Probabilistic graphical models and algorithms for genomic analysis

by

Poe Xing

B.S. (Tsinghua University) 1993
M.S. (Rutgers University) 1998
Ph.D. (Rutgers University) 1999

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Richard Karp, co-Chair
Professor Michael Jordan, co-Chair
Professor Stuart Russell, co-Chair
Professor Gene Myers
Professor Terry Speed

Fall 2004

The dissertation of Poe Xing is approved:

co-Chair Date

co-Chair Date

co-Chair Date

Date

Date

University of California, Berkeley

Fall 2004

Probabilistic graphical models and algorithms for genomic analysis

by

Poe Xing

Abstract

Probabilistic graphical models and algorithms for genomic analysis

by

Poe Xing

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Richard Karp, co-Chair

Professor Michael Jordan, co-Chair

Professor Stuart Russell, co-Chair

In this thesis, I discuss two probabilistic modeling problems arising in metazoan genomic analysis: identifying motifs and *cis*-regulatory modules (CRMs) from transcriptional regulatory sequences, and inferring haplotypes from genotypes of single nucleotide polymorphisms. Motif and CRM identification is important for understanding the gene regulatory network underlying metazoan development and functioning. I discuss a modular Bayesian model that captures rich structural characteristics of the transcriptional regulatory sequences and supports a variety of motif detection tasks. Haplotype inference is essential for the understanding of genetic variation within and among populations, with important applications to the genetic analysis of disease propensities. I discuss a Bayesian model based on a prior distribution constructed from a Dirichlet process – a nonparametric prior which provides control over the size of the unknown pool of population haplotypes, and on a likelihood function that allows statistical errors in the haplotype/genotype relationship. Our models use the "probabilistic graphical model" formalism, a formalism that exploits the conjoined capabilities of graph theory and probability theory to build complex models out of simpler pieces. I discuss the mathematical underpinnings for the models, how they formally incorporate biological prior knowledge about the data, and I present a generalized mean field theory and a generic algorithm for approximate inference on such models.

co-Chair

Date

co-Chair

Date

co-Chair

Date

Dedicate to my wife — Wei

and

to my parents

for encouraging me

to pursue my dream

and

for sharing my joy and frustration

in this endeavor

Acknowledgements

I wish to thank my advisers at Berkeley, Richard Karp, Michael Jordan and Stuart Russell, for their kindness, patience, and cooperativeness in working as a "dream team" along with me to make possible a smooth transformation for me from a novice to a professional in computer science during the past five years, and for giving me so much freedom to discover and explore new subjects in machine learning, statistics and computational biology. I thank Richard Karp for his generous support, invaluable trust, inspiring discussions and insightful suggestions on my research in computational biology, and for being a great friend and a source of encouragement and understanding. I thank Michael Jordan for his great patience and unparalleled technical guidance early in my development, and his inspiration, enthusiasm and encouragement on my research in machine learning. I am also greatly indebted to Stuart Russell, for sharing with me his wisdom and humor, his insightful critiques and stimulating ideas, and for his extensive technical and moral support on my research and career development. A Ph.D. under their mentorship is the experience of a lifetime.

My other committee members have also been very supportive. Gene Myers has been a warm supporter on my endeavor in computational biology, and a source of new problems, new ideas and objective opinions from the non-machine-learning community. Terry Speed inspired my interest in statistical genetics, and has also brought lots of useful outsider's perspective to the thesis. In particular, it was in writing a term paper for one of his classes that I worked out the first piece of this dissertation — the new motif detection model.

I would like to thank my many friends and colleagues at Berkeley with whom I have had the pleasure of working over the years. These include Eyal Amir, David Blei, Nando de Freitas, Bhaskara Marthi, Brian Milch, Erik Miller, Kevin Murphy, Andrew Ng, Xuanlong Nguyen, Mark Paskin, Matthias Seeger, Yee-Whye Teh, Martin Wainwright, Yair Weiss, Andy Zimdars, Alice Zheng, and all the members of the SAIL and RUGS groups. Their encouragement and friendship and their help have brought me incredible joy during my Berkeley days. I particularly want to thank Brian Milch for his critical reading of this thesis, which greatly improved its clarity and readability

(although any remaining errors in the thesis are of course my fault).

I would also like to thank Eric Horvitz, Tanveer Syeda-Mahmood and Jeonghee Yi for hiring me as an intern at Microsoft Research in 2001, as a consultant at IBM Research in 2001, and as an instructor at IBM Research during 2001-2002, respectively, during which I gained valuable experience in industrial R&D and in advanced teaching.

I want to extent my gratitude to other friends in Soda hall, on campus, and beyond for friendship and support. Finally, I wound like to thank my wife Wei for bearing with my countless weekend and late night stays in the office, for listening to my wild ideas and endless details, and for her encouragement on my work with love, patience and understanding.

Contents

Chapter 1

Introduction

Understanding the structure and functional organization of the genome is a fundamental problem in biology. This thesis introduces new computational statistical approaches for analyzing two particular types of genomic data: gene regulatory sequences, and single nucleotide polymorphisms. It presents the methodology of applying the *probabilistic graphical model* formalism to designing novel parametric and non-parametric Bayesian models for genomic data, in accordance with biological prior knowledge or genetic hypotheses about the population of subjects under investigation. In particular, it presents algorithms for the problems of *motif detection* and *haplotype inference*, and develops the general theory and algorithms of *generalized mean field approximation* for *variational inference* on large-scale, hybrid, multivariate probabilistic models.

Although the major goal of this thesis is to develop probabilistic models and computational algorithms for deciphering biological data and exploring the mechanisms and evolution of biological systems based on mathematical principles, most of the ideas and results reported here can also serve as building blocks of generic intelligent systems for a wide range of applications that involve predictive understanding and reasoning under uncertainty.

1.1 Genomic Analysis and the Graphical Model Approach

1.1.1 The Architecture and Function of the Genome

According to the central dogma, the genetic information that determines the functional and morphological properties of the cells in a living organism is encoded in the DNA genome [Crick, 1970]. Biochemically, DNAs are double-stranded macromolecules representable as a pair of long complementary sequences of characters — A, T, G and C, denoting four kinds of basic elements, known as *nucleotides*, that make up the DNA molecules. Residing in (and inherited via) the DNA molecules, are a rich set of coding sequences referred to as *genes*, which determine the structures and functions of an essential set of biopolymer molecules, mostly proteins, but also including RNAs, which are the main determinants of various cellular and physiological activities taking place in a living system, such as biochemical catalysis, signal transduction, cellular defense, etc. [Lewin, 2003]. Also abundant in the DNAs are a large number of so-called non-coding sequences, whose role was originally thought to be purely structural (e.g., serving as the physical scaffold of a *chromosome* — a long thread of DNA tightly packaged with the aid of several auxiliary proteins), but have been recently discovered to play essential roles in the cellular implementation of the *gene regulation network* [Davidson, 2001; Alberts *et al.*, 2002].

DNAs usually reside in the nucleus (or the nuclear region for prokaryotic organisms) of the cell. Via a process called *transcription* (to be explained shortly), some genes in the DNA genome are copied to molecules called messenger RNA (mRNA), which can travel out of the nucleus to the protein synthesis apparatus, where proteins are assembled based on the coding information carried by mRNA via a process called *translation* [Alberts *et al.*, 2002]. Although different cells of an organism have the same DNA genome, it is well known that they have different protein composition and perform different functions [Davidson, 2001]. For example, red blood cells are rich in hemoglobins that can carry oxygen, whereas muscle cells contain a large number of myosins for muscular contraction. Even the same cell may bear different protein contents at different times during its life span. This kind of diversity is a consequence of spatially and temporally regulated

expression of genes. It is believed that much of the information that determines when and in what cellular environment a gene is expressed is encoded in certain genomic sequences, which possibly account for a major portion of the total sequence of the genome, especially in the higher eukaryotes, such as human [Davidson, 2001; Michelson, 2002].

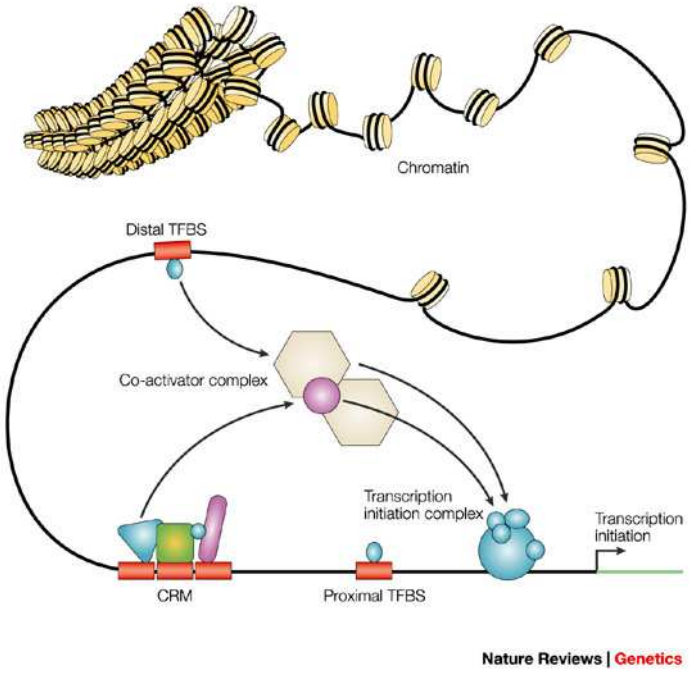


Figure 1.1: The transcriptional regulatory machinery (adapted from [Wasserman and Sandelin, 2004]). TFBS: transcription factor binding site, CRM: *cis*-regulatory module; chromatin: a long, extended thread of DNA packed with histone proteins.

The creation of diverse cell types from an invariant set of genes is governed by complex biochemical processes that regulate gene activities. Transcription, the initial step of gene expression, is central to the regulatory mechanisms. Transcription refers to the process of making a single-stranded mRNA molecule using one of the DNA strands as template. The timing and volume of transcription are controlled by complex transcription regulatory machinery made up of both protein and DNA elements [Ptashne, 1988; Ptashne and Gann, 1997]. As shown in Fig. 1.1, the signals that activate or suppress the transcription of a gene are physically mediated by different types of gene regulatory proteins called *transcription factors* (TFs). To bring these signals into effect on a target

gene at a specific time in a specific cell, certain TFs must recognize specific binding sites in the vicinity of the target gene, so that they can jointly interact with the basal transcription apparatus, made up of an RNA polymerase and some general TFs, to turn on or off transcription in the right physiological/developmental context.

DNA *motifs* are the protein binding sites on DNA sequences that can be recognized by specific TFs to integrate complex gene regulatory signals (hence they are also referred to as transcription factor binding sites, or TFBS). These sites are usually located in the vicinity of the transcription initiation sites of the genes under their regulation — an extended sequence region generally referred to as the *transcriptional regulatory region* [Lewin, 2003]. Depending on which organism the genomic sequences are from, the complexities of the transcriptional regulatory regions vary significantly. Their lengths range from a few hundred base pairs (e.g, in simple bacteria such as E. coli) to over several hundred thousand base pairs (e.g., in more complex insects such as *Drosophila*); their locations can be either immediately proximal to the transcription initiation sites, or much further upstream or even downstream (i.e., into the intron regions of gene sequences); and their contents range anywhere from sparse single-motif-promoters, to multiple complex *cis*-regulatory modules (CRMs) each containing arrays of multiple motifs [Davidson, 2001] (Fig. 1.1). Motifs, together with their specific pattern of deployment (e.g., ordering, contexts) in the genome, constitute the hardwired part of the transcription regulatory machinery, which is present in every cell of an organism, although different subsets of motifs will be involved in gene regulation in different cells. Deciphering the gene control circuitry encoded in DNA, its structure and its functional organization is a fundamental problem in biology, and is a focus of this thesis.

1.1.2 The Populational Diversity and Evolution of the Genome

When the human genome project was launched over a decade ago, there was an interesting debate over who should have the honor (but not without the courage of relinquishing the utmost privacy) to have his/her genome sequenced. One rumor goes that the chief of the Celera company had taken this privilege. This debate struck a key issue in genetics — that at the very sequence level, there exist

individual distinctions and even populational diversities in the DNA genome. This phenomenon is referred to as *genetic polymorphism*.

A polymorphism is a neutral genetic variant that appears in at least 1% of the human population, and does not directly elicit any substantial advantage or disadvantage for the survival of the individual bearing it [Kruglyak and Nickerson, 2001]. Polymorphisms are often regarded as fingerprints of ancestral genetic alterations left on modern genomic sequences during evolution and can serve as genetic markers of population- or disease-related phenotypes [Clark, 2003]. Common polymorphisms include insertion/deletion of minisatellites, microsatellites, Alu segments, etc., which are non-functional DNA segments of various sizes; as well as single nucleotide polymorphisms (SNPs) [Stoneking, 2001].

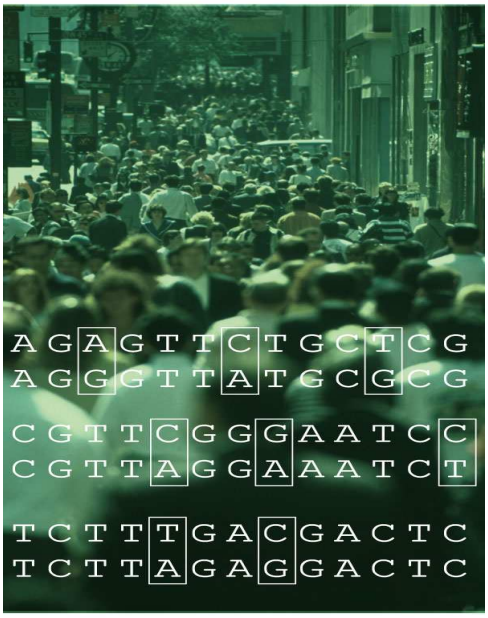


Figure 1.2: Single nucleotide polymorphisms as appeared in two chromosomes from a population (adapted from [Chakravarti, 2001]).

SNP refers to the existence of two possible kinds of nucleotides at a single chromosomal locus in a population; each variant is called an *allele* (Fig. 1.2). SNPs reflect past mutations that were mostly (but not exclusively) unique events, and two individuals sharing a variant allele are thereby marked with a common evolutionary heritage [Patil *et al.*, 2001; Stoneking, 2001]. In other words,

our genes have ancestors, and analyzing shared patterns of SNP variations can identify them. The real importance of SNPs lies in their abundance. It is estimated that there are more than 5 million common SNPs each with frequency 10-50% in the whole human population, which translates to about one SNP in every 600 base pairs in the human genome [Zhang *et al.*, 2002]. These SNPs account for more than 90% of human DNA sequence difference.

As SNPs are remnants of ancient neutral DNA alterations dated back to a time measured at a *genealogical* scale, they contain more fine-grained information on molecular evolution than that revealed by orthologous genomic sequences from multiple species, whose differences are accumulated over a *geological* period of time and are subject to selection. In general, the higher the frequency of a SNP allele, the older the mutation that produced it, so high-frequency SNPs largely predate human population diversification. Therefore, population-specific alleles may bear important information about human evolution that involves specific migrations (such as those that populated Polynesia and the Americas) [Stoneking, 2001].

Most human variation that is influenced by genes can be related to SNPs (either as associated markers or causative elements), especially for such medically (and commercially) important traits as how likely one is to become afflicted with a particular disease, or how one might respond to a particular pharmaceutical treatment, as discussed in [Chakravarti, 2001]. Even when a SNP is not directly responsible, the dense distribution of SNPs in the genome suggests they can also be used to locate genes that influence such traits based on a linkage disequilibrium test (for gametic association between the putative causal gene(s) and SNPs in the vicinity) [Akey *et al.*, 2001; Daly *et al.*, 2001; Pritchard, 2001]. For higher organisms, accurate inferences concerning population history or association studies of disease propensities and other complex traits usually demand the analysis of the states of sizable segments of the subject's chromosome(s) [Kenneth and Clark, 2002]. To this end, it is advantageous to study haplotypes, which consist of several closely spaced (hence linked) SNPs and often prove to be more powerful discriminators of genetic variations within and among populations, and hence serve as more informative markers for linkage analysis and evolutionary studies.

1.1.3 Probabilistic Graphical Models and Genomic Analysis

Due to the stochastic nature of genomic data, and the abundance of empirical biological prior knowledge about their properties, the general methodologies adopted in this thesis are built on probabilistic models that accommodate uncertainty and statistical errors associated with the data, and that incorporate certain prior information in a principled way.

The models we develop in this thesis use a formalism called *probabilistic graphical models* [Pearl, 1988; Cowell *et al.*, 1999; Lauritzen and Sheehan, 2002], which refer to a family of probability distributions defined in terms of a directed or undirected graph with probabilistic semantics (Fig. 1.3).

