc[1] database. We used this dataset to reconstruct the regulatory mechanism (TF temporal profiles and strength of genetic interactions) and cluster the dynamics of these key regulators. Table 6.1 shows the predictions of our method in comparison with Shi et al. (2008).

The co-occurrence matrix in figure 6.6a shows higher co-occurrence probabilities for TFs that behave similarly by switching to on states to respond to oxidative stress such as FNR and ArcA. These two TFs are known as direct and indirect sensors of oxygens respectively (Partridge et al., 2007). Another TF which is co-clustered with FNR and ArcA is MetJ; this is due the key role of MetJ in methionine biosynthesis which is interrupted during the adaption to aerobic conditions (Partridge et al., 2007). Figure 6.6b shows the posterior probability distribution for different number of clusters. It can be seen that the proposed method predicts two clusters of TFs with highest probability. The second cluster consists of TFs which are not following a well-defined pattern (MetR, SigE, CpxR). A higher probability for a total of 3 clusters can be explained by the examining the profile of MetR which is slightly different than CpxR and SigE in the second cluster. These groups of TFs can be more useful when combined with the experimental setup (such as environmental perturbation during the full length of experiment) to see how perturbations are related to the dynamics of TFs clustered together.

The results of K-means clustering (with $K = 2$) using TF profiles inferred from FHMM are shown in figure 6.6d where the first four TFs are co-clustered while the remaining two TFs

[1] www.ecocyc.org

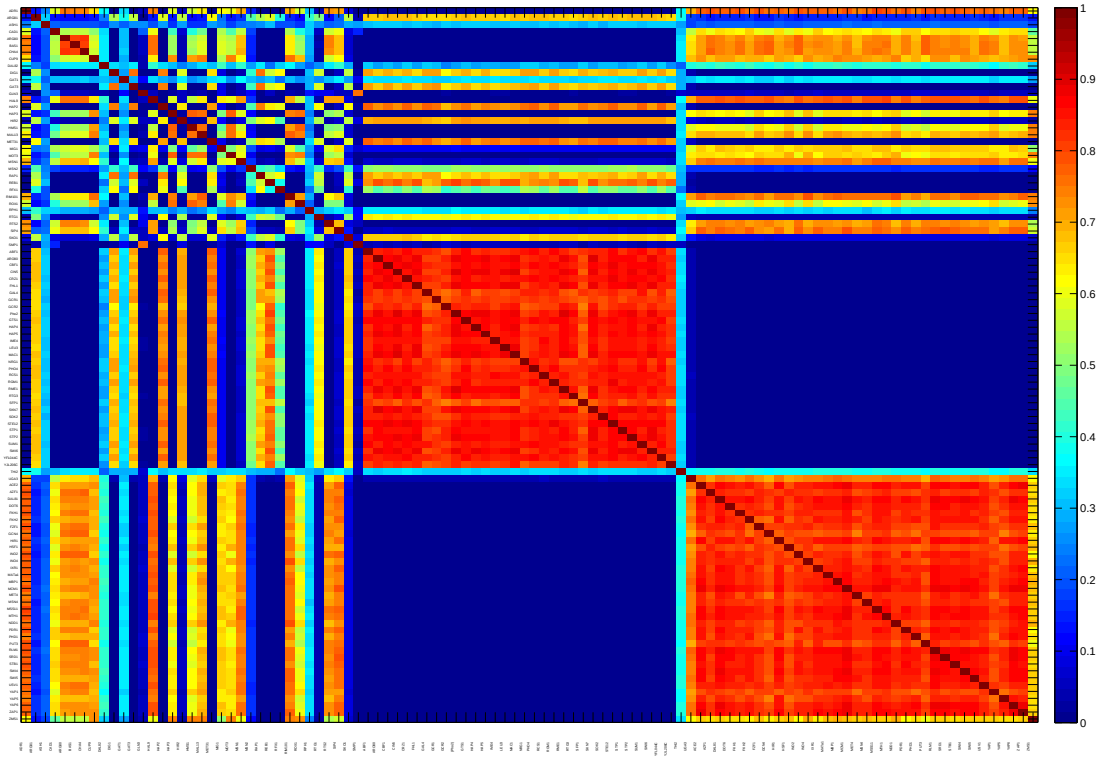
in another cluster. While this is similar to the inference results of our model (figure 6.6a), K-means algorithm with $K = 3$ gives results that are in close agreement with the co-occurrence matrix inferred by our proposed method. This co-occurrence matrix with $K = 3$ is shown in figure 6.7.