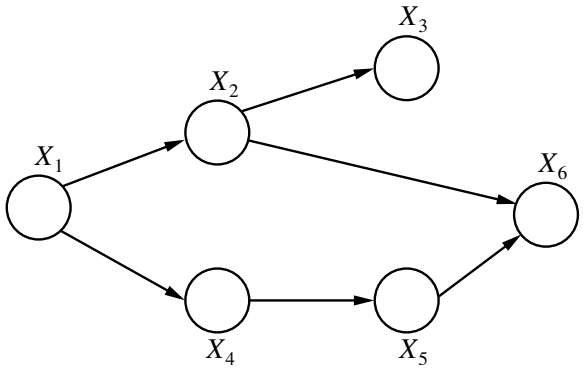


Figure 1.3: A directed graphical model for a joint probability distribution over $\{x_1, x_2, x_3, x_4, x_5, x_6\}$. It entails $p(x_1, x_2, x_3, x_4, x_5, x_6) = p(x_1)p(x_2|x_1)p(x_4|x_1)p(x_3|x_2)p(x_5|x_4)p(x_6|x_2, x_5)$.

A graphical model has both a structural (or topological) component — encoded by a graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges of the graph; and a parametric component — encoded by numerical "potentials" $\{\phi_C(\mathbf{x}_C) : C \subset \mathcal{V}\}$, a set of positive numbers associated with the state configurations of subsets of nodes in the graph. Each node in the graph represents a random variable X_i, which can be either *observed* or *latent*, as indicated by the shading of the node [1]; the presence of edges between nodes denotes direct dependencies between the corresponding variables. Independent and identically distributed (*iid*) random variables can be represented by a macro called a *plate*, which allows a subgraph to be replicated. For example, the assertion that

[1] In the sequel, we use upper-case X (resp. $\mathbf{X}$) to denote a random variable (resp. variable set), and lower-case x (resp. $\mathbf{x}$) to denote a certain state (or value, configuration, etc.) taken by the corresponding variable (resp. variable set).

variables $\{X_i\}$ are conditionally *iid* given θ can be represented by a plate over X_i (Fig.1.4a). The family of joint probability distributions associated with a given graph can be parameterized in terms of a product over potential functions associated with subsets of nodes in the graph. For directed graphical models (associated with acyclic directed graphs), which are often referred to as *Bayesian networks*, each node, X_i, and its parents, $\mathbf{X}_{\pi_i}$, constitute the basic subset on which a potential function is defined, and the potential function turns out to be the *local conditional probability* $p(x_i|\mathbf{x}_{\pi_i})$. Hence, we have the following representation for the joint probability:

$$p(\mathbf{x}) = \prod_{i \in \mathcal{V}} p(x_i|\mathbf{x}_{\pi_i}). \tag{1.1}$$

For undirected graphical models, which are often referred to as *Markov random fields*, the basic subsets are *cliques* (completely connected subsets of nodes) of the graph, $\{\mathbf{X}_{D_\alpha} : \alpha \in \mathcal{A}\}$, where D_α denotes the set of node indices of clique α, and $\mathcal{A}$ denotes the index set of all cliques. The joint probability in this case is:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{\alpha \in \mathcal{A}} \phi_\alpha(\mathbf{x}_{D_\alpha}), \tag{1.2}$$

where Z is a normalizing constant, ensuring that $\int p(\mathbf{x}) d\mathbf{x} = 1$ (or $\sum_{\mathbf{x}} p(\mathbf{x}) = 1$ for discrete models).

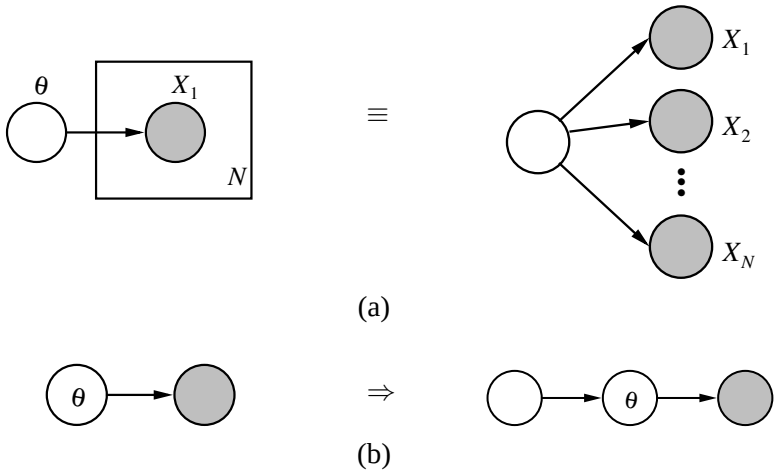


Figure 1.4: Various graphical models. Shaded nodes denote observed variables. (a) Plate. (b) From a flat parametric model to a Bayesian model.

Graphical models provide a compact graph-theoretic representation of probabilistic distributions in a way that clearly exposes the structure of a complex domain. They also provide a convenient vehicle to adopt the Bayesian philosophy, because hierarchical Bayesian models can be naturally specified as directed graphical models. For example, putting a prior on the model parameter θ, now treated as a random variable, is equivalent to adding a parent node that denotes the hyperparameter and associating the newly introduced edge with a prior distribution (Fig. 1.4b). A distinctive feature of the graphical model approach is its naturalness in formulating large probabilistic models of complex phenomena, by facilitating modular combination of heterogeneous submodels, using the property of the product rule of the joint distribution. Thus, a complex model can be assembled in a piecewise fashion, and even solved via a divide-and-conquer approach, as will be done in this thesis.

The field of computational genomics is fertile ground for the application of graphical models, and many of its complex problems can be readily handled within this formalism in a canonical and systematic way [Lauritzen and Sheehan, 2002]. For example, in a typical statistical genetics setting, we may want to model some complex genetic patterns with both observed and hidden variables using a likelihood model, and we concern ourselves with a sample set of N individuals (Fig. 1.5, bottom level). If we imagine that the genetic pattern of each individual is stochastically sampled from K possible populational genetic patterns, or in other words, they form K clusters, then we can make this explicit by adding the plate and nodes denoting K cluster centroids and the associated variances (Fig. 1.5, middle level). However, usually we do not know the number of clusters and where the centroids lie; in that case we can use a non-parametric Bayesian prior model to introduce a distribution over the space of all possible centroid sets (Fig. 1.5, top level). By this modular construction, we end up with a graphical model that corresponds to an infinite mixture model, as depicted in Fig 1.5. As you will see shortly, this graphical model is actually the formal foundation of a haplotype inference model we will develop in this thesis.

In summary, the graphical model framework provides a clean mathematical formalism that has made it possible to understand the relationships among a wide variety of network-based approaches

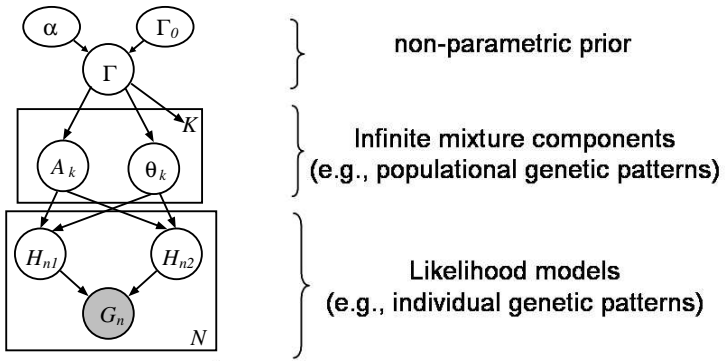


Figure 1.5: A graphical model representation of an infinite mixture model for complex populational genetic patterns.

to statistical computation, and in particular to understand many domain-specific statistical inference algorithms and architectures as instances of a broad probabilistic methodology. These features of graphical models help to greatly simplify the design of complex probabilistic models needed for our problems, and hopefully also make them easier to understand.

1.2 Thesis Overview

1.2.1 The Problem

In silico motif detection is the task of identifying potential motif patterns from DNA sequences using a pattern recognition program. Most contemporary motif detection algorithms were originally motivated by promoter analysis of yeast or bacteria genomes, which in general have a simple motif structure and organization [Bailey and Elkan, 1995a; Lawrence and Reilly, 1990; Lawrence *et al.*, 1993; Liu *et al.*, 1995; Hughes *et al.*, 2000; Liu *et al.*, 2001]. Therefore, these algorithms usually employ a naive approach for motif modeling, which typically assumes that, locally, the probabilities of the nucleotides at different sites within a motif are independent of each other; and globally, instances of motifs are distributed uniformly and independently in the regulatory sequence. In most cases, such an approach does not incorporate any prior knowledge of motif structures and motif organizations, even though there is a wealth of valuable information regarding these properties present in the biological community. These deficiencies, although well recognized very

early on, did not become a practical performance bottleneck (due to the small size and modest complexity of the study sequences being considered) until the recent completion of several grand sequencing projects that involve much more complex multicellular higher eukaryotes, such as *Drosophila* and human [Venter *et al.*, 2001]. With the availability of genomic sequences of these complex organisms, contemporary research in functional genomics is moving toward understanding the mechanisms and coding schemes of gene regulation networks driving biological processes unique to complex organisms, such as embryogenesis, differentiation, etc., which bear great relevance to medical and pharmaceutical interests [Markstein *et al.*, 2002; Berman *et al.*, 2002; Michelson, 2002]. A hallmark of the gene regulatory sequences of higher eukaryotes is the remarkable sophistication of the control program they employ to direct combinatorially fine-tuned gene expression in a time- and space-specific manner [Davidson, 2001]. The presence of highly sophisticated deterministic and stochastic constraints on motif deployment and the diverse categorization of motif structures in the aforementioned control programs, and the enormous size of the regulatory sequences in which motifs must be found, render existing methods inadequate for uncovering motif signals from the complex genomic background. More powerful models and computational algorithms are needed to cope with such challenge.

For autosomal loci in the genome of diploid organisms, when only the *genotypes* of multiple SNPs for each individual are provided, the haplotype for those individuals with multiple heterozygous genotypes is inherently ambiguous [Clark, 1990; Hodge *et al.*, 1999]. The problem of inferring haplotypes from genotypes of SNPs is essential for the understanding of genetic variations within and among populations, with important applications to the genetic analysis of disease propensities and other complex traits [Clark, 2003]. The problem can be formulated as a mixture model, where the set of mixture components corresponds to the pool of haplotypes in the population [Excoffier and Slatkin, 1995; Niu *et al.*, 2002; Stephens *et al.*, 2001; Kimmel and Shamir, 2004]. The size of this pool is unknown; indeed, knowing the size of the pool would correspond to knowing something significant about the genome and its history. Extant methods have largely bypassed explicitly modeling the uncertainty of this important quantity. Speaking

under a broader context, this problem is closely related to the perennial problem of "how many clusters?" in the clustering literature, and is particularly salient in large data sets where the number of clusters needs to be relatively large and open-ended. Current approaches based on fixing the number of clusters and using the mixing proportions or an information-theoretic score to gauge the appropriate number are clearly not adequate.

For many bioinformatics problems, including the problems we address in this thesis, probabilistic models have an inherent appeal, because they provide an elegant and powerful methodology to formulate various types of important problems such as classification, clustering, prediction and reasoning under uncertainly, and can systematically handle issues such as missing values, noisy data, prior knowledge, data fusion, etc. [Lauritzen and Sheehan, 2002; Jordan, 2004]. However, large-scale probability models, as are often needed in bioinformatics problems, have outgrown the ability of current (and probably future) exact inference algorithms to compute posteriors and learn parameters. This is particularly true for the models developed in this thesis, which involve high-dimensional Bayesian missing data problems. Although Monte Carlo algorithms [Gilks *et al.*, 1996] enjoy asymptotic correctness, and are often easy to implement, their prohibitive computational cost renders them practically infeasible for some of the challenging problems, as we encountered in motif detection. Some extant deterministic approximate inference algorithms, such as loopy belief propagation [Pearl, 1988; Murphy *et al.*, 1999], provide an alternative solution, but their generality and quality remain an open problem, which hinders their widespread application.

1.2.2 Contributions of This Thesis

In this thesis, we present a modularly designed hierarchical Bayesian Markovian model for motif detection in complex genomic sequences. This model, referred to as **LOGOS**, captures the dependency structure of regulatory elements at two levels: the conservation dependencies between sites within motifs, and the clustering of motifs into regulatory modules. In order to uncover unknown motifs *de novo* from higher eukaryotic genomes based solely on un-curated sequence data (a realistic scenario we have to face in animal genome annotation), **LOGOS** employs a mixture of

profile motif models, which can be trained on biologically identified motifs categorized according to protein-binding mechanisms and which can serve as a structured Bayesian prior for a probabilistic motif representation. Such a model biases the likelihoods of nucleotide strings toward those corresponding to biologically meaningful motifs rather than trivial patterns recurring in the genomic sequence, but does so without *a priori* committing to any specific consensus sequences. To our knowledge, this is the first model that enables *de novo* motif detection to benefit from prior knowledge of biologically identified motifs, and classifies motifs based on protein binding mechanisms. To model the locational organization of motifs in the genome, **LOGOS** also uses a hidden Markov model (HMM) to encode the syntactic rules of motif dependencies, with model parameters smoothed under empirical Bayesian priors. Using the graphical model formalism, the aforementioned model ingredients addressing different aspects of motif properties can be integrated into a composite joint probabilistic model. The modular architecture of **LOGOS** manifests a principled framework for developing, extending and computing expressive biopolymer sequence models.

The second result is an extension of the finite mixture models to the more flexible paradigm of countably-infinite mixture models. We present a nonparametric Bayesian model using the Dirichlet process prior, in the context of SNP haplotype inference for multiple SNPs. The model, which is referred to as *DP-haplotyper*, defines a prior distribution over both the centroids and the cardinality of a mixture model, that is, the identities and the numbers of the possible haplotypes in a population (rather than setting the number of haplotypes to an *ad hoc* fixed constant in extant models). It also employs a flexible likelihood model for each haplotype (i.e., each mixture component) to model the relationship between the haplotypes and the genotypes. As a result, DP-haplotyper accommodates growing data collections as well as noisy and/or incomplete observations during experimental genotyping, and imposes an implicit bias toward a small variety of haplotypes (i.e., a small number of centroids in the mixture model terminology) which is reminiscent of parsimony methods. This model outperforms the state-of-the-art haplotyping program, and is very promising as a building block for expressive models necessary in more complex problems related to SNP analysis.

Finally, the thesis presents a generalized mean field (GMF) theory for variational inference in

exponential family graphical models (to be defined in the sequel). A GMF method uses a family of tractable distributions defined on arbitrary disjoint model decompositions to approximate an intractable distribution, and solves the optimal approximation using a generic message passage procedure provably convergent to globally consistent fixed points of marginals and leading to a lower bound on the likelihood of observed data under the distribution. This framework generalizes several previous studies on model-specific structured variational approximation, yet specializes a previous study suggesting non-disjoint model decompositions, and appears to strike the right balance between quality of approximation and computational complexity. This algorithm has been used as the main inference engine for motif detection using the **LOGOS** model. The thesis also shows that the task of model decomposition, which is a prerequisite for the GMF algorithm, can be automated and optimized using graph partitioning; it demonstrates the empirical superiority of a minimal cut over other partition schemes, as well as giving theoretical justifications. This combination of GMF inference with combinatorial optimization represents an initial foray into the development of a truly turnkey algorithm for distributed approximate inference with bounded performance.

1.2.3 Importance for Bioinformatics, Computer Science and Statistics

The immediate use of these models and algorithms is in allowing us to develop software for solving certain long-standing computational genomics problems, specifically, motif detection and haplotype inference, under realistic and complex biological contexts, with noisy and incomplete measurements, and in light of empirical prior knowledge as well as theoretical insight from biological literature.

Biological systems are intrinsically complex and stochastic. In recognition of this, we have strived to develop large-scale mathematical models using principles of probability theory, graph theory and information theory to capture and appropriately handle these issues. It is our belief that the lack of mathematical sophistication in many extant bioinformatics models and programs is a concession to computational complexity, rather than a reflection of the biological reality of the systems or mechanisms under study. As a step toward dealing with these realities, this thesis also

concentrates on exploring computational techniques that can reliably and efficiently solve challenging large-scale probabilistic models.

Throughout the thesis, the formalism of probabilistic graphical models has been used to construct problem-specific Bayesian models, and guide the implementation of computational algorithms for inference and learning in solving the associated computational biology problem. The longer term value of this thesis and the most important idea from it, we would hope, is that, in certain problem domains, one can use probabilistic graphical models from beginning to end as a general-purpose modeling language to systematically, modularly, and formally build large-scale models for a complex domain in a *divide-and-conquer* and bottom-up fashion, avoiding being entangled in the immensely complex and often messy details one has to face in these domains; and to exploit the availability of general-purpose inference and learning algorithms for graphical models. As you proceed, the creation of the **LOGOS** model from the *MotifPrototyper* and *CisModuler* models, and the elaboration of *Pedi-haplotyper* from the basic *DP-haplotyper* hopefully serve as motivating examples.

We would particularly like to point out that, when pursuing probabilistic (in particular, Bayesian) approaches to complicated statistical problems, such as those in the biological domain, it is helpful, conceptually, to distinguish two separate issues [Stephens and Donnelly, 2003]:

- The **model** (e.g., prior distribution or likelihood function) for the quantities of interest. Examples (detailed shortly in the technical section) include, special prior models for the *positional weight matrices* of motifs, or for the *ancestral haplotype templates* of individual haplotypes. For a given data set, different model assumptions will in general lead to different posterior distributions and hence to different estimates.

- The **computational algorithm** used. For challenging problems, including the ones addressed in this thesis, the posterior distribution cannot be calculated exactly. Instead, computational methods — such as a variational inference algorithm, or Monte Carlo algorithms — are used to approximate it. Different computational tricks, or different settings of the "free knobs" in

the algorithms (e.g., number of iterations, convergence test, etc.), will change the quality of the approximation to the true posterior.

Not separating these two aspects in the face of a complex problem can be counter-productive. For example, it is not unusual to see summary sentences or listings like "we compare our algorithm *TIGER* with the extant algorithms *CAT*, EM, the Gibbs sampler, and the hidden Markov model ...", which is technically confusing and misleading, and strictly speaking, formally inappropriate. It obscures the technical ingredients of each algorithm, and conceals possible distinctions (or very often, lack of technical distinctions) between different algorithms—be it a model distinction, an algorithmic distinction for computation, or a distinction in the implementation. For instance, algorithm "TIGER" may also employ a Gibbs sampling algorithm for computation, and the "EM" and "Gibbs sampler" may have adopted the same probabilistic model. This blurring can cause unnecessary confusion when analyzing different models and possible duplication of previous work, and makes it difficult for practitioners or end-users to pick the appropriate algorithm for a certain task, and for developers to identify technical aspects subject to improvement. In this thesis, we intentionally make explicit these two aspects of computational probabilistic methodology in the exposition of existing and new models and algorithms.

The main theme of this thesis is the application of statistical machine learning approaches to computational biology. However, computational biology is not about simple matching between textbook algorithms and biological datasets. Close interactions between well-designed biological experiments and elegant yet realistic formulation of the mathematical models, as well as the development of efficient algorithms, are all essential to computational biology research. This thesis attempts to reflect the intimate interactions between biological concepts, mathematical formalisms, and computational algorithms, via an exposition that starts from highly problem-specific modeling efforts, followed by generalizations and combinations thereof, and eventually motivates an attempt to develop a generic computation technique. We believe that progress in the fields of machine learning and in biological research can be synergistic. Insights gained from theoretical and algorithmic research in machine learning can bring a new perspective and tools for studying biological objects,

and can foster new applications. On the other hand, biological research, facing systems of immense complexity and stochasticity rarely encountered elsewhere, challenges advanced mathematical and computational techniques for analysis and interpretation, and could lead to new developments that find broader application in fields outside biology that involve predictive understanding, learning and reasoning under uncertainty.

1.3 Technical Results of This Thesis

1.3.1 A Modular Parametric Bayesian Model for Transcriptional Regulatory Sequences

Most conventional motif models lack a clean formalism for imposing useful controls over where to search for motifs (hence, all regions are taken as equally likely to harbor motifs) and what substring patterns are preferred over others as candidate motifs (therefore, all recurring substring patterns are equally likely to be accepted as functionally meaningful motifs). In Chapter 2, we propose a principled framework for introducing such controls for motif modeling. The goal is to develop a formalism that is expressive (in terms of being able to capture the internal structures, organizational rules, and other properties of motifs, and readily incorporating prior knowledge about these properties from biological literature), yet mathematically and algorithmically transparent and well-structured, hence simplifying model construction, computation and extension. Based on the product rule of the joint probability in the graphical model formalism, we outline the formal architecture of a modular motif model with the following three components: the *local alignment model*, which captures the intrinsic properties within motifs, including characteristic position weight matrices (PWMs) and site dependencies; the *global distribution model*, which models the frequencies of different motifs and the dependencies between motif occurrences in a sequence; and the *background model*, which defines the distribution of non-motif nucleotide sequences. The model components can be designed separately, and then fused into a consistent, more expressive joint model.

1.3.1.1 Profile Bayesian models for motif sequence pattern

It is well known that the DNA-binding domains of gene-regulatory proteins fall into several distinctive classes, such as the zinc-finger class or the helix-turn-helix class. This classification strongly suggests that different motif patterns with different consensus sequences may share some local structural regularities intrinsic to a family of different motifs corresponding to a specific class of DNA-binding proteins.

In Section §2.4, we address the problem of modeling generic features of *structurally* but not *textually* related DNA motifs, that is, motifs whose consensus sequences are entirely different, but nevertheless share "meta-sequence features" reflecting similarities in the DNA binding domains of their associated protein recognizers. We present MotifPrototyper, a profile hidden Markov Dirichlet-multinomial (HMDM) model that is able to capture regularities of the *nucleotide-distribution prototypes* and the *site-conservation couplings* typical to a particular family of motifs that correspond to regulatory proteins with similar types of structural signatures in their DNA binding domains. Central to this framework is the idea of formulating a profile motif model as a family-specific structured Bayesian prior model for the PWMs of motifs belonging to the family being modeled, thereby relating these motif patterns at the *meta-sequence level*.

The HMDM model assumes that positional dependencies within a motif are induced at a higher level among a finite number of informative Dirichlet priors, rather than directly between the position-specific distributions (which are generally set to be multinomials) of the nucleotides of the sites inside a motif. Under this framework, one can explicitly capture meta-sequence features, such as different conservation patterns of nucleotide distribution (e.g., being *homogeneous* or *heterogeneous*), and the 1st-order Markov dependencies of such patterns between adjacent sites. In general, the HMDM model can be used to formally encode prior knowledge about the intrinsic structure of a family of different motifs sharing meta-sequence features, by learning the parameters of the model from experimentally identified motifs of the family. This can be done by using a stochastic EM algorithm to compute the empirical Bayes estimate of the parameters. The result is a family-specific Bayesian profile model that implicitly encodes meta-sequence features shared in this family.

We then show how the family-specific profile HMDMs, or MotifPrototypers, can be used to classify aligned multiple instances of motifs into different classes each corresponding to a certain class of DNA-binding proteins; and most importantly, how a mixture model built on top of multiple profile models can facilitate a Bayesian estimation of the PWM of a novel motif. The Bayesian estimation approach connects biologically identified motifs in the database to previously unknown motifs in a statistically consistent way (which is not possible under the single-motif-based representations described previously) and turns *de novo* motif detection, a task conventionally cast as an *unsupervised* learning problem, into a *semi-unsupervised* learning problem that makes substantial use of existing biological knowledge.

A recent paper by Barash *et al.* proposes several expressive Bayesian network representations (e.g., tree network, mixture of trees, etc.) for motifs, which are also intended for modeling dependencies between motif sites [Barash *et al.*, 2003]. An important difference between these two approaches is that, in Barash's Bayesian network representations, the site-dependencies are modeled directly at the level of site-specific nucleotide distributions in a "sequence-context dependent" way; whereas in the HMDM model, the site-dependencies are modeled at the level of the **prior distributions** of the site-specific nucleotide-distributions in a "conservation-context dependent" way. Thus, Barash's motif models have one-to-one correspondence with particular motif consensus patterns, and need to be trained on an one-model-per-motif basis. On the other hand, the HMDM model corresponds to a generic signature structure at the meta-sequence level; it is not meant to commit to any specific consensus motif sequence, but aims at generalizing across different motifs bearing similar conservation structures. In terms of the resulting computational task in *de novo* motif detection, Barash's model needs to be estimated in an *unsupervised* fashion and makes no use of the biologically identified motifs in the database, whereas the HMDM model helps to turn the model estimation task into a *semi-unsupervised* learning problem that draws a connection between novel motifs to be found and the biologically identified motifs via a shared Bayesian prior, so that the patterns to be found are biased toward biologically more plausible motifs. It is interesting to note that

these two approaches are complementary in that Barash's models provide a more expressive likelihood model of the motif instances, and the HMDM model can be straightforwardly generalized to define a prior distribution for these more expressive models (e.g., replacing the Markov chain for the prototype sequence in the HMDM model with a tree model and/or introducing Dirichlet mixture priors for the parameters of Barash's models).

1.3.1.2 Bayesian HMM for motif organization

In complex multi-cellular organisms such as higher eukaryotes, the distribution of motif strings in the genome often follows a general principle called modular organization. That is, the motifs that are involved in regulating the expression of a given gene are not distributed uniformly and at random in the regulatory region of the gene. Instead, they are organized into a series of discrete sequence regions called *cis*-regulatory modules, each of which controls a distinct aspect of the gene. Within each module certain combinations of motifs occur with increased frequency; these motifs are capable of integrating, amplifying, or attenuating multiple regulatory signals via combinatorial interaction with multiple regulatory proteins. This architecture is somewhat analogous to the grammatical rules we use to synthesize natural language from words. A motif detection algorithm that ignores these syntactic rules often fails to correctly score true signals in a motif-dense region but on the other hand is sensitive to false positives in the background region.

Taking an approach that has been widely adopted in many language and sequence segmentation problems, we assume that underlying each sequence of nucleotides is a 1st-order hidden Markov model, whose realizable state sequences correspond to segmentations of the DNA sequence. For states corresponding to motif sites, the PWM of the corresponding motif is used to define the emission probabilities of observed nucleotides. For a non-motif state, it is assumed that probability of the corresponding nucleotide is kth-order Markovian. What is unique about this specialized HMM model, which we refer to as CisModuler, is the design of the state space of the hidden variables, which corresponds to a rich set of possible functional annotations of each position in the transcriptional regulatory sequences; and the state-transition scheme, which encodes the stochastic syntactic

rules of the CRM organizations of motifs known from the literature. Also somewhat novel is that this model is trained in a semi-unsupervised fashion, from unlabeled sequences under a Bayesian prior centered around empirical guesses of state transition probabilities. Thus, soft controls over the distances between motif instances and motif modules, and over their dependencies, can be imposed based on empirical knowledge from some reasonable sources (e.g., domain experts, literature, etc.), and, due to the Bayesian approach, are subject to dominance by (rather than over) the evidence when the study data is abundant.

1.3.1.3 The LOGOS model

A combination of the MotifPrototyper and CisModuler models, using the product rule of joint probability in a graphical model, leads to a novel Bayesian model that is significantly more expressive than any extant motif detection model. It is referred to as **LOGOS**, for integrated **LO**cal and **G**l**O**bal motif **S**equence model.[2] In **LOGOS**, the functional annotations of a DNA sequence that determine the motif locations and modular structures are determined by a CisModuler HMM model; but the emission probabilities of the motif states, or the PWMs of the motifs, are assumed to be generated from the MotifPrototyper model or a mixture of MotifPrototypers, whereby prior knowledge regarding both global motif organization and local motif structure is incorporated. As in other recent motif models, the background model used by **LOGOS** is a local 3rd-order Markov model. Under the trained prior models, **LOGOS** performs *de novo* motif detection in a semi-unsupervised fashion.

Note that **LOGOS** defines a very general framework for modeling gene regulatory sequences, using a modular graphical model. Each module can be designed separately to model different aspects of the motif properties and can be updated without overhauling the whole model. The Bayesian missing data problem associated with **LOGOS** is a challenging computational problem that cannot be handled by extant exact inference algorithms. Nevertheless, the modular structure of the **LOGOS** model motives a divide-and-conquer approach for approximate inference using the

[2] Not to be confused with '*logo*,' a graphic representation of an aligned set of biopolymer sequences first introduced by Tom Schneider [Schneider and Stephens, 1990] to help in visualizing the consensus and the entropy (or "information") patterns of monomer frequencies. A *logo* is not a motif finding algorithm, but is often used as a way to present motifs visually.

GMF algorithm (described shortly). GMF essentially couples *local* exact inference computations for each submodel of **LOGOS** using an iterative procedure, and leads to a variational approximation to the Bayesian estimation. The theoretical and algorithmic issues of variational inference in general and of GMF in particular are addressed in Chapter 4.

Thanks to the flexibility of assembling a full motif model with different combinations of submodels under the **LOGOS** framework, several variants of the **LOGOS** model that differ in model expressiveness (e.g., MotifPrototyper + CisModuler, PWMs + uniform global model, etc.) are constructed to examine the performance gain (or loss) due to different model components. There is strong evidence that improvements introduced in this thesis on both the local aspect (i.e., MotifPrototyper over the independent PWMs) and the global aspect (i.e., CisModuler over the uniform model) of the motif model improve performance. Due to the lack of a sufficient number of well annotated human regulatory sequences for model evaluation, validations are primarily conducted on yeast and *Drosophila* DNA sequences. It is evident that on both the regulatory sequences of yeast and those of *Drosophila*—whose sizes and complexity are comparable to that of human—the **LOGOS** model outperforms the popular MEME and AlignACE algorithms.

1.3.2 A Non-Parametric Bayesian Model for Single Nucleotide Polymorphisms

The problem of inferring haplotypes from genotypes of single nucleotide polymorphisms can be formulated as a mixture model, where the mixture components correspond to the haplotypes in the population. The size of the pool of haplotypes is unknown, and biologically, a parsimonious bias toward a more compact haplotype reconstruction (i.e., a pool with smaller number of distinct population haplotypes sufficient for explaining the genotypes) is desired. Thus methods for fitting the genotype mixture must crucially address the problem of estimating a mixture with an unknown number of mixture components and the parsimony bias. Chapter 3 presents a Bayesian approach to this problem based on a nonparametric prior known as the *Dirichlet process* [Ferguson, 1973], which attempts to provide more explicit control over the number of inferred haplotypes than has been provided by the statistical methods proposed thus far. The resulting inference algorithm has

commonalities with parsimony-based schemes.

In the setting of finite mixture models, the Dirichlet process—not to be confused with the Dirichlet distribution—is able to capture uncertainty about the number of mixture components [Escobar and West, 2002]. The basic setup can be explained in terms of an urn model, and a process that proceeds through data sequentially. Consider an urn which at the outset contains a ball of a single color. At each step we either draw a ball from the urn, and replace it with two balls of the same color, or we are given a ball of a new color which we place in the urn, with a parameter defining the probabilities of these two possibilities. The association of data points to colors defines a "clustering" of the data. As pointed out by Tavare and Ewens [1998], this process is not only a mathematically convenient model to deal with uncertainty of the cardinality of a mixture model, but it indeed corresponds to an interesting metaphor of "biological evolution without selection."

To make the link with Bayesian mixture models, we associate with each color a draw from the distribution defining the parameters of the mixture components. This process defines a *prior distribution* for a mixture model with a random number of components. Multiplying this prior by a likelihood yields a *posterior distribution*. In Chapter 5, Markov chain Monte Carlo algorithms are developed to sample from the posterior distributions associated with Dirichlet process priors.

The usefulness of this framework for the haplotype problem should be clear—using a Dirichlet process prior we in essence maintain a pool of haplotype candidates that grows as observed genotypes are processed. The growth is controlled via a parameter in the prior distribution that corresponds to the choice of a new color in the urn model, and via the likelihood, which assesses the match of the new genotype to the available haplotypes. This latter point also manifests an advantage of the probabilistic formalism in that it is straightforward to elaborate the observation model for the genotypes to include the possibility of errors. Trading off these errors against the size of the pool of haplotypes can be gauged in a natural and statistically consistent way. Overall, the Dirichlet process mixture naturally imposes an implicit bias toward small ancestral pools during inference (reminiscent of parsimony methods), and does so in a well-founded statistical framework that permits errors. We call the this non-parametric Bayesian model *DP-haplotyper*.

The state-of-the-art algorithm for haplotype inference is the algorithm known as "PHASE." The performance of DP-haplotyper is equivalent to PHASE on the easier phasing problems that we study, and improves on PHASE for the hardest problem; also DP-haplotyper requires less computation time. It also provides an upgrade path to models that permit recombination and incorporate pedigrees as we outline in section §3.3, and can potentially generalize to linkage analysis and other population genetics problems. Thus, DP-haplotyper serves as a promising building block for more expressive models necessary for more complex problems.