6.4.4 Yeast Cell Cycle Data

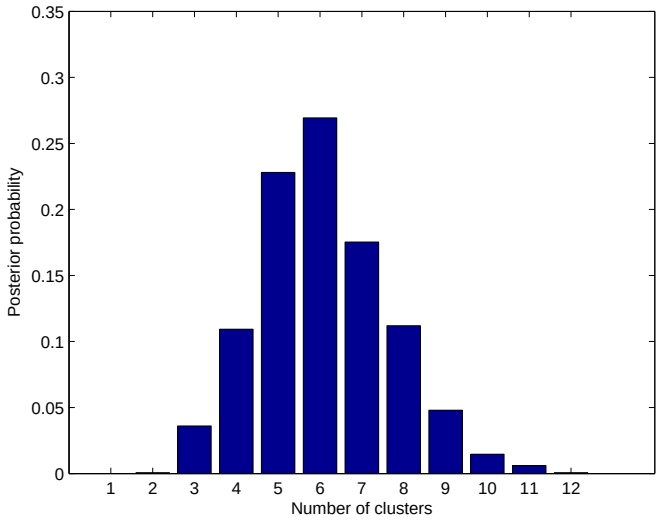
Yeast cell cycle dataset (Spellman et al., 1998) provides the expression profiles of most of the genes in yeast over a complete cell cycle. Although this dataset is old, it is well suited for models of transcription regulation as it is well studied and serves as a standard benchmark for comparison and validation of the model. In this study, three different time-series experiments were conducted on three strains of yeast and these experiments were synchronised by three independent methods. Here, we will focus on the cdc15 synchronised time-series expression data, consisting of 6181 genes expression profiles over 24 equally spaced time-points. To obtain the connectivity of GRN of yeast, we turned to Lee et al. (2002) where this information is available for 113 TFs and 6270 genes. We preprocessed both datasets in such a way that each gene is bound by at least one TF and all TFs regulate at least one downstream target. If, for a gene, no regulator is available we remove the expression profile of that gene to make both the datasets consistent. This preprocessing leaves us 1975 gene expression profiles with a network connectivity information of 1975 genes regulated by 104 TFs.

The histogram in figure 6.8b shows that the dynamics of 104 TFs are best explained when clustered in 6 clusters. We used a threshold of 0.8 for co-occurrence probabilities to find clusters of TFs and rearranged the rows of inferred co-occurrence matrix such that TFs with high co-occurrence probabilities fall together. This co-occurrence matrix is shown in figure 6.8a.

As it can be seen from figure 6.8a, most of the TFs are grouped in two large clusters. The cluster at the lower right corner of figure 6.8a accounts for those TFs that follow a periodic pattern which are ACE2, SWI4, SWI5, MBP1, STB1, FKH1, FKH2, NDD1, MCM1 and few more. These results are consistent with Lee et al. (2002) where these key regulators are identified as co-expressed through the cell cycle and play an important role in cell division. Our model clusters all the key cell cycle regulators identified in Lee et al. (2002) except SKN7. Furthermore, some TFs (DAL81, INO2, INO4, MET4, MSN4, YAP5, YAP6) with similar dynamics are identified in the same cluster suggesting that they may also play a role in regulating the cell cycle; this hypothesis can be tested with evidence from biological experiments. Another large cluster groups together those TFs that are not following a well defined pattern. One small cluster consists of only 3 regulators at the top left of the co-occurrence matrix which



(a)



(b)

Figure 6.8: (a) Inferred co-occurrence matrix from Spellman et al. (1998) dataset (b) Posterior probability distribution over number of clusters.

remain in the on state throughout the cell cycle.

6.5 Conclusion

We introduce a probabilistic method to infer and cluster TF activities based on their latent dynamics by combining the gene expression data with ChIP-on-chip data. The motivation for clustering TF activities is twofold: first of all, biological considerations indicate that, as TF activation can be achieved using a finite set of mechanisms, different TFs may indeed have very similar dynamics of activation. Secondly, the clustering permits a principled Bayesian estimation of the transition probabilities of the underlying Markov chains, which is otherwise extremely hard given the short time series usually available in biology. Using time-series data to identify groups of Markov chains with model-based clustering (Fraley and Raftery, 2002) provides a natural way to model (short) time-series data, arising in a multitude of different applications. Different methods have been proposed for this task (Ramoni et al., 2002; Pamminger and Frühwirth-Schnatter, 2010), mostly based on finite mixture of first order, time homogeneous Markov chains. Although these methods perform well on some applications, selecting the number of clusters remains an issue in many cases, and heuristics such as *AIC*, *BIC* can be problematic, and fail to quantify the uncertainty in this crucial modelling step. To our knowledge, the solution we present, based on non-parametric Bayesian mixture modelling, is a novel and elegant way of addressing this problem. Nonparametric Bayesian methods have been popular in the machine learning and statistics community in recent years, and have been used in time series modelling. In particular, a recent paper (Van Gael et al., 2009) discussed the use of nonparametric Bayesian models in FHMMs. There, however, the nonparametric limit was used to allow the number of factors to be unknown; in our case, the nonparametric prior is one step further up in the hierarchy, and is used to group different factors in an unknown number of clusters. In the bioinformatics literature, Savage et al. (2010) also used nonparametric Bayesian methods to model jointly gene expression and ChIP-on-chip data to find transcription modules; however, in that paper the role of regulation by TF proteins was left implicit, and dynamical models were not considered.

We believe the encouraging results presented indicate that this methodology may be a useful data modelling and exploration tools. In the future, we would like to include clustering ideas in more complex and realistic models of regulation which allow non-linear regulation (Asif and Sanguinetti, 2011; Opper and Sanguinetti, 2010).

Chapter 7

Future Directions

In this thesis, we proposed to use latent variables to model the activities of TF profiles using the observed characteristics of biological networks. We used three methods for this: SSM, cFHMM and DPM-FHMM.

State space models have previously been for inference in transcriptional regulation; however, we exploit the sparse structure of the regulatory network in modelling latent TFAs and gene-specific regulatory activities. This leads to computationally efficient algorithms that can be used on genome-wide scale unlike previous methods. We extended this method and developed a customised software package that is easy to use without any expertise; this software is being used by biologist as an analysis tool to make predictions about the TF activities that are extremely difficult to measure.

We proposed cFHMM to model the non-linear, pair-wise interaction of TFs from gene expression in chapter 5. This method provides novel biological insights as well as confirming the previously known combinatorial interactions. Although latent variables can cope with the post-transcriptional and translational modifications to mRNA, this method does not explicitly model these modifications. Another method based on FHMM (Shi et al., 2008) has been proposed recently that models the post-transcriptional and transcriptional modifications as well. It seems natural to combine these approaches that can provide biologically useful information about two important but different aspects of regulatory activities in gene regulatory network; combinatorial transcriptional regulation and post-transcriptional and translational modifications.

One challenging problem of inference in latent Markov chains from limited length of observed sequences is that of estimating the transition rates of the Markov chains. This becomes even severe in case of biological sequences as the expression data are usually limited to few time-points due to cost of the experimental setup. To address this issue, we propose to use sufficient statistics from multiple Markov chains (TF profiles) in estimating the transition rates instead of single Markov chain. This scheme has shown to provide better estimates of the latent

profiles. This approach has an added benefit; data from similar TF profiles is pooled together for estimating the transition rates which naturally leads to clustering of TF profiles. This approach does not require any assumptions about the initial transition rates as these are inferred. Plausibly, this estimation scheme will lead to better results when used in models where initial transition rates are kept fixed (Asif and Sanguinetti, 2011).