1.3.3 The Generalized Mean Field Algorithms for Variational Inference

A critical limitation of using sophisticated probabilistic models for complex problems has been the time and space complexity of the inference and learning algorithms. For example, to predict motif locations and estimate motif PWMs under the **LOGOS** model, one has to manipulate (e.g., marginalize) a posterior distribution over the Cartesian product of a continuous state space and a discrete one, both of very high dimension. Such computations are prohibitively expensive for any exact algorithms. Although applying Monte Carlo algorithms is possible, efficiency and performance concerns motivated us to pursue deterministic approximation methods based on a variational calculus technique.

In Chapter 4, we present a class of generalized mean field algorithms for approximate inference in exponential family graphical models. GMF is analogous to cluster variational methods such as generalized belief propagation (GBP). While those (GBP) methods are based on overlapping clusters of variables in the model to define local marginals to be approximated and messages to be exchanged among local marginals, GMF is based on nonoverlapping variable clusters. Unlike the cluster variational methods, GMF is proved to converge to a globally consistent set of cluster marginals and a lower bound on the likelihood, while providing much of the flexibility associated with cluster variational methods.

Given an arbitrary decomposition of the original model into disjoint clusters, the GMF algorithm computes the posterior marginal for each cluster given its own evidence and the *expected*

sufficient statistics, obtained from its neighboring clusters, of the variables in the cluster's Markov blanket (to be defined in the sequel) — thence referred to as the Markov blanket messages. The algorithm operates in an iterative, message-passing style until a fixed point is reached. We show that under very general conditions on the nature of the inter-cluster dependencies, the cluster marginals retain exactly the intra-cluster dependencies of the original model, which means that the inference problem within each cluster can be solved independently of the other clusters (given the Markov blanket messages) by any inference method.

One way to understand the algorithm is to consider a situation in which all the Markov blanket variables of each cluster are observed. In that case, the joint posterior decomposes:

$$p(\mathbf{x}_{C_1}, \ldots, \mathbf{x}_{C_n} | \mathbf{x}_E) = \prod_i p(\mathbf{x}_{C_i} | \mathcal{MB}(\mathbf{x}_{C_i}), \mathbf{x}_{E_i, C_i}),$$

where $\mathcal{MB}(\mathbf{x}_{C_i})$ denotes the Markov blanket of cluster C_i, and $\mathbf{x}_{E_i,C_i}$ denotes the evidence node within cluster i. GMF approximates this situation, using the expected Markov blanket (obtained from neighboring clusters) instead of an observed Markov blanket and iterating this process to obtain the best possible "self-consistent" approximation.

In its use of expectations in messages between clusters, GMF resembles the expectation propagation (EP) algorithm [Minka, 2001], but in the basic EP algorithm the messages convey the influence of only a single variable. In providing a generic variational algorithm that can be applied to a broad range of models with convergence guarantees, GMF resembles VIBES [Bishop *et al.*, 2003], whose original version was based on a decomposition into individual variables, and later generalized to allow more coarse-grained disjoint decompositions similar to what we used for GMF [Bishop and Winn, 2003]. Thus GMF is a generic algorithm suitable for approximate inference in large, complex probability models.

Disjoint clusters have another virtue as well, which is explored in the second half of Chapter 4 — they open the door to a role for graph partitioning algorithms in choosing clusters for inference. We provide a preliminary formal analysis and a thoroughgoing empirical exploration on how to choose a good partition of the graph automatically using graph partitioning algorithms, so that the

entire GMF inference algorithm can be implemented in a fully autonomous way, with little or no human intervention. We present a theorem that relates the weight of the graph cut to the quality of the bound of GMF approximation, and study random graphs and a variety of settings of parameter values. We compare several different kinds of partitioning algorithms empirically and the results turn out to provide rather clear support for a clustering algorithm based on minimal cut, which is consistent with implications drawn from the formal analysis.

The combination of GMF inference with graph partitioning based on combinatorial optimization make it possible to develop truly turnkey algorithms for distributed approximate inference with bounded performance.

1.4 Thesis Organization

The thesis stands at the intersection of several areas, namely, computer science, statistics, molecular biology and genetics, and draws heavily on statistical machine learning, Bayesian statistics, optimization theory, graph theory, and various biology-related sub-areas. Nonetheless, the reader is not assumed to have a thorough background in any of these areas, but a general knowledge of the basic concepts and techniques (e.g., discrete and continuous probability, EM algorithms, etc.) and I have made some effort to make the thesis readable to a general audience in machine learning, statistics, and computational biology.

Chapters 2-5 present the main contributions in this thesis. Chapters 2, 3, and 4 are self-contained and can be read separately from the rest, whereas Chapter 5 should be read in the context of Chapters 2 and 3. Chapter 2 describes a modular parametric Bayesian model for motif detection in complex genomes. Chapter 3 presents a non-parametric Bayesian model for inferring the haplotypes of SNPs in a population. Chapter 4 presents a generalized mean field theory and algorithm for variational inference in exponential family graphical models (to be defined in the sequel) and its application to motif detection using the models developed in Chapter 2. Chapter 5 provides Monte Carlo algorithms for inferring motifs and haplotypes based on models in Chapters 2 and 3.

Those readers most interested in novel motif detection techniques as well as a detailed overview

of extant methods are advised to read Chapter 2 first and then chapter 4 and section 2 of Chapter 5. Those interested in new models for haplotype inference should start with Chapter 3 and continue to sections 3-5 of Chapter 5. Those interested in approximate inference theory and algorithms are advised to read Chapter 4, and then Chapter 2 as an instance of large-scale application.

Chapter 6 summarizes the results of this thesis, draws a few conclusions and presents a set of open questions and directions for further investigation.

Some of the material in this thesis has appeared before in [Xing *et al.*, 2003a; Xing *et al.*, 2003b; Xing *et al.*, 2004a; Xing *et al.*, 2004c; Xing *et al.*, 2004b; Xing and Karp, 2004].

Chapter 2

Modeling Transcriptional Regulatory Sequences for Motif Detection

— A Parametric Bayesian Approach

Motifs are short recurring string patterns scattered in biopolymer sequences such as DNA and proteins. The characteristic sequence patterns of motifs and their locations often relate to important biological functions, such as serving as the *cis*-elements for gene regulation or as the catalytic sites for protein activity. The identification of motif sites within biopolymer sequences is an important task in molecular biology and is essential in advancing our knowledge about biological systems.

It is well known that only a small fraction of the genomic sequences in multi-cellular higher organisms constitute the protein coding information of the genes (e.g., only 1.5% for human genomes [Alberts *et al.*, 2002]), whereas the rest of the genome, besides playing purely structural roles such as forming the centromeres and telomeres of the chromosomes, contains a large number of short DNA motifs that make up the immensely rich codebook of the gene regulation program, known as the *cis-regulatory system* [Blackwood and Kadonaga, 1998; Davidson, 2001]. It is believed that this regulatory program determines the level, location and chronology of gene expression, which significantly, if not predominantly, contributes to the developmental, morphological and behavioral diversity of complex organisms [Davidson, 2001].

For proteins, functional specificities are usually realized by the presence of sporadic, but structurally pivotal and/or biochemically reactive *activity sites* in the amino acid sequences [Lockless

and Ranganathan, 1999; Li *et al.*, 2003]. Therefore, proteins with very different overall sequences and structures can fall into common functional categories, such as *kinase* and *methylase*, and bear common polypeptide motifs (which constitute the activity sites) embedded in diverse sequence and structural environments. Polypeptide motifs are regarded as signatures of unique biophysical and biochemical functions. Due to the functional importance of polypeptide motifs to their host proteins, they usually represent regions in protein sequences that resist drift and are prone to stabilizing selection [Page and Holmes, 1998]. Thus, protein motifs provide important clues to understanding the function and evolution of proteins and organisms.

Motifs can be identified via "wet lab" biological experiments, such as DNAase protection assay (for DNA motifs) [Ludwig *et al.*, 2000] and site-specific mutagenesis (for protein motifs) [Haldimann *et al.*, 1996], which are often very labor-intensive and time-consuming, but arguably most reliable in the biological sense (although in some cases the truthfulness of *in vitro* assays or mutational perturbation results with respect to biological reality is debatable). The best collections of experimentally identified and verified motifs can be found in the TRANSFAC and the PROSITE databases [Wingender *et al.*, 2000; Sigrist *et al.*, 2002]. But since experimental motif identification is often very expensive and tedious, with the rapid accumulation of genomic sequence information from more and more species, advances in molecular biology call for the development of more cost-effective, computation-based methods for motif detection directly from the sequence data. In this chapter, we review previous advances and extant methods in this direction and present a new Bayesian approach we developed. Some of the material in this chapter has appeared before in [Xing *et al.*, 2003a; Xing *et al.*, 2004b; Xing and Karp, 2004]. To simplify our exposition, we use DNA motif detection as a running example, but it should be clear that the models we present are readily applicable to protein motifs.

2.1 Biological Foundations and Motivations

Transcription, the process of making a single-stranded RNA molecule using one of the two DNA strands of a gene sequence as a template, is exquisitely but robustly controlled by the interactions

between the transcription factors that bind the *cis*-regulatory elements in DNA, the basal transcriptional apparatus, additional co-factors, plus the influences from the chromatin structures (Fig. 1.1). An initial step in the analysis of the function and behavior of any gene is the identification of genomic regions that might harbor the *cis*-regulatory elements, and the elucidation of the identities and organization of these elements.

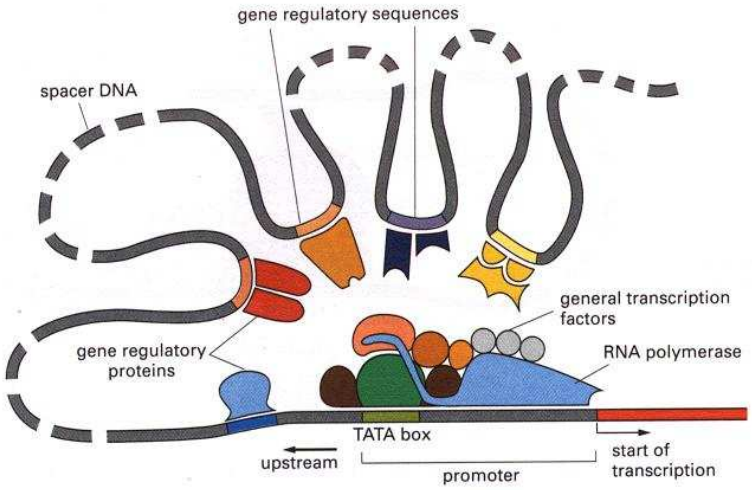


Figure 2.1: Motif recognition and transcriptional regulation.

DNA motifs can be recognized by specific regulatory proteins, which relay complex regulatory signals to the basal transcriptional machinery made up of an RNA polymerase and general transcriptional factors via physical interactions, and accordingly turn on/off or fine-tune the expression of a gene [Ptashne and Gann, 1997] (Fig. 2.1). The specific motif-protein recognition underlying the physical foundation of transcription regulation suggests that there exists a unique structural complementarity between each motif sequence and the corresponding protein recognizer [Stormo and Fields, 1998; Stormo, 2000; Benos *et al.*, 2002]. For a simple organism such as a bacterium, the *cis*-regulatory systems usually contain a small number of motifs located closely proximal to the transcription initiation sites of the genes [Alberts *et al.*, 2002]. On the other hand, in complex multi-cellular organisms such as higher eukaryotes, the distribution of motif sites in the genomic sequences often follows a general principle called *modular organization* [Davidson, 2001]. The top panel of Fig. 2.2 shows a diagram of the regulatory region of the *Drosophila even-skipped*

(eve) gene. This gene is involved in establishing the body segmentation during *Drosophila* embryogenesis by expressing itself in different parts of the early embryo, know as the *stripes* (middle panel, Fig. 2.2), at different times, to determine the developmental fate of the corresponding stripes [Harding *et al.*, 1989; Goto *et al.*, 1989; Stanojevic *et al.*, 1991; Small *et al.*, 1996; Sackerson *et al.*, 1999; Fujioka *et al.*, 1999]. For example, the first two stripes shown in Fig. 2.2 will grow into the head of the animal, and the third one will become a pair of legs [Gilbert, 2003; Alberts *et al.*, 2002]. As shown in this diagram, the motifs that are involved in regulating the expression of this gene are not distributed uniformly and at random in the regulatory region of the gene. Instead, they are organized into a series of discrete sequence regions called *cis-regulatory modules* (or CRMs), each of which controls a distinct aspect of the gene's expression pattern, namely, when, in which stripe, and in roughly how many copies it is to be transcribed [Davidson, 2001; Michelson, 2002]. This general architecture applies to most transcriptional regulatory sequences in complex organisms, but does not apply to simple uni-cellular biological systems [Davidson, 2001].

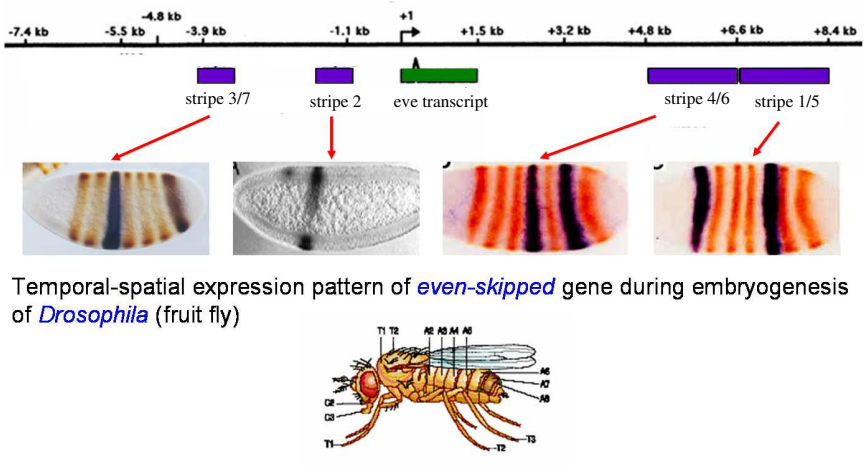


Figure 2.2: The *Drosophila* CRMs and their roles in early embryogenesis.

As mentioned, *in silico* motif detection is the task of identifying potential motif patterns from DNA sequences using a pattern recognition program. However, unlike another pattern recognition problem on DNA sequences — gene finding — which searches for "macroscopic" entities such

as genes (or more precisely, exons of the genes) in the genome, motif finding focuses on "microscopic" substring patterns embedded in a long stream of noisy background full of false positive signals. To the best of our knowledge, most of the biologically verified motifs are very short (i.e., about $6 \sim 30$ monomers), stochastic (i.e., different instances of the same motif usually differ slightly in sequence content, as detailed shortly in §2.2.1), and poorly structured (i.e., they contain no substructure bearing universal sequence signatures such as the intron-exon junctions for genes). Thus, a coarse-grained model, such as a generic HMM that captures a universal intron/exon boundary signature and the overall nucleotide frequencies of coding sequences, as used in the GENESCAN program [Burge and Karlin, 1997], is infeasible for detecting small and very diverse motif signals. More specific models for short sequence patterns, which correspond to regulatory proteins that bear unique functions, are necessary to represent and search for DNA motifs.

What distinguishes a motif sequence from other random patterns in the background? Besides the fact that a motif has a recurring consensus polynucleotide pattern, numerous studies of the biophysical mechanisms of DNA-protein binding underlying the *cis-trans* regulatory interactions reveal that a typical binding protein (e.g., a transcription factor with helix-turn-helix binding motifs or tandem zinc-fingers) only interacts with a DNA motif through a few highly specific amino acid-nucleotide interactions, but is tolerant of variations in other sites [Stryer, 1995; Eisen, 2003]. It is also well known that for higher eukaryotic organisms, motifs usually cluster into CRMs [Davidson, 2001]. Each CRM consists of a locally enriched battery of motifs occurring in a certain combination and ordering, capable of enhancing or integrating multiple regulatory signals via concurrent physical interaction with multiple TFs [Berman *et al.*, 2002]. The spatial organization of CRMs in the regulatory regions of the genes is also essential for coordinating gene activities. These features are not directly reflected in the composition or consensus of a sequence pattern, and therefore can be referred to as **meta-sequence features**.