An important feature of the models proposed in this thesis is the large-scale learning and inference which requires that only realistic or somewhat simplified models of transcriptional regulation can be considered. It would be natural to use the clustering scheme proposed in chapter 6 to more realistic models of transcriptional regulation (Opper and Sanguinetti, 2010).

DPM-FHMM clusters the TF profiles based on the TF dynamics; the temporal structure of these latent profiles is not taken into account at the top level of the hierarchy. It would be interesting to see how TF profiles are clustered if the temporal structure of latent TF profiles is considered while clustering these profiles. Intuitively, clustering based on TF profiles will provide clusters of TFs corresponding to their role in particular biological processes.

Appendix A

Calculations for Inference in Combinatorial Transcriptional Regulation with non Time-series Data

A.1 Gibbs Sampler

We presented the inference mechanism for the dynamic case of the combinatorial transcriptional regulation model in section (5.3). Here we describe the static case of the model where the expression data is not from a time-series microarray experiment.

For Gibbs sampling, conditional posterior distribution over $\theta_{\mathbf{i}}$ can be written by using the conditional independence properties of graphical models (Bishop, 2006). It is given by

$$p(\theta_{\mathbf{i}}|g_i^t, \mathbf{TF}_t, \mathbf{X}) = \frac{\prod_{t=1}^{T} \mathcal{N}(g_i^t|\mu_i(\mathbf{TF}_t), \sigma^2).p(\theta_{\mathbf{i}}|\alpha^2)}{\sum_{\theta_{\mathbf{i}}} \left[\prod_{t=1}^{T} \mathcal{N}(g_i^t|\mu_i(\mathbf{TF}_t), \sigma^2).p(\theta_{\mathbf{i}}|\alpha^2)\right]} \tag{A.1}$$

Here $\mathbf{TF}_t$ is the set of states of all the TFs at experimental condition t; for $M = 2$, $\mathbf{TF}_t \in [(0,0),(0,1),(1,0),(1,1)]$. $\mu_i(\mathbf{TF}_t)$ is given in equation (5.2). Simplifying (A.1) leads to a Normal distribution for posterior update of θ_i with the following mean and covariance

$$\Sigma_i^{-1} = \frac{1}{\sigma^2} \sum_{t=1}^{T} \mathbf{X}_i^T \mathbf{e}_t \mathbf{e}_t^T \mathbf{X}_{\mathbf{i}} + \frac{1}{\alpha}^2 [I] \tag{A.2}$$

$$\mu_i = \frac{1}{\sigma^2} \left[\sum_{t=1}^{T} g_i^t \mathbf{e}_t^T \mathbf{X}_{\mathbf{i}}\right] \Sigma_i \tag{A.3}$$

After updating the μ_i and Σ_i for all genes, $\theta_{\mathbf{i}}$ is sampled from a multivariate normal distribution using μ_i and Σ_i.

Again using Bayes' rule, we can write posterior distribution over $\mathbf{TF}_t$ as

$$p(\mathbf{TF}_t|g_i^t,\theta_\mathbf{i},\mathbf{X}) = \frac{\prod_{i=1}^{N}\mathcal{N}(g_i^t|\mu_i(\mathbf{TF}_t),\sigma^2).p(\mathbf{TF})}{\sum_{\mathbf{TF}_t}\prod_{i=1}^{N}\mathcal{N}(g_i^t|\mu_i(\mathbf{TF}_t),\sigma^2).p(\mathbf{TF})} \tag{A.4}$$

$p(\mathbf{TF})$ is taken to be uniform ($\frac{1}{2^M}$). At each condition t there are total 2^M posterior probabilities corresponding to 2^M possible states. Each of the probabilities $(p_1, p_2, \ldots, p_{2^M})$ corresponds to one of the 2^M possible states of the posterior distribution at condition t.

A.2 Variational Inference

Using Gibbs sampler for inference is expensive in terms of the computational time as the conditional posterior distribution is sampled from the joint distribution. Variational formulation of the same model gives comparable results to the of Gibbs sampler and it is computationally efficient than MCMC techniques. The joint likelihood of the model is

$$p(g_i^t,\theta_\mathbf{i},\mathbf{TF}_t,\sigma^2,\alpha^2) = p(g_i^t|\mathbf{TF}_t,\theta_\mathbf{i},\sigma^2)\ p(\theta_\mathbf{i}|\alpha^2)$$

where

$$p(g_i^t|\mathbf{TF}_t,\theta_\mathbf{i},\sigma^2) = \prod_{i=1}^{N}\prod_{t=1}^{T}\mathcal{N}(g_i^t|\mu(\mathbf{TF}),\sigma^2)$$

and

$$p(\theta_i|\alpha^2) = \prod_{i=1}^{N}\mathcal{N}(0,\alpha^2)$$

Taking the expectation of the log of the joint likelihood w.r.t. $\theta_\mathbf{i}$ gives the posterior distribution over the parameters of θ_i

$$q^*(\theta_i) = \prod_{i=1}^{N}\mathcal{N}(\theta_i|\mathbf{m},\Sigma)$$

The mean and covariance of this multivariate Normal distribution are given by

$$\Sigma^{-1} = \frac{1}{\sigma^2}\sum_{t=1}^{T}\mathbf{X}_i\left\langle\mathbf{e}_t\mathbf{e}_t^T\right\rangle\mathbf{X}_i + \alpha^{-2}\mathbf{I}$$

$$\mathbf{m} = \frac{1}{\sigma^2}\left[\sum_{t=1}^{T}g_i^t\left\langle\mathbf{e}_t^T\right\rangle\mathbf{X}_i\right]\Sigma^{-1}$$

Again taking the expectation of the log of joint likelihood w.r.t. $\mathbf{e}_t$ gives posterior distribution over the states of all the TFs at condition t

$$\ln q^*(\mathbf{TF}_t) = -\frac{1}{2}\sum_{i=1}^{N}\left\{\sum_{t=1}^{T}\left(\frac{1}{\sigma^2}\mathbf{e}_t^T\mathbf{X}_i\left\langle\theta_i\theta_i^T\right\rangle\mathbf{X}_i\mathbf{e}_t - \frac{2}{\sigma^2}g_i^t\mathbf{e}_t^T\mathbf{X}_i\left\langle\theta_i\right\rangle\right)\right\}$$

$$q^*(\mathbf{TF}_t) = \exp\left[-\frac{1}{2}\sum_{i=1}^{N}\left\{\sum_{t=1}^{T}\left(\frac{1}{\sigma^2}\mathbf{e}_t^T\mathbf{X}_i\left\langle\theta_i\theta_i^T\right\rangle\mathbf{X}_i\mathbf{e}_t - \frac{2}{\sigma^2}g_i^t\mathbf{e}_t^T\mathbf{X}_i\left\langle\theta_i\right\rangle\right)\right\}\right]$$

Bibliography

Antoniak, C. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, pages 1152–1174.

Asif, A. and Moura, J. M. F. (2005). Block matrices with L-block-banded inverse: inversion algorithms. *IEEE Transactions on Signal Processing*, 53:630–642.

Asif, H. M. S. (2010). TFInfer user manual. `http://homepages.inf.ed.ac.uk/s0976841/TFInfer/TFInferHelp.html`.

Asif, H. M. S., Rolfe, M., Green, J., Lawrence, N., Rattray, M., and Sanguinetti, G. (2010). TFInfer: a tool for probabilistic inference of transcription factor activities. *Bioinformatics*, 26(20):2635.

Asif, H. M. S. and Sanguinetti, G. (2009). Probabilistic inference of transcription factor concentrations and gene-specific regulatory activities for time-independent data. *Supp. Proc. of Prib2009*.

Asif, H. M. S. and Sanguinetti, G. (2011). Large-scale learning of combinatorial transcriptional dynamics from gene expression. *Bioinformatics*, 27(9):1277.

Barenco, M., Tomescu, D., Brewer, D., Callard, R., Stark, J., and Hubank, M. (2006). Ranked prediction of p53 targets using hidden variable dynamical modelling. *Genome Biology*, 7(3).