The meta-sequence features of motif structure and motif organization, which are believed to be crucial in distinguishing biologically meaningful motifs from a random background or trivial recurring patterns, have raised significant challenges to conventional motif-finding algorithms, most

of which rely on models that describe motifs only at their sequence level and use simplifying independence assumptions that decouple potential associations among sites within each single motif and among multiple instances of motifs [Bailey and Elkan, 1995a; Bussemaker *et al.*, 2000; Hughes *et al.*, 2000; Liu *et al.*, 2001; Gupta and Liu, 2003]. Therefore, although there is much success for motif detection on short, well curated bacterial or yeast gene regulatory sequences using extant methods, generalization to longer, more complex and weakly characterized input sequences such as those from higher eukaryotic genomes seems less immediate [Papatsenko *et al.*, 2002; Rajewsky *et al.*, 2002]. A recent survey by Eisen [2003] raises concerns over the inability of some contemporary motif models to incorporate biological knowledge of global motif distribution, motif structure and motif sequence composition.

2.2 Problem Formulation

2.2.1 Motif Representation

To formulate the motif detection problem, we begin with a brief discussion on how to represent a motif pattern. The representations of motif patterns largely fall into two categories: deterministic representations, and stochastic representations.

For concreteness, Fig. 2.3 shows an example of a stretch of regulatory DNA sequence that contains instances of multiple motifs. All the sub-strings highlighted with the same color in this example correspond to the binding sites that can be recognized by the same TF. The simplest way to represent a motif pattern corresponding to a TF is to consider each motif as a "word" — a deterministic substring pattern. However, as shown in Fig. 2.3, the instances of a motif are merely "similar," but not identical to each other. Thus some flexibility is needed to accommodate discrepancies among instances of the same motif. Usually, biologists record a motif pattern using a *multiple alignment* of all the instances of a motif (Fig. 2.3a). An inspection of the alignment shown in Fig. 2.3a suggests that the word "TTTTTATG" may be a reasonable representation of this motif because it records the most frequent nucleotide at each column of the alignment (although nucleotides "T" and "A" draw a tie at the 6th column). A word derived from a multiple alignment

in this way is called a *consensus* of the motif [Stormo, 2000]. In using a consensus sequence to match additional instances of the motif it represents, some deviations, such as k mismatches with the consensus (where k is a small integer compared to the length of the consensus, say, m), are usually allowed between the instances and the consensus. A *regular expression* — in the foregoing case, "TTTTTXTG", where "X" means "don't care" — is another popular deterministic representation. It can be used to restrict the allowable mismatches to certain positions when matching for motif instances [Mehldau and Myers, 1993]. There have been several approaches for motif detection directly based on word enumeration [van Helden *et al.*, 1998; Sinha and Tompa, 2000; Bussemaker *et al.*, 2000], some of which will be reviewed in the sequel.

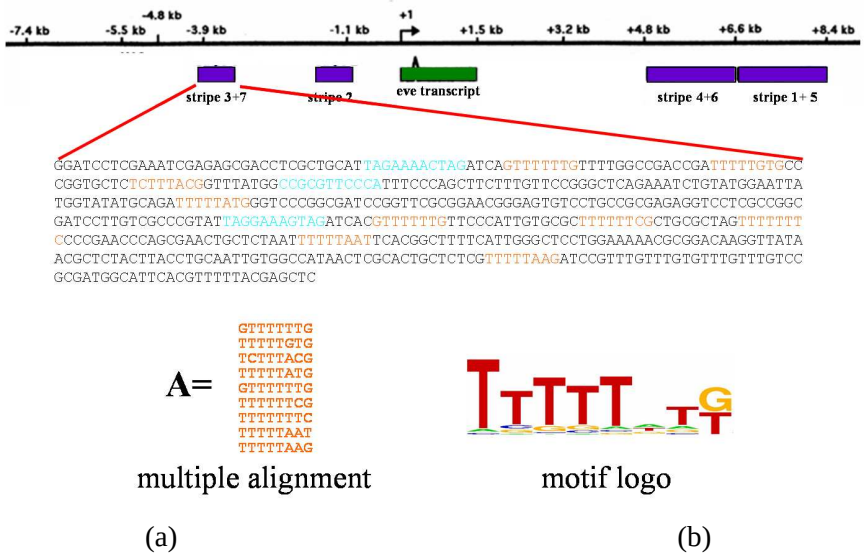


Figure 2.3: A close-up of motif instances and CRMs in a *Drosophila* sequence.

Alternative to the deterministic representations, due to the stochastic nature of motif patterns, it is also natural to consider motif instances as samples drawn from a stochastic representation, which usually corresponds to a generative probabilistic model. For example, a motif can be represented by a *position weight matrix* [Cardon and Stormo, 1992; Hertz and Stormo, 1996], which records the nucleotide frequencies at each column of the alignment. Pictorially, a motif pattern can also be represented by a *sequence logo* [Schneider and Stephens, 1990], in which the height

of each column corresponds to the degree of conservativeness, measured by the entropy of the nucleotide distribution at a column of the motif alignment; the height of each character in a column relates to the relative frequency of the respective nucleotides in the corresponding position in the motif (Fig. 2.3b). With a stochastic motif representation, one can rank the "strength" of matches of candidate motif instances to a given motif representation with a score that bears probabilistic or information-theoretic interpretation, such as likelihood or log odds [Stormo, 2000]. In this thesis, we focus on probabilistic representations of motifs and explore models and algorithms for learning and prediction under such representations.

2.2.2 Computational Tasks for *In Silico* Motif Detection

The term "motif detection", or "motif finding", has been heavily loaded in the literature, often with ambiguous meanings in terms of the exact nature of the intended computational task. To avoid possible confusion in the forthcoming exposition, in the following we make explicit three distinct, but related, computational tasks underlying a typical motif detection problem, and assign a technically unambiguous handle to each.

First, given a set of experimentally identified instances of a certain motif (i.e., all the DNA segments elucidated from a DNAase-protection assay for a specific DNA-binding regulatory protein), we call the task of extracting a motif representation, or a motif model, from such a set, **motif training**. In machine learning terminology, motif training can be understood as a *supervised learning* problem, and the aforementioned set of instances of the motif is called a *training set*. As elaborated in the sequel, depending on the choice of motif representation, different approaches can be used for motif training, which typically begins with a multiple alignment of all the instances in the training set, followed by specific procedures to learn the representations, such as a regular expression, a consensus, or a probabilistic model, from the resulting alignment. In particular, when a motif patten is represented as a probabilistic model (e.g., the local models in the sequel), motif training boils down to *parameter estimation* of the probabilistic model.

Second, given a model or a representation of a known motif, the task of searching for the

presence of the sites of this motif in an unannotated set of sequences via computational means is called **motif scan**. Frequently, generalization to simultaneous *multiple-motif scan* is needed. In many combinatorial motif detection algorithms, motif scan is typically formulated as a "k-m string matching" problem, that is, finding all substrings of length m (i.e., m-mers, where m is the length of the given motif pattern) that has at most k mismatches with the motif pattern. Accordingly, the motif patterns to be scanned for are represented by their respective consensus sequences. From a machine learning point of view, in the simplest case, motif scan can be formulated as a standard *classification* problem for all substrings in the sequences according to a deterministic or probabilistic motif model. Under a more sophisticated formulation, in which the contextual information and the dependencies among instances are to be considered, an explicit locational distribution model of motifs (e.g., the global model in the sequel) can be used, and motif scan can be cast as the problem of *probabilistic inference* for latent random variables in the model that indicate the locations of the motifs.

Finally, given only a set of unannotated sequences potentially containing previously uncharacterized motifs (i.e., motifs whose representations are not known), the task of learning the representations of these unknown motifs and at the same time locating all the instances of these motifs in the study sequences is referred to as ***de novo* motif detection**. Under a combinatorial setting, which typically adopts a deterministic representation of a motif pattern, *de novo* motif detection often amounts to finding all over-represented m-mers from the sequences, where m is the length of the anticipated motifs. Some parameters are needed to qualify a match (e.g., the k in the aforementioned k-m score) and to determine how many over-represented patterns are to be accepted (e.g., a cutoff value for the minimal number of matches) [Papatsenko *et al.*, 2002]. Under a probabilistic framework, which will be studied in detail in this thesis, one can view *de novo* motif detection as a coupled *missing value inference* and *parameter estimation* problem, often formulated as an *unsupervised learning* problem [Bailey and Elkan, 1995a].

As we will elaborate shortly, besides avoiding possible ambiguities about what one means by "motif detection", the foregoing clarification of the computational tasks underlying the motif detection problem also bodes well for the logic of a **modular formulation** of the *in silico* motif detection

problem, and a *divide-and-conquer* strategy to solve such problems. Just as a quick overview, it is not difficult to realize that the models (and the algorithms) for *motif training* and *motif scan*, respectively, can be viewed as submodels (and subroutines) of the more difficult *de novo* motif detection problem in that, computationally, *de novo* motif detection often amounts to an iterative procedure (modulo some technical issues regarding how to jump start the iteration, whether it will ever converge, etc.) that alternates between: 1) scanning for instances of a motif using a newly-trained motif model, and, 2) training an updated motif model using the newly-scanned set of motif instances. Hence, the full model and the algorithm for *de novo* motif detection is in essence a combination of the two models underlying motif scan and motif training, respectively. This modular logic indeed underlies the two main families of algorithms currently in use for *de novo* motif detection under various model settings, namely, expectation-maximization (EM) (e.g., [Lawrence and Reilly, 1990; Bailey and Elkan, 1995a]) and Monte Carlo (MC) (e.g. [Lawrence *et al.*, 1993; Liu *et al.*, 2001]) algorithms. In the sequel, we will adopt this logic to analyze several extant motif models and present new models and algorithms for motif detection.

2.2.3 General Setting and Notation

Now we introduce the necessary notation for the formal presentation. We denote a regulatory DNA sequence by a character string $y = (y_1, \ldots, y_T) \in \mathbb{N}^T$, where $\mathbb{N} = \{\text{A,T,C,G}\}$ denotes the set of all possible nucleotides (nt) that make up a DNA sequence (for proteins, this set can be redefined as the set of all possible amino acids). An indicator string x signals the locations of the motif occurrences (the range of x dependents on its specific definition and the model, see later sections for details). Following biological convention, we denote the *multi-alignment* of M instances of a motif of length L by an $M \times L$ matrix $\mathbf{A}$, in which each *column* corresponds to a *position* or *site* in the motif. The multi-alignment of all instances of motif *k* specified by the indicator string x in sequence y is denoted by $\mathbf{A}^{(k)}(x, y)$. We define a *counting matrix* $h(\mathbf{A}^{(k)})$ (or $h^{(k)}(x, y)$) for each motif alignment, where each column $h_l = [h_{l,1}, \ldots, h_{l,4}]^t$ is an integer vector with four elements (the superscript t denotes vector or matrix transpose), specifying the number of occurrences of

each nucleotide at position l of the motif. (Similarly we define the *counting vector* h_{bg} for the background sequence $y - \mathbf{A}$, where the somewhat abusive use of the minus sign means excluding all motif sub-sequences in $\mathbf{A}$ from y.) We assume that the nucleotides at position l of motif *k* admit a *position-specific multinomial distribution* (PSMD), $\theta_l^{(k)} = [\theta_{l,1}^{(k)}, \ldots, \theta_{l,4}^{(k)}]^t$. The ordered set of position-specific multinomial parameters of all positions of motif k, $\theta^{(k)} = [\theta_1^{(k)}, \ldots, \theta_{L^{(k)}}^{(k)}]$, is referred to as a *position weight matrix*. It is clear that the counting matrix $h^{(k)}$ corresponds to the *sufficient statistics* for estimating the PWM $\theta^{(k)}$. Formally, the problem of motif training is that of estimating $\theta^{(k)}$ given the multiple alignment $\mathbf{A}^{(k)}$, for each k; the problem of motif scan is that of inferring $x^{(n)}$ given a sequence $y^{(n)}$ and $\theta^{(k)}$, $\forall n, k$; and the problem of *de novo* motif detection is that of inferring $\mathbf{x}=\{x^{(1)}, \ldots, x^{(N)}\}$ and estimating $\boldsymbol{\theta}=\{\theta^{(1)}, \ldots, \theta^{(K)}\}$ simultaneously, given a set of sequences $\mathbf{y}=\{y^{(1)}, \ldots, y^{(N)}\}$. For simplicity, we omit the superscript k (motif type index) of the variable θ and the superscript n (sequence index) of the variables x and y wherever it is clear from the context that we are focusing on a generic motif type or a generic sequence.

2.2.4 The LOGOS Framework: a Modular Formulation

Without loss of generality, assume that the locations of motifs in a DNA sequence, as indicated by x, are governed by a **global distribution model** $p(x|\Theta_g, \mathcal{M}_g)$, and for each type of motif, the nucleotide sequence of all its instances (collected in an alignment matrix) jointly admits a **local alignment model** $p(\mathbf{A}(x,y)|x, \Theta_l, \mathcal{M}_l)$. Further assume that the background non-motif sequences are modeled by a conditional model, $p(y-\mathbf{A}(y,x)|x, \Theta_{bg}, \mathcal{M}_{bg})$, where the background nt-distribution parameters Θ_{bg} are usually assumed to be estimated *a priori* from the entire sequence. The symbols $\Theta_{[\cdot]}$ and $\mathcal{M}_{[\cdot]}$ stand for the parameters (e.g., the PWMs) and model classes (e.g., a product multinomial model as described in the sequel) in the respective submodels. Thus, marginalizing over all possible values of the indicator sequence x, the likelihood of a regulatory sequence y is:

$$\begin{aligned} p(y|\Theta, \mathcal{M}) &= \sum_x p(x|\Theta_g, \mathcal{M}_g) p(y|x, \Theta_l, \mathcal{M}_l, \Theta_{bg}, \mathcal{M}_{bg}) \\ &= \sum_x p(x|\Theta_g, \mathcal{M}_g) p(\mathbf{A}|x, \Theta_l, \mathcal{M}_l) p(y-\mathbf{A}|x, \Theta_{bg}, \mathcal{M}_{bg}), \end{aligned} \tag{2.1}$$

where $\mathbf{A} \triangleq \mathbf{A}(x, y)$. Note that Θ_l here is not necessarily equivalent to the PWMs, $\boldsymbol{\theta}$, of the motifs, but is a generic symbol for the parameters of a more general model of the aligned motif instances. (E.g., in the HMDM model to be defined shortly, Θ_l refers to the hyperparameters that describe a distribution of PWMs.)

Equation (2.1) makes explicit the modular structure of the probabilistic framework for generic motif models. The submodel $p(x|\Theta_g, \mathcal{M}_g)$ captures properties such as the frequencies of different motifs, the dependencies between motif occurrences, and the global organization of motif instances. On the other hand, the submodel $p(\mathbf{A}|x, \Theta_l, \mathcal{M}_l)$ captures the intrinsic properties within motifs that can help to improve sensitivity and specificity to genuine motif patterns. Depending on the value of the latent indicator x_t (e.g., motif or not) at each position t, y_t follows different probabilistic distributions, such as a specific nucleotide distribution of a particular position inside a motif or a background distribution. This probabilistic architecture is named **LOGOS**, for integrated **LO**cal and **G**l**O**bal motif **S**equence model.

As equation (2.1) suggests, the specific submodels in **LOGOS** can be designed separately, and they are roughly aligned with our specification of the actual computational tasks underlying the motif detection problem. The local model alone suffices to solve the motif training task, the global model plus a given set of motif representations suffices to answer the motif scan problem, and their combination represents the *de novo* motif detection problem. Recall that the graphical model formalism facilitates a modular combination of heterogeneous submodels, using the property of the product rule of the joint distribution. **LOGOS** is an instance of such a modeling strategy, and essentially facilitates a bottom-up approach for solving the complex *de novo* motif detection problem, by starting from relatively simpler subproblems. This strategy clearly exposes the main technical issues involved in the motif detection problem, which helps in analyzing existing algorithms and understanding their merits and limitations. It also enables one to design more sophisticated models in a piecewise manner to address different aspects of the problem without being overburdened by the complexity of the overall problem, and to envisage a straightforward path toward solving even more complex problems, such as joint modeling of motifs and gene expression patterns, by using

existing or designing new models for each problem (now viewed as subproblems) separately, and integrating them under the joint graphical model formalism.

2.3 An Overview of Related Work

In the following, we briefly review some representative models for motif detection in the literature. We will describe these models from the **LOGOS** point of view, by making explicit the background, local and global components of the model, even though almost none of the models were originally constructed and described in such a way, so that the pros and cons of these models can be clearly understood and compared.