Beal, M. (2003). *Variational algorithms for approximate Bayesian inference*. University of London.

Beal, M., Falciani, F., Ghahramani, Z., Rangel, C., and Wild, D. (2005). A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 21(3):349–356.

Bhoite, L. T., Allen, J. M., Garcia, E., Thomas, L. R., Gregory, I. D., Voth, W. P., Whelihan, K., Rolfes, R. J., and Stillman, D. J. (2002). Mutations in the Pho2 (Bas2) Transcription

Factor That Differentially Affect Activation with Its Partner Proteins Bas1, Pho4, and Swi5. *Journal of Biological Chemistry*, 277(40):37612–37618.

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer New York.

Blackwell, D. (1947). Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, pages 105–110.

Boyer, L., Lee, T., Cole, M., Johnstone, S., Levine, S., Zucker, J., Guenther, M., Kumar, R., Murray, H., Jenner, R., et al. (2005). Core transcriptional regulatory circuitry in human embryonic stem cells. *Cell*, 122(6):947–956.

Boys, R., Henderson, D., and Wilkinson, D. (2000). Detecting homogeneous segments in DNA sequences by using hidden Markov models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 49(2):269–285.

Caspi, R., Foerster, H., Fulcher, C., Kaipa, P., Krummenacker, M., Latendresse, M., Paley, S., Rhee, S., Shearer, A., Tissier, C., et al. (2008). The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic acids research*, 36(suppl 1):D623–D631.

Celeux, G., Hurn, M., and Robert, C. (2000). Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association*, pages 957–970.

Coffman, J., Rai, R., Cunningham, T., Svetlov, V., and Cooper, T. (1996). Gat1p, a GATA family protein whose production is sensitive to nitrogen catabolite repression, participates in transcriptional activation of nitrogen-catabolic genes in *Saccharomyces cerevisiae*. *Molecular and cellular biology*, 16(3):847.

Cowles, M. and Carlin, B. (1996). Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, pages 883–904.

Dahl, D. (2006). Model-based clustering for expression data via a Dirichlet process mixture model. *Bayesian Inference for Gene Expression and Proteomics*, pages 201–218.

Daignan-Fornier, B. and Fink, G. (1992). Coregulation of purine and histidine biosynthesis by the transcriptional activators BAS1 and BAS2. *Proceedings of the National Academy of Sciences of the United States of America*, 89(15):6746.

Davidge, K., Sanguinetti, G., Yee, C., Cox, A., McLeod, C., Monk, C., Mann, B., Motterlini, R., and Poole, R. (2009). Carbon monoxide-releasing antibacterial molecules target respiration and global transcriptional regulators. *Journal of Biological Chemistry*, 284(7):4516.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38.

Ferguson, T. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, pages 209–230.

Fraley, C. and Raftery, A. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

Frühwirth-Schnatter, S. (2001). Markov chain Monte Carlo estimation of classical and dynamic switching and mixture models. *Journal of the American Statistical Association*, 96(453):194–209.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian data analysis*. Chapman and Hall/CRC, London.

Ghahramani, Z. and Jordan, M. (1997). Factorial hidden Markov models. *Machine Learning*, 29(2):245–273.

Gibbons, J. and Chakraborti, S. (2003). *Nonparametric statistical inference*, volume 168. CRC Press.

Gnedin, A. and Kerov, S. (2001). A characterization of gem distributions. *Combinatorics, Probability and Computing*, 10(03):213–217.

Graham, A., Sanguinetti, G., Bramall, N., McLeod, C., and Poole, R. (2011). Dynamics of a starvation-to-surfeit shift: A transcriptomic and modelling analysis of the bacterial response to zinc reveals transient behaviour of the fur and soxs regulators. *Microbiology*.

Hahn, S., Pinkham, J., Wei, R., Miller, R., and Guarente, L. (1988). The HAP3 regulatory locus of *Saccharomyces Cerevisiae* encodes divergent overlapping transcripts. *Molecular and cellular biology*, 8(2):655.

Harbison, C., Gordon, D., Lee, T., Rinaldi, N., Macisaac, K., Danford, T., Hannett, N., Tagne, J., Reynolds, D., Yoo, J., et al. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431:99–104.

Holmes, C. (2010). *Bayesian nonparametrics*, volume 28. Cambridge Univ Pr.

Jaynes, E., Bretthorst, G., and MyiLibrary (2003). *Probability theory: the logic of science*. Cambridge University Press Cambridge.

Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

Karp, P., Riley, M., Saier, M., Paulsen, I., Collado-Vides, J., Paley, S., Pellegrini-Toole, A., Bonavides, C., and Gama-Castro, S. (2002). The ecocyc database. *Nucleic acids research*, 30(1):56.

Keseler, I., Collado-Vides, J., Santos-Zavaleta, A., Peralta-Gil, M., Gama-Castro, S., Muñiz-Rascado, L., Bonavides-Martinez, C., Paley, S., Krummenacker, M., Altman, T., et al. (2011). EcoCyc: a comprehensive database of *Escherichia Coli* biology. *Nucleic Acids Research*, 39(suppl 1):D583.

Kitano, H. (2002). Computational systems biology. *Nature*, 420(6912):206–210.

Lawrence, N., Girolami, M., and Rattray, M. (2010). *Learning and inference in computational systems biology*. The MIT Press.

Lawrence, N. D., Sanguinetti, G., and Rattray, M. (2006). Modelling transcriptional regulation using Gaussian processes. In *Advances in Neural Information Processing Systems 19*.

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., Zeitlinger, J., Jennings, E. G., Murray, H. L., Gordon, D. B., Ren, B., Wyrick, J. J., Tagne, J.-B., Volkert, T. L., Fraenkel, E., Gifford, D. K., and Young, R. A. (2002). Transcriptional Regulatory Networks in *Saccharomyces Cerevisiae*. *Science*, 298(5594):799–804.

Liao, J., Boscolo, R., Yang, Y., Tran, L., Sabatti, C., and Roychowdhury, V. (2003). Network component analysis: reconstruction of regulatory signals in biological systems. *Proceedings of the National Academy of Sciences*, 100(26):15522–15527.

Mann, T. (2006). Numerically stable hidden Markov model implementation. *An HMM scaling tutorial*.

McAdams, H. and Arkin, A. (1997). Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences*, 94(3):814.

McLachlan, G. and Krishnan, T. (2008). *The EM algorithm and extensions*, volume 382. John Wiley and Sons.

Medvedovic, M. and Sivaganesan, S. (2002). Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics*, 18(9):1194.

Morris, J., Apeltsin, L., Newman, A., Baumbach, J., Wittkop, T., Su, G., Bader, G., and Ferrin, T. (2011). clustermaker: a multi-algorithm clustering plugin for cytoscape. *BMC Bioinformatics*, 12(1):436.

Müller, P. and Quintana, F. (2004). Nonparametric Bayesian data analysis. *Statistical science*, pages 95–110.

Nachman, I. (2004). *Probabilistic modeling of gene regulatory networks from data*. PhD thesis, Hebrew University of Jerusalem.

Nachman, I., Regev, A., and Friedman, N. (2004). Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, 20 Suppl 1.

Opper, M. and Sanguinetti, G. (2010). Learning combinatorial transcriptional dynamics from gene expression data. *Bioinformatics*.

Pamminger, C. and Frühwirth-Schnatter, S. (2010). Model-based clustering of categorical time series. *Bayesian Analysis*, 5(2):345–368.

Partridge, J., Sanguinetti, G., Dibden, D., Roberts, R., Poole, R., and Green, J. (2007). Transition of *Escherichia Coli* from aerobic to micro-aerobic conditions involves fast and slow reacting regulatory components. *Journal of Biological Chemistry*, 282(15):11230–11237.

Ptashne, M. and Gann, A. (2002). *Genes & signals*. CSHL Press.

Quackenbush, J. et al. (2002). Microarray data normalization and transformation. *nature genetics*, 32(supp):496–501.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Ramoni, M., Sebastiani, P., and Cohen, P. (2002). Bayesian clustering by dynamics. *Machine Learning*, 47(1):91–121.