2.3.1 Background Models

2.3.1.1 The models

It is generally assumed that the sequences outside the motifs have diverged sufficiently to be modeled as random background. Thus a simple but very popular model for all the non-motif nucleotides in the the background sequence is an *iid* multinomial model:

$$p(y - \mathbf{A}|x, \Theta_{bg}) = \prod_{t \in \mathbb{B}} \prod_{i \in \mathbb{N}} \left[\theta_{bg}\right]^{\mathbb{I}(y_t, i)} = \prod_{i \in \mathbb{N}} \left[\theta_{bg}\right]^{h_{bg,i}}, \tag{2.2}$$

where $\mathbb{B}$ is the set of indices of the background positions, and θ_{bg} denotes the vector of multinomial parameters of the background model, which is usually directly computed as the overall nucleotide frequency distributions of the entire input sequence, assuming that motif instances are sparse in the sequence and thus would not bias the estimated frequencies [Bailey and Elkan, 1995a; Hughes *et al.*, 2000; Liu *et al.*, 2001]. (This assumption is somewhat unwarranted in some early literature in which the input sequences are usually assumed to be sets of $100 \sim 200$-mers, each containing, say, one motif, which suggests a quite significant 10% motif coverage! e.g., [Cardon and Stormo, 1992; Lawrence *et al.*, 1993])

Several recent papers have stressed the importance of using a richer background model for the non-motif sequences [Thijs *et al.*, 2001; Liu *et al.*, 2001; Huang *et al.*, 2004]. In particular, a number

of higher-order Markov models have been explored by various authors and reportedly contribute to notable improvements in the performance of motif scan and *de novo* motif detection. Under a global kth-order Markov model for non-motif nucleotide sequences, the conditional probability of a single nucleotide j at site t is contingent on the k preceding bases following the usual Markov dependency definition

$$p(Y_t = i|X_t = bg) = p(Y_t = i|y_{t-1}, \ldots, y_{t-k}) = f_i(y_{t-1}, \ldots, y_{t-k}).$$

Thus the probabilities of all the background can be computed by enumerating all $(k+1)$-tuples of nucleotides in the entire sequence y (note that these probabilities need to be computed only once, and then stored for repeated references during probabilistic inference). The total time for this operation is $O(T)$, where T is the total length of the input sequences. One can also use a local kth-order Markov model, in which the conditional probability of a nucleotide i at position t, $f_i^t(\cdot)$, is estimated from a local window centered at position t.

2.3.1.2 The use of background models

As detailed in the sequel, one family of motif scan algorithms seek to score candidate sequence segments for their similarity to a known motif pattern. The background model plays an important role in formulating a good scoring function. For example, the standard *likelihood ratio* score for candidate segment $y_{t,t+L-1}$ at positions t to $t+L-1$ is computed as follows

$$r_t = \frac{p(y_{t,t+L-1}|\Theta_l, \mathcal{M}_l)}{p(y_{t,t+L-1}|\Theta_{bg}, \mathcal{M}_{bg})}. \tag{2.3}$$

A variant of the likelihood ratio score is the *log odds* score

$$l_t = \log r_t = \log p(y_{t,t+L-1}|\Theta_l, \mathcal{M}_l) - \log p(y_{t,t+L-1}|\Theta_{bg}, \mathcal{M}_{bg}). \tag{2.4}$$

From these two scoring schemes, it is apparent that, even though the motif model, which defines the probability of a motif segment, is the most important component in these scoring functions, a good background model will help to improve the contrast of motif to background, and therefore the discriminating power of r_t or l_t at each position.

Note that while probabilities are used in constructing the scoring functions, the scores themselves (e.g., likelihood ratio, log odds) cannot be interpreted statistically. Usually, they will be compared against an *ad hoc* cutoff value to generate computational motif predictions, and choosing the score cutoff values for each motif and background model is generally difficult. This may have contributed to the large number of false positive predictions seen in practice. To assess the significance for a set of predicted motif instances, Liu *et al.* [1995] developed a rank test that compares the prediction results from the study data with those from control data generated by a random shuffle of the study data. They applied a Wilcoxon signed rank test to the predictions made from paired (concatenated) study and control data, and obtain a p value of the prediction from the study data under the rationale that, under the null hypothesis, the motifs are equally likely to be solicited from either the study or the control sequences. Huang *et al.* [2004] proposed a p-value based scoring scheme, which computes the probability that the null (i.e., background) model can achieve a standard log-likelihood score for a candidate sequence segment at least as high as that of a signal (i.e., motif) model defined by PWMs. They developed an exact algorithm based on probability generating functions to compute the p-value for a general kth-order Markov background model with respect to motif models represented by PWMs. The CREME program by Sharan *et al.* [2003] proposed a number of closed-form statistical scores for assessing the significance of single motif abundance or abundance of a motif cluster (multiple spatially-close motifs) out of a subregion of a study sequence over that of the background sequences. Note that "abundance" (i.e., the number of motif matches), rather than the score of the matches, is tested for significance in the CREME program, and it was demonstrated to be a competent method to scan for CRMs in higher eukaryotic transcription regulatory sequences.

Generally, depending on the choice of grammatical models for global sequence annotation, the background model can be plugged in as a conditional model for the background state and contribute to various scoring functions for motif detection. For example, it can be used as an emission model under a background state in the case of a HMM global model (see §2.5 for details). Rather than contributing an r or l score, in these cases, the background model will contribute indirectly to the

posterior probability distribution of the indicator sequence x.

2.3.2 Local Models — for the Consensus and Stochasticity of Motif Sites

A local alignment model attempts to captures the consensus and the accompanied stochasticity of the set of binding sites (i.e., motif instances) corresponding to a certain TF.

2.3.2.1 Product multinomial model

The position weight matrix introduced in §2.2.3 is the most commonly used representation for a motif pattern in extant motif detection algorithms [Bailey and Elkan, 1995a; Hughes *et al.*, 2000; Liu *et al.*, 2001; Frith *et al.*, 2001; Liu *et al.*, 2002; Gupta and Liu, 2003]. Statistically, a PWM can be used to define a *product multinomial* (PM) model for every observed instance of a motif [Liu *et al.*, 1995]. Formally, given the PWM, $\theta = [\theta_1, \theta_2, \ldots, \theta_L]$, of a motif, the probability of an observed instance of this motif, which corresponds to a row in the motif alignment matrix $\mathbf{A}$, say, $A_m = [A_{m,1}, A_{m,2,}, \ldots, A_{m,L}]$, is

$$p(A_m|\Theta_l) = \prod_{l=1}^{L} \prod_{i \in \mathbb{N}} \left[\theta_{l,i}\right]^{\mathbb{I}(A_{m,l},i)}. \tag{2.5}$$

For an alignment of M motif instances, $\mathbf{A} = \{A_m\}_{m=1}^{M}$, the joint probability of all motif instances in $\mathbf{A}$ is

$$p(\mathbf{A}|\Theta_l) = \prod_{m=1}^{M} p(A_m|\Theta_l) = \prod_{l=1}^{L} \prod_{i \in \mathbb{N}} \left[\theta_{l,i}\right]^{h_{l,i}}. \tag{2.6}$$

Recall that $h \equiv \{h_{l,i}\}$ is the *nucleotide count matrix* associated with alignment $\mathbf{A}$, thus, $h_{l,i} = \sum_m \mathbb{I}(A_{m,l}, i)$.

The PM model inherently assumes that the nt-contents of positions within the motif are independent of each other. Thus, a PWM only models independent statistical variations with respect to a consensus pattern of a motif, but ignores potential couplings between positions inside the motif — a limitation that often weakens its ability to discern genuine instances of a motif from a very complex background that may harbor random recurring patterns, due to the low signal/noise ratios reflected in the likelihood-based scores computed from the PM model.

Given a set of aligned instances of a certain motif (i.e., a *training alignment*), under the PM model, the PWM of this motif can be obtained via maximal likelihood estimation (MLE), which is equivalent to computing the nt-frequency at each motif position. If the training alignment contains only a small number of motif instances, MLE tends to lead to a non-robust model (i.e., with a high variance associated with the estimates of the model parameters), which tends to generalize poorly to unseen instances of the same motif. For example, if a particular nucleotide does not appear at a certain position among all the instances in the training alignment, possibly just because the alignment is too small to be sufficiently representative, then every candidate instance from a new dataset that bears this nucleotide at this position (but is otherwise highly consistent with the motif consensus,) will be assigned a zero probability. This artifact is called *overfitting* in statistical learning, and should be avoided when learning from a small training dataset. In the motif modeling literature, the most popular remedy is to add to the actual count matrix h a uniform *pseudo-count matrix* (i.e., all elements of the matrix are equal) [Lawrence *et al.*, 1993; Bailey and Elkan, 1994], which can be regarded as the nt-count from an imaginary set of "motif instances". The column sum of the pseudo-count matrix, typically set to 1, can be understood as the total number of "imaginary motif instances" from which the "count" is obtained. The larger this number is, the more difficult to override the pseudo-counts with the actual counts from the training data.

Mathematically, incorporating uniform pseudo-counts into the MLE of a PWM is equivalent to introducing a symmetric Dirichlet prior (Appendix A.1) for the values of each column of the PWM, and the resulting motif model is also called a *product Dirichlet* (PD) model [Bailey and Elkan, 1995b; Liu *et al.*, 1995]. Note that pseudo-counts or PD models are primarily used for smoothing, rather than for explicitly incorporating prior knowledge about motifs, and the parameters are chosen *ad hoc*.

2.3.2.2 Constrained PM models

Although there are some obvious limitations of PWMs, they have proved to be reasonably effective in describing the set of sequences bound by a given TF and have shown considerable predictive

power [Stormo, 2000]. However, in an unsupervised *de novo* motif finding scenario where the PWMs have to be estimated *ab initio*, the estimated PWMs under the PM model, or even with pseudo-counts or symmetric Dirichlet priors, are sensitive to noise and random or trivial recurrent patterns (e.g., poly-N or repetitions of short k-mers such as CpG islands). Furthermore, the PM model is unable to capture potential position dependencies inside the motifs.

Various pattern-driven approaches have been developed to handle motifs with specific motif patterns. For example, in the early Lawrence and Reilly paper [1990], the authors introduced constraints on the parameters of the PWM to enforce palindromicity. Frech *et al.*'s method [1993] proposed to originate the PWMs from a highly conserved consensus core and then extend the core in one or both directions. Some of the recent methods provide *ad hoc* ways of allowing motifs to have two conserved blocks separated by a few background sites, such as splitting a "two-block" motif into two coupled sub-motifs [Liu *et al.*, 2001; Bailey and Elkan, 1995a]. The fragmentation model of Liu *et al.* [1995] allows an arbitrary $L < W$ positions in an aligned segment of width W to constitute the conserved motif sites.

Note that in addition to the nt-frequencies represented by the matrix elements, PWMs can also provide the *information content profile* [Schneider *et al.*, 1986] of the corresponding motif. The information content (IC) at a position l in a motif is given by

$$I_l = \log_2 |\mathbb{N}| + \sum_{i \in \mathbb{N}} \log_2 \theta_{l,i}, \tag{2.7}$$

and can be thought as a measure of how conserved position l is.

Keles *et al.* [2003] noted that the PWMs describing motifs with very different nt-specificities can have similar information profiles, and speculated that there is a direct relationship between the structural footprint of a TF and the information content profile of the corresponding motif. They developed a method that explicitly enforces nt-biases, e.g., high versus low information contents at various positions, when computing the MLE of the PWM from samples. They have proposed several canonical information content patterns, such as the one with a *U-shaped* contour, or a *bell-shaped* contour, to be plausible constraints for *de novo* motif detection, and have developed a sequential

quadratic programing method to solve the constrained optimization problem. A constrained EM algorithm was developed by Kechris *et al.* [2004] to incorporate similar IC constrains for estimating motif PWMs.

The IC-constraint approach represents a significant advance in learning local motif models because it takes into consideration the commonalities shared among motifs of different TFs (with different nt-specificities), and reveals something intrinsic to biologically genuine motifs. However, it defines a hard constraint that must be respected no matter how many actual motif instances are used to estimate the PWM, and cannot be overridden when the IC of an abundant novel motif deviates from the predetermined constraints.

2.3.2.3 Motif Bayesian networks

A recent article by Barash *et al.* [2003] proposed a family of more sophisticated representations to capture richer characteristics of motifs. These representations are based on directed probabilistic graphical models, i.e., Bayesian networks. Barash *et al.* suggested that a mixture of product multinomial models (MPM),

$$p_{MPM}(\mathbf{A}|\Theta_l) = \sum_j w_j p_{PM}(\mathbf{A}|\theta^{(j)}), \tag{2.8}$$

where w_j is the *weight* of the jth mixture component and $\theta^{(j)}$ is the PM parameter of the jth mixture component, can capture potential multi-modalities of the biophysical mechanism underlying the protein-DNA interaction between a TF and its target motif sites. Under the MPM model, a motif is characterized by multiple PWMs, each corresponding to a component PM model. Barash *et al.* further proposed a tree-based Bayesian network capable of capturing pairwise dependencies of nucleotide contents between nonadjacent positions within the motif. A natural combination of the above two models leads to a more expressive model, a mixture of trees, which captures more complex dependency characteristics of motifs. In a series of experiments with simulated and real data, Barash *et al.* showed that these more expressive motif models lead to better likelihood scores for motifs, and can improve the sensitivity and specificity of motif detection in yeast regulatory sequences under a simple scenario of motif occurrence (*i.e.*, at most one motif per sequence).

In principle, it is possible to construct even more expressive models for motifs by systematically exploiting the power of graphical models, although fitting more complex models reliably demands more training data. Thus, striking the right balance between expressiveness and complexity remains an open research problem in motif modeling.

2.3.3 Global Models — for the Genomic Distributions of Motif Sites

The local model of a motif pattern only creates aligned multiple instances of a motif, but does not complete the generation of the observed sequence set, even with the addition of the background model. It is necessary to have a set of "rules" that define where and how instances of one or multiple motifs are embedded in the background sequence so that they can constitute a sort of "language" or "program" interpretable by the TFs in a liquid solution environment, and in a TF composition and concentration sensitive manner. In the **LOGOS** framework, these "rules" are encoded in the global distribution model for the indicator variable sequence x that can specify the locations and organization of all motif instances.

2.3.3.1 The *oops* and *zoops* model

The probabilistic model for *de novo* motif detection developed by Lawrence and Reilly in their seminal 1990 paper [Lawrence and Reilly, 1990] assumes that each of the N input DNA sequences contains exactly one binding site of the same TF. This assumption is an idealization of a scenario in which a set of "co-regulated" genes are analyzed, and the co-regulation is induced by a single TF that can bind to a unique motif site present in the regulatory region of each of the regulated genes. Accordingly, this model is called a "one motif per sequence" (oops) model. Although hardly a realistic model, the oops model has historical importance (and is still in use in many contemporary programs) in that it provides a clean abstraction that helps in understanding the motif detection problem and points out a direction for formulating and upgrading the global model. Formally, let $y^{(n)} = \{y_t^{(n)}\}_{t=1}^{T_n}$, $y_t^{(n)} \in \mathbb{N}$, denote the data of the nth sequence with length T_n; and let $X^{(n)} \in \{1, \ldots, T_n - L + 1\}$ denote the latent *address* variable of a motif with length L in sequence n (the address of the motif is defined as the position of the most proximal nucleotide in the motif instance,

w.r.t. the end of the study sequence). The oops model assumes that the location of the (only) motif in each sequence admits a uniform distribution over all possible positions in the sequence. That is,

$$p(X^{(n)} = t) = \frac{1}{T_n - L + 1}. \tag{2.9}$$

Given $x^{(n)}$, the oops model assumes that nucleotides at positions not corresponding to the motif, hence falling into the background, are independently and identically distributed; whereas the nucleotides of all positions within a motif instance jointly follow a local motif model (e.g., a PM model as in [Lawrence and Reilly, 1990]). Motif instances in different input sequences are assumed to be independent and identically distributed. Thus, given the PWM θ of the motif and θ_{bg} for the background, and denoting the nt-count vector of the entire sequence $y^{(n)}$ by $h^{(n)}$, the joint probability distribution of the observed sequence $y^{(n)}$ and the latent addresses $x^{(n)}$ of the motif therein is:

$$\begin{aligned} p(X^{(n)} = t, y^{(n)} | \Theta = \{\theta, \theta_{bg}\}) &= p(X^{(n)} = t) p(y^{(n)} | X^{(n)} = t, \Theta = \{\theta, \theta_{bg}\}) \\ &= \frac{1}{T_n - L + 1} \prod_{l=0}^{L-1} \prod_{j \in \mathbb{N}} \left[\theta_{l,j}\right]^{\mathbb{I}(y^{(n)}_{t+l}, j)} \cdot \prod_{j \in \mathbb{N}} \left[\theta_{bg,j}\right]^{h^{(n)}_{bg,j}}, \\ &= \frac{1}{T_n - L + 1} \prod_{l=0}^{L-1} \prod_{j \in \mathbb{N}} \left[\frac{\theta_{l,j}}{\theta_{bg,j}}\right]^{\mathbb{I}(y^{(n)}_{t+l}, j)} \cdot \prod_{j \in \mathbb{N}} \left[\theta_{bg,j}\right]^{h^{(n)}_{j}} \end{aligned} \tag{2.10}$$

which leads to the following posterior distribution of the latent variable $X^{(n)}$:

$$p(X^{(n)} = t | y^{(n)}, \Theta) = \frac{\prod_{l=0}^{L-1} \prod_{j \in \mathbb{N}} \left[\theta_{l,j} / \theta_{bg,j}\right]^{\mathbb{I}(y^{(n)}_{t+l}, j)}}{\sum_{t'=1}^{T-L+1} \prod_{l=0}^{L-1} \prod_{j \in \mathbb{N}} \left[\theta_{l,j} / \theta_{bg,j}\right]^{\mathbb{I}(y^{(n)}_{t'+l}, j)}}. \tag{2.11}$$

Thus, the probability of position t being a motif start address is proportional to the likelihood ratio of a sub-sequence of length L started at t being a motif sequence with respect to its probability of being a background sequence, which is exactly the likelihood ratio score we described in §2.3.1.

A simple extension of the oops model is the *zoops* model, for "zero or one motif per sequence", adopted by Bailey and Elkan [1994] in their precursor of the MEME algorithm. As the name suggests, this model is slightly more flexible than oops. For each sequence, zoops introduces an indicator variable $Z_n \in \{0, 1\}$, which indicates the presence ($Z_n = 1$) or absence ($Z_n = 0$) of

a motif instance in sequence n. The prior distribution of Z_n can be defined as a simple Bernoulli distribution, $Z_n \sim Ber(\alpha, 1-\alpha)$, and the indicator is independent and identically distributed for each sequence. Under this setting, the conditional likelihood of the observed sequence is

$$\begin{aligned} p(y^{(n)}|X^{(n)}=t, Z_n=1, \Theta=\{\theta,\theta_0\}) &= \frac{1}{T_n-L+1}\prod_{l=0}^{L-1}\prod_{j\in\mathbb{N}}\left[\frac{\theta_{l,j}}{\theta_{0,j}}\right]^{\mathbb{I}(y_{t+l}^{(n)},j)}\cdot\prod_{j\in\mathbb{N}}\left[\theta_{0,j}\right]^{h_j^{(n)}}, \\ p(y^{(n)}|Z_n=0, \Theta=\{\theta,\theta_0\}) &= \prod_{j\in\mathbb{N}}\left[\theta_{0,j}\right]^{h_j^{(n)}}. \end{aligned} \tag{2.12}$$

Thus the probability of having a motif at position t of sequence n is regularized by the prior probability of motif presence,

$$\begin{aligned} p(X^{(n)}=t, Z_n=1|y^{(n)},\Theta) &= \frac{p(y^{(n)}|X^{(n)}=t, Z_n=1,\Theta)p(X^{(n)}=t)p(Z_n=1)}{p(y^{(n)}|\Theta)} \\ &= \frac{\prod_{l=0}^{L-1}\prod_{j\in\mathbb{N}}\left[\theta_{l,j}/\theta_{0,j}\right]^{\mathbb{I}(y_{t+l}^{(n)},j)}}{\sum_{t'=1}^{T-L+1}\prod_{l=0}^{L-1}\prod_{j\in\mathbb{N}}\left[\theta_{l,j}/\theta_{0,j}\right]^{\mathbb{I}(y_{t'+l}^{(n)},j)}+\frac{(1-\alpha)(T_n-L+1)}{\alpha}}. \end{aligned} \tag{2.13}$$

2.3.3.2 General uniform and independent models

Essentially, both the oops and zoops models are *uniform* (over all possible positions in a study sequence) and *independent* (between motif instances and between study sequences) global models, or in short, UI models, which are intended for simple and idealized motif-bearing sequences. It is straightforward to generalize the baseline UI model to handle slightly more complex scenarios, such as multiple motifs per sequence. For example, rather than allowing at most one motif per sequence, some motif detection algorithms assume that a fixed number of motif instances can be present independently with uniform probability at all possible locations in a sequence [Bailey and Elkan, 1995a]. That is, the joint distribution of the addresses of, say, M motif instances in a study sequence n, $x^{(n)}=\{x_1^{(n)},\ldots,x_M^{(n)}\}$, can be written as $p(x^{(n)})=\prod_{m=1}^{M}p(x_m^{(n)})$, where $p(x_m^{(n)}=t)$ is the prior probability of the mth motif instance at location t in sequence n, in this case, a uniform distribution over all valid t's and the same for all $m=1,\ldots,M$. It can be shown that the marginal posterior probability of any one of the addresses under this setting, $p(x_m^{(n)}|y^{(n)},\Theta)$, is proportional

to the likelihood ratio score of each sequence position (i.e., Eq. (2.11)), and the joint posterior of all addresses is the product of the posterior probabilities of individual addresses, $p(x^{(n)}|y^{(n)},\Theta) = \prod_m p(x_m^{(n)}|y^{(n)},\Theta)$. This model is called an *M motifs per sequence* model (*mops*), and is used in various contemporary algorithms, such as MEME. A drawback of this approach is the requirement of a prespecified number of motif instances. An inaccurately supplied number will lead to either significant false positives, or false negatives, or both. Also problematic is that the mops model ignores possible constraints on co-occurrences of motifs.

A slightly more sophisticated model for multiple motif locations per sequence is an *iid* Bernoulli indicator model [Liu *et al.*, 1995], which assumes that each host sequence y is associated with a binary indicator sequence, $x = (x_1, \ldots, x_T), x_t \in \{0, 1\}$, where 0 signals background and 1 signals a motif starting at position t; each X_t is an independent Bernoulli random variable, $x_t \sim Ber(\alpha)$; and the Bernoulli parameter $[\alpha, 1-\alpha]$ follows a Beta prior. This model is essentially a clustering model for all possible L-mers of the sequences, allowing any L-mer to be either a motif or background.

Under both the mops and Bernoulli indicator models, there is no formal *model constraint* to prevent having overlapping motif instances. Although overlapping motifs are possible in real genomic sequences, the possibility of overloading every sequence position with multiple motif instances is not desirable. Therefore, in practice, both models are augmented heuristically with an artificial *non-overlapping constraint*, which requires that no L-mer is allowed to harbor, say, more than l motif start positions, where $1 < l \leq L$. This constraint is enforced by either rescaling the joint posterior $p(x|y)$ (as in an EM-based inference strategy for MEME [Bailey and Elkan, 1995a]), or simply throwing away any overlapping motif samples (when using a Gibbs-sampling-based inference strategy [Liu *et al.*, 1995]). Nevertheless, these heuristics may result in inconsistencies between the computed motif distribution and the one defined by the model, and incur a sizable overhead due to wasteful computations.

Despite their simplicity and some unwarranted heuristic assumptions, UI models appear to be competent in motif scan and *de novo* motif detection for bacterial or simple yeast sequence sets, in

which the input sequences are usually short and enriched (e.g., pre-screened according to mRNA co-expression). But some recent studies including our own experiments suggest that the correctness of motif detection based on the UI assumptions starts to break down for less well pre-screened input sequences or for those with clustered motif occurrences, such as the *Drosophila* gene regulatory sequences [Berman *et al.*, 2002]. Higher eukaryotic genomes indeed present a challenge to the computational identification of motifs because of their long non-coding regions and large number of repeat elements.

2.3.3.3 The dictionary model

Bussemaker *et al.* proposed a novel formulation of the motif-finding problem, which is based on word segmentation and dictionary construction [Bussemaker *et al.*, 2000]. In their MobyDick algorithm, they view the regulatory DNA sequences as sentences written in an unknown language built from an alphabet of 4 characters (i.e., $\mathbb{N} = \{A, T, G, C\}$ as defined previously), with no separators between words. (In fact, some major human languages, such as Chinese and Japanese, are of this kind, although using a much larger alphabet, e.g., $> 10^4$, for Chinese.) Under this framework, the motif-finding problem can be cast as finding over-represented words from consecutive lists of characters, and the algorithm boils down to an iterative procedure alternating between building up a dictionary of words and estimating the values of parameters of a language model from a given word segmentation (e.g., word frequencies), and finding the optimum segmentation of the sequence given the dictionary and the language model. From the **LOGOS** point of view, the noise-less "words" in the motif dictionary represent a deterministic local model of the motifs. Consequently, a degenerate motif pattern could be represented by several similar words during the construction of the dictionary, which can be merged into a consensus afterward. The "language model" adopted by the MobyDick algorithm consists of an array of word-usage probabilities and an assemblage scheme (analogous to a "grammar" of word usage in natural language) of the sequences, which manifests a novel global model for motif distribution, and largely contributes to the strength of the MobyDick algorithm. This language model assumes that each word is associated with a frequency of its

usage, and a sequence is realized by a non-overlapping concatenation of words sampled according to their frequency (which implicitly assumes that the background model corresponds to a large set of short words with low frequencies). The conditional probability of a sequence given a possible segmentation specified by indicator sequence x under this setting is

$$p(y|x) = \frac{1}{Z} \prod_k \rho_k^{n_k(x)}, \tag{2.14}$$

where ρ_k denotes the frequency of word k, $n_k(x)$ denotes the counts of word k in y under segmentation x, and Z is a normalization constant (i.e., the partition function).

Some key advantages of this global model are its emphasis on combinatorial analysis of a large set of potential motifs (up to all possible substring patterns of k-mers allowable by the computing resource), and an explicit non-overlapping constraint on individual substrings induced by the word segmentation. Since the number of segmentations of an average-sized sequence could be huge, computing the partition function Z of Eq. (2.14) and various derivatives of Z is non-trivial, and a dynamic programming algorithm is developed.

To account for the sequence variations of each motif pattern, in a recent paper, Gupta and Liu [2003] extended the MobyDick model to a stochastic dictionary (SD) model by replacing the words in the dictionary with PWMs. From a **LOGOS** point of view, this corresponds to upgrading the local model of MobyDick from deterministic words to PM. Let x denote a word-segmentation of sequence set y, $\mathcal{H} = \{h^{(1)}, \ldots, h^{(D)}\}$ denote the set of nt-count matrices for all the words (with PWMs $\{\theta^{(k)}\}_{k=1}^{D}$) due to segmentation x, and $\mathcal{N} = \{n_1, \ldots, n_D\}$ denote the counts of word occurrences. The complete data likelihood given all PWMs $\boldsymbol{\theta}$ and the word usage probabilities ρ is:

$$p(\mathcal{N}, \mathcal{H}, x|\boldsymbol{\theta}, \rho) \quad \propto \quad \prod_{k=1}^{D} \rho_k^{n_k} \prod_{l=1}^{L_k} \prod_{j=1}^{4} [\theta_{l,j}^{(k)}]^{h_{l,j}^{(k)}}. \tag{2.15}$$

Essentially, SD adopts a specific distribution model for motif instances which treats the observed sequences as being generated by concatenating words independently drawn from a dictionary according to a vector of word usage probabilities, while retaining the PM model for aligned

motif instances. This is equivalent to upgrading the UI model to a finite mixture model with mixture components being all the PWMs in the dictionary weighted by the word usage probabilities. Due to this nice connection to the conventional motif models, many of the modeling ideas originally introduced to the conventional models were readily adaptable to the SD model, such as smoothing the model with conjugate priors for the PWM and word usage probabilities, and an extension allowing stochastic insertions and deletions in motif instances (to model gaped motifs.) Since in the model SD, the motif indicator X is defined as a segmentation variable, with a huge state space that prohibits exact inference, inference and parameter estimation are performed using a Monte Carlo procedure.

2.3.3.4 The sliding-window approaches

The uniform and independent models and the word-segmentation models described above treat all regions in a sequence equally, ignoring potential coupling of multiple motif instances in any subregions of the sequence. However, as discussed in the introduction, in higher eukaryotic genomes, motifs are often organized into *cis*-regulatory modules, in which the motif occurrences tend to be significantly enriched compared to the background region, and encode some complex combinatorial signals. This architecture implies that during motif scan, locally clustered weak motif signals may need to be treated with higher weights because they may suggest co-occurring weak binding sites in a CRM; whereas an occasional seemingly strong signal out of a long stretch of sequence with low score may need to be weighted lower because it may be just a spurious signal in the background.

The sliding-window method is one of the most popular approaches for motif and CRM prediction that tries to incorporate the aforementioned architectural features of motif distribution [Halfon *et al.*, 2002; Papatsenko *et al.*, 2002; Rajewsky *et al.*, 2002; Nazina and Papatsenko, 2004]. Typically, a sliding-window method counts the number of matches of some minimal strength to given motif patterns within a certain window of DNA sequences using certain scoring functions, such as a likelihood ratio (when the motifs are represented by PWMs [Sharan *et al.*, 2003]) or the k-m score

(when the motifs are represented by deterministic words) [Papatsenko *et al.*, 2002]. From a modeling point of view, this family of algorithms assumes that motifs are uniformly and independently distributed only within each window. An *ad hoc* window size needs to be specified and careful statistical analysis of matching strength is required to determine a good cutoff or scoring scheme.

The CREME algorithm [Sharan *et al.*, 2003] uses a comparative genomic approach to identify the putative CRM regions (thus avoiding the need to specify the window size), and a number of sophisticated scoring functions were proposed to measure the statistical significance of local enrichment of candidate motif matches in these regions.

Nazina and Papatsenko [2004] addressed the issue of compensating the matching scores for co-occurring weak motif sites using an updatable "word-frequency" measure, which leads to higher scores for motifs occurring more frequently within a window of a given size. This approach is analogous to a MobyDick model applied to each window. A sliding-window version of the stochastic dictionary model was used by Rajewsky *et al.* in their Ahab/Argos program [Rajewsky *et al.*, 2002].

2.3.3.5 The hidden Markov model

Another way to handle sequences bearing rich motif content and architecture is to explicitly model the organizations of the motifs using a stochastic sequential model that encodes "rules" to generate such motif organizations. For example, the program Cister [Frith *et al.*, 2001] assumes that the indicator sequence of a study sequence, x, admits a 1st-order Markov model,

$$p(x) = p(x_1) \prod_{t=2}^{T} p(x_t|x_{t-1}), \tag{2.16}$$

whose state space consists of background states and motif states. The occurrences of motifs and CRMs are induced by an emission model, $p(y_t|x_t)$, which generates state-specific nucleotide outputs belonging to a motif or the background. Note that the stochastic rules of the motif organization, such as how often a CRM appears, how long a CRM tends to be, and how often motifs appear in a CRM, are encoded in the state-transition probabilities of these indicator variables. Since the indicators x are not observed, this is a classical hidden Markov model with discrete output.

An HMM for motif scan renders both the window size and the score cutoff unnecessary, and takes into account not only the strengths of motif matches, but also the spatial distances between matches (arguably more informative than co-occurrences within a window). The hidden Markov model used in Cister translates to a set of soft specifications of the expected CRM length and the inter-motif distance (i.e., in terms of geometric distributions). However, since training data for fitting the HMM parameters hardly exists, these parameters have to be determined based on empirical guesses.

2.3.4 Other Models

The local and global models discussed in the previous sections concern pure DNA sequence data, and implicitly assume that the sequences to be analyzed come from a single species. With the availability of near complete sequences of several complex genomes, such as human and *Drosophila*, and the anticipation of sequencing more evolutionarily related species in near future, comparative genomic analysis of sequences from multiple evolutionarily related species has become a promising direction for *in silico* motif detection [Pennacchio and Rubin, 2001; Rubin, 2001; Wasserman and Sandelin, 2004]. The emergence of high-throughput gene expression or protein-binding profiling techniques, such as microarray analysis [Shalon *et al.*, 1996] and ChIP-array analysis [Ren *et al.*, 2000], provides another source of information to decode the transcription regulatory program. In particular, joint analysis of regulatory sequences together with the expression patterns of the genes regulated by these sequences appears to be a practical approach for motif detection, and is potentially more informative than methods solely based on sequence data [Segal *et al.*, 2003b; Wasserman and Sandelin, 2004]. A detailed discussion of motif detection models along these two directions is beyond the scope of this thesis. In the following, we briefly overview major research along these lines, and we point out their connection to the **LOGOS** framework and how an integration can be pursued.