Rasmussen, C. (2000). The infinite Gaussian mixture model. *Advances in Neural Information Processing Systems*, 12:554–560.

Rogers, S., Khanin, R., and Girolami, M. (2007). Bayesian model-based inference of transcription factor activity. *BMC Bioinformatics*, 8(S2).

Rolfe, M. D., Beek, A. T., Graham, A. I., Trotter, E. W., Asif, H. M. S., Sanguinetti, G., de Mattos, J. T., Poole, R. K., and Green, J. (2011). Transcript profiling and Inference of *Escherichia Coli* K-12 ArcA activity across the range of physiologically relevant oxygen concentrations. *Journal of Biological Chemistry*, 286(12):10147–10154.

Sabatti, C. and James, G. (2006). Bayesian sparse hidden components analysis for transcription regulation networks. *Bioinformatics*, 22(6):739.

Sabatti, C. and Lange, K. (2002). Genomewide motif identification using a dictionary model. *Proceedings of the IEEE*, 90(11):1803–1810.

Sanguinetti, G., Lawrence, N., and Rattray, M. (2006). Probabilistic inference of transcription factor concentrations and gene-specific regulatory activities. *Bioinformatics*, 22(22):2775.

Sanguinetti, G., Ruttor, A., Opper, M., and Archambeau, C. (2009). Switching regulatory models of cellular stress response. *Bioinformatics*, 25(10):1280–1286.

Savage, R., Ghahramani, Z., Griffin, J., de la Cruz, B., and Wild, D. (2010). Discovering transcriptional modules by Bayesian data integration. *Bioinformatics*, 26(12):i158.

Scott, S., Abul-Hamd, A., and Cooper, T. (2000). Roles of the Dal82p domains in Allophanate/Oxalurate-dependent gene expression in *Saccharomyces Cerevisiae*. *Journal of Biological Chemistry*, 275(40):30886.

Shi, Y., Simon, I., Mitchell, T., and Bar-Joseph, Z. (2008). A combined expression-interaction model for inferring the temporal activity of transcription factors. In *Research in Computational Molecular Biology*, pages 82–97. Springer.

Smoot, M., Ono, K., Ruscheinski, J., Wang, P., and Ideker, T. (2011). Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, 27(3):431.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces Cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, 9(12):3273–3297.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809.

Teh, Y. (2007). Dirichlet process. *Encyclopedia of Machine Learning*.

Teh, Y., Jordan, M., Beal, M., and Blei, D. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Tipping, M. (2004). Bayesian inference: An introduction to principles and practice in machine learning. *Lecture notes in computer science*, pages 41–62.

Tu, B., Kudlicki, A., Rowicka, M., and McKnight, S. (2005). Logic of the yeast metabolic cycle: temporal compartmentalization of cellular processes. *Science*, 310(5751):1152.

Tuna, S. and Niranjan, M. (2010). Reducing the algorithmic variability in transcriptome-based inference. *Bioinformatics*, 26(9):1185.

Van Gael, J., Teh, Y., and Ghahramani, Z. (2009). The infinite factorial hidden Markov model. *Advances in Neural Information Processing Systems*, 21:1697–1704.

Wikipedia (2012a). Chip-on-chip — Wikipedia, the free encyclopedia. [Online; accessed 23-Feb-2012].

Wikipedia (2012b). Dna microarray hybridisation — Wikipedia, the free encyclopedia. [Online; accessed 23-Feb-2012].

Xing, E., Jordan, M., and Russell, S. (2003). A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of the 19th Annual Conference on Uncertainty in AI*.

Xing, Y., Fikes, J., and Guarente, L. (1993). Mutations in yeast HAP2/HAP3 define a hybrid CCAAT box binding domain. *The EMBO Journal*, 12(12):4647.

Xu, Z. and Tsurugi, K. (2007). Role of Gts1p in regulation of energy-metabolism oscillation in continuous cultures of the yeast *Saccharomyces Cerevisiae*. *Yeast*, 24(3):161–170.