2.3.4.1 Comparative genomic approach

Under the assumption that mutations within functional regions of the genome will accumulate more slowly than mutations in regions without sequence-specific functions, the comparison of sequences from orthologous genes and their associated regulatory regions can indicate segments that might direct transcription. For the prediction of motifs and CRMs, a major family of algorithms motivated by comparative genomic analysis is phylogenetic footprinting [Blanchette *et al.*, 2002; Ureta-Vidal *et al.*, 2003]. A phylogenetic footprinting algorithm usually consists of three components: 1) defining suitable orthologous gene sequences for comparison, 2) aligning the promoter sequences of orthologous genes, and, 3) identifying segments of significant conservation. For each component, there exist a wide variety of methods/programs, whose details are beyond the scope of this thesis. To name a few, to generate a multiple alignment of regularity regions of orthologous genes, BLASTZ [Schwartz *et al.*, 2003] and LAGAN [Brudno *et al.*, 2003] are often used because they tend to find a proper balance between preserving short stretches of highly conserved regions and finding long but marginally conserved regions; to interpret the aligned data, one can use a VISTA [Loots *et al.*, 2002] browser to plot the amount of nt-identity across the aligned sequences from multiple species within a sliding window, or use a dynamic programming algorithm to find an optimal segmentation of homogeneous and heterogeneous regions from the alignment [Xing *et al.*, 2001].

Kellis *et al.* [2003] developed a suite of techniques that work together for whole genome motif detection on the basis of within-genome over-representativeness and cross-genome evolutionary conservation of motif patterns. In their approach, they adopt a regular expression representation for a motif pattern, and begin with exhaustive enumeration of all over-represented regular expressions in all the genomes under their study to generate a long list of candidate core motifs referred to as "partial words." Then they iteratively prune this list based on three evolutionarily-motivated criteria drawn from an empirical study of the conservation patterns of the *gal4* motifs in multiple yeast species: 1) overall genome-wide intergenic conservation, 2) preference for intergenic (i.e. between

gene sequences) conservation over genic (i.e., within a gene) conservation, 3) differential conservation in upstream-only vs. downstream-only regions. Finally they extend the qualified core regular expressions to include neighboring positions, collapsing degenerate regular expressions based on sequence similarity and genome-wide co-occurrence. They reported a remarkable analysis of the *Saccharomyces cerevisiae* genome in light of draft sequences of three related yeast species, in which they confirmed numerous well characterized motifs and identified several previously unknown motifs. It is noteworthy that Kellis's approach does not attempt to pursue formal modeling of the motif properties under an evolutionary context, such as a stochastic motif model for intra-species variation, evolutionary model for inter-species variations, global distribution model for motif organization, etc.. Their approach also heavily relies on high-quality gene finding results, and the assumption that even for each instance of a motif, one can expect an one-to-one correspondence of its presence across species (i.e., several particular instances of a motif, one in each species, are evolved from the same ancestor, whereas other instances of the same motif in a species have their own counterparts in other species), and that this correspondence can be revealed in a multiple alignment of the whole genomes (which implicitly assumes that, certain, but not arbitrary instances of a motif across species are "orthologous", and multiple "paralogous" instances of the same motif in each particular species are order-preserving across species so that they can be aligned to their respective orthologous counterparts in other species in a single multiple alignment). This is a rather strong assumption, which may not hold for higher eukaryotic genomes.

In summary, extant phylogenetic footprinting and other comparative genomics approaches are restricted to short regulatory sequences from very closely related species, or genomes of simple organisms in which high-quality gene identification is possible and the regulation involves simple motif organization. For evolutionarily distant species and large complex genomes, not only are the non-coding regions hard to identify and align, but also the assumption that the aligned non-coding sequences are orthologous is often not substantiated for small and degenerate functional elements such as motifs and CRMs. Formal modeling of motifs under an evolutionary context is still an open and little addressed problem, and could lead to important methodological advances in motif

detection.

2.3.4.2 Joint models for motifs and expression profiles

A direct consequence of combinatorial interactions between TFs and their corresponding motifs is the highly regulated and coordinated transcription of the genes under their control. It is reasonable to expect that the expression profile of genes in a certain genome must bear some information useful for predicting the presence and the identity of regulatory motifs. Indeed, this idea was used even when motif detection models and algorithms were still in their infancy, although in a rather primitive fashion. For example, many algorithms assume that co-expression of a set of genes implies their co-regulation, and furthermore, implies co-existence of instances of the same motif in their respective regulatory regions [Cardon and Stormo, 1992; Helden *et al.*, 2000]. Unfortunately, this assumption may not hold true for genes under complex control mechanisms. Among the more sophisticated and explicit applications of expression profiles, especially high throughput data from mRNA microarrays, for inferring motif patterns are the "regression-based methods," which try to capture some of the interactions among motifs (e.g., their cumulative effects), and relate them to gene expression levels via a deterministic function. For example, Bussemaker *et al.* [2001] used a linear regression model to capture correlations between the abundances of regulatory elements and gene expression. It is straightforward to generalize this method to a logistic regression model that captures non-linear response (i.e., binary "on" and "off" response) between gene expression and motif presence. Keles *et al.* [2004] proposed a more expressive logic regression function to capture complex interactions, such as logical OR, between motifs. Segal *et al.* [2003a; 2003b] went beyond merely fitting deterministic mapping functions between motif spectrum and gene expression, and proposed to jointly model probabilistic distributions of gene expressions recorded in time series or other experimental conditions, together with the locations and stochastic variations of motifs, using a large-scale probabilistic graphical model. Segal's approach spearheads an emerging trend of using the systems biology principle in computational analysis of biological data, that is, combining correlated data from heterogeneous sources for complex prediction tasks. A recent publication went

as far as combining gene expressions, motif-bearing sequences and sequences from evolutionarily related species under a unified computational protocol [Chiang *et al.*, 2003]. However, although some extant models and algorithms have reached an unprecedented level of complexity in terms of the size and diversity of objects being modeled, the level of sophistication and the biological foundation of the models for different aspects of the heterogeneous biological data are far from satisfactory and lack systematic verification. For example, in almost all cases, the simple PM model and UI model were used to model the local and global aspects of motifs. More investigations are needed to make full and appropriate use of the systems biology approach for *in silico* motif detection.

2.3.5 Summary: Understanding Motif Detection Algorithms

The design of motif detection algorithms can be understood as a quest for realistic and well-founded mathematical models that capture the biological nature of the structure, organization and function of TF binding sites in the genome; for efficient computational algorithms that solve such models; and for data fusion strategies that integrate diverse sources of experimental data and produce consistent and both biologically and mathematically interpretable hypotheses and predictions. Different algorithms can differ in only one of these three aspects (e.g., using different techniques, such as EM or Monte Carlo methods, for probabilistic inference), or more aspects. To understand the essential differences between different motif detection algorithms, it is important to analyze these algorithms with respect to the aforementioned three aspects and identify their merits and deficiencies in these aspects, so that improvement can be made systematically and purposefully. In this section, we attempted to provide an overview of a wide range of *modeling strategies* currently in use, which is, to our opinion, the most important aspect that determines the capacity of a motif detection algorithm. In Table 2.1, we briefly summarize representative motif detection algorithms and/or software packages in the literature in terms of their model specificities, as well as the computational algorithm and data fusion strategies.

As Table 2.1 makes clear, many early methods lack any mechanism for incorporating knowledges about meta-sequence features of the motifs at both the local and global level. Recent studies

Table 2.1: A summary of popular motif detection software/algorithms

Software/ Algorithm	local model	global model	data fusion	inference algorithm	task	ref.
MEME	PM/PD	UI (mops)	-	EM	*de novo*	[Bailey and Elkan, 1995a]
BioProspector	PM/PD	UI (mops)	-	Gibbs	*de novo*	[Liu *et al.*, 2001]
AlignACE	PM	UI (Bernouli indicator)	-	Gibbs	*de novo*	[Hughes *et al.*, 2000]
MobyDick	word	word concatenation	-	DP	*de novo*	[Bussemaker *et al.*, 2000].
SD	PM	word concatenation	-	Gibbs	*de novo*	[Gupta and Liu, 2003]
Cister	PM	HMM	-	Forward-Backward	scan	[Frith *et al.*, 2001]
CREME	word/PWM	window	-	various tests	scan	[Sharan *et al.*, 2003]
Ahab/Argos	word/PWM	window	-	exhaustive search	scan/ *de novo*	[Rajewsky *et al.*, 2002]
PRM	PM	UI	sequence + microarray	BP	*de novo*	[Segal *et al.*, 2003a; Segal *et al.*, 2003b]
FootPrinter	word	UI	sequence of multiple species	Phylogenetic Footprinting	*de novo*	[Blanchette and Tompa, 2003; Blanchette *et al.*, 2002]

have tried to address these problems from several different angles. Though these attempts head in the direction of more expressive motif models, it is not clear whether these ideas can be integrated to assemble a powerful yet transparent and computationally efficient motif detection algorithm. There is a trend of combining heterogeneous source of data and constructing composite models for such data from simple building blocks. It is argued that the correlations and internal dependencies between different sources of data could serve to validate each other, and lead to more reliable predictions. However, it is also possible that, due to the lack of sophistication of each component submodel for different aspects of a complex model, and the difficulties of performing exact inference computations on such models, errors resulting from approximate inference or from deficiencies of each submodel may tend to propagate rather cancel. Thus, large composite models built from a plethora of heterogeneous components may generate highly unreliable predictions if the biological legitimacy, computational tractability and quality of approximate computation of the model components are unwarranted.

In summary, numerous advances notwithstanding, successful results of *in silico* motif discovery remain limited to simple bacterial and yeast sequences. Performance on sequences with complex intra- or inter motif structures are far less robust. One of the possible causes for compromised

generalizability and scalability of many extant algorithms is believed to be their incorrect independence assumptions about motif sites and motif occurrences, which leads to inability to capture possible intra-motif spatial dependencies corresponding to the signature physical structure for unique recognition and stable molecular interaction, and inter-motif dependencies that elicit synergy or simply avoid overlap. As mentioned in the introduction, there have been some recent attempts at addressing these issues at various levels [Hertz and Stormo, 1999; Helden *et al.*, 2000; Frith *et al.*, 2001; GuhaThakurta and Stormo, 2001; Rajewsky *et al.*, 2002; Barash *et al.*, 2003; Nazina and Papatsenko, 2004]. In the following, we will develop an expressive modular motif model that builds on these previous lines of research.

2.4 MotifPrototyper: Modeling Canonical Meta-Sequence Features Shared in a Motif Family

For the gene regulatory system to work properly, a TF must display much higher binding affinities to its own recognition sites than to non-site DNA. This correspondence suggests possible regularities in the DNA motif structure that match the structural signatures in the DNA-binding domains of their corresponding TFs. Can these regularities hidden in the true DNA motif patterns be exploited to improve sensitivity and specificity during motif discovery? As Michael Eisen has pointed out (private communications), there should be great potential for improving motif recognition by modeling and exploiting such structural regularities.

As reviewed in the previous sections, all extant local models of DNA motifs are essentially motif-specific and are intended to generalize only to different instances of the same motif. An important issue that remains little addressed is how to build models that can generalize over different motifs that are somewhat related (for instance, belonging to a family of regulatory sites that are targets of TFs bearing the same class of binding domains) even though they do not share apparent commonality in consensus sequences. This issue is important in computational motif analysis because,

- often, we want to roughly predict the biological property of an *in silico* identified motif pattern

(*e.g.*, to what kind of TFs it is likely to bind) to reduce the search space of experimental verification;

- we may need to introduce some generic but biologically meaningful bias during *de novo* motif detection so that we can distinguish a biologically plausible binding site (*i.e.*, specifically recognizable by some TF) from a trivial recurring pattern (*e.g.*, micro-satellites);

- we may also want to restrict attention to a particular class of proteins in performing tasks such as: "find a regulatory site that potentially binds to type X TF", or "find co-occurring regulatory sites that can be recognized by type X and type Y TFs, respectively."

These tasks are important in inferring gene regulatory networks from genomic sequences, possibly in conjunction with relevant expression information.

In this section, we address the problem of modeling generic features of *structurally* but not *textually* related DNA motifs, that is, motifs whose consensus sequences are entirely different, but nevertheless share "meta-sequence features" reflecting similarities in the DNA binding domains of their associated protein recognizers. We present MotifPrototyper, a profile hidden Markov Dirichlet-multinomial (HMDM) model which can capture regularities of *nt-distribution prototypes* and *site-conservation couplings* typical to each particular family of motifs that corresponds to TFs with similar types of structural signatures in their DNA binding domains. Central to this framework is the idea of formulating a profile motif model as a family-specific structured Bayesian prior model for the PWMs of motifs belonging to the family being modeled, thereby relating these motif patterns at the *meta-sequence level*. In the following, after a brief discussion of the biological motivation underlying our model, we will first develop the theoretical framework of the HMDM model, and then show how to learn family-specific profile HMDMs, or MotifPrototypers, from biologically identified motifs categorized in standard biological databases; how the model can be used as a classifier for aligned multiple instances of motifs; and most importantly, how a mixture model built on top of multiple profile models can facilitate Bayesian estimation of the PWM of a novel motif. The Bayesian estimation approach connects biologically identified motifs in the database

to previously unknown motifs in a statistically consistent way (which is not possible under the single-motif-based representations described previously) and turns *de novo* motif detection, a task conventionally cast as an *unsupervised* learning problem, into a *semi-unsupervised* learning problem that makes substantial use of existing biological knowledge.

2.4.1 Categorization of Motifs Based on Biological Classification of DNA Binding Proteins

TF categorization from TRANSFAC r6.0				
Superclass	class	# of training matrices		# of test alignments
1: *Basic domains*	1.1 : Leucine zipper factors (bZIP)	34	82	48
	1.2 : Helix-loop-helix factors (bHLH)	13		
	1.3 : Helix-loop-helix/leucine zipper factors	22		
	1.4 : NF-1	6		
	1.5 : RF-X	1		
	1.6 : bHSH	6		
2: *Zinc-coordinating DNA-binding domains*	2.1 : Cys4 zinc finger or nuclear receptor type	14	52	36
	2.2 : diverse Cys4 zinc fingers	13		
	2.3 : Cys2His2 zinc domain	21		
	2.4 : Cys5 cycteine-zinc cluster	3		
	2.5 : Zinc fingers of alternating composition	1		
3: *Helix-turn-helix*	3.1 : Homeo domain	41	76	64
	3.2 : Paired box	6		
	3.3 : Fork head / winged helix	4		
	3.4 : Heat shock factors	7		
	3.5 : Tryptophan clusters	17		
	3.6 : TEA domain	1		
4: *beta-scaffold factors*	4.1 : RHR (Rel homology region)	15	60	13
	4.2 : STAT	13		
	4.3 : p53	2		
	4.4 : MADs box	9		
	4.5 : β-Barrel α-helix TFs	1		
	4.6 : TATA-biding proteins	3		
	4.7 : HMG	5		
	4.8 : Heteromeric CCAAT factors	9		
	4.9 : Grainyhead	2		
	4.10: Cold-shock domain factors	0		
	4.11: Runt	1		
0: *Other TFs*	...	...	1	

Table 2.2: The TRANSFAC categorization of transcription factors (for the training set, the counts are made on TFs that have more than 10 biologically identified binding sites; for the test set, TFs with at least 6 sites are counted.)

Unlike proteins or genes, which usually have a one-to-one correspondence to monomer sequences and hence are directly comparable based on sequence similarity, a DNA motif is a collective object referring to a set of similar short DNA substrings that can be recognized by a specific protein transcription factor. Different motifs are characterized by differences in consensus, stochasticity

and number of occurrences. Since each motif usually corresponds to a profile of gap-less, multiple-aligned instances rather than a single sequence as for genes and proteins, comparisons based on sequence similarity for different motif patterns are not as straightforward as for genes or proteins.

From a biological point of view, perhaps the most informative way of categorizing DNA motifs is according to the regularities of the DNA-binding domains of their corresponding transcription factors. Advances in structural biology have provided an extensive categorization of the biophysical structures of DNA-binding proteins. The most recent update of the TRANSFAC database [Wingender *et al.*, 2000] lists 4219 entries, many of which are homologous proteins from different species but are nevertheless indicative of the vast number of transcription factors now known that regulate gene expression. Table 2.2 shows a fraction (the top two levels in the cluster hierarchy) of the TRANSFAC categorization of TFs. This categorization provides a good indication of the types of binding mechanisms involved in motif-TF recognition. For concreteness, the following is a brief summary of the structural regularities of four of the major classes of DNA-binding proteins, paraphrasing [Stryer, 1995]. Due to the correspondence between a TF and a DNA motif, the TF categorization strongly suggests possible features in the structure of motif sequences that are intrinsic to a family of motifs corresponding to a specific class of TFs.

The *leucine zipper* signature (Figure 2.4a) under the superclass of basic-domain is an important feature of many eukaryotic regulatory proteins. The hallmark of leucine zipper proteins is the presence of leucine at every 7th position in a stretch of 35 residues. This regularity suggests the presence of a zipper-like α-helical coiled coil bringing together a pair of DNA-binding modules to bind two adjacent DNA sequences. Leucine zippers can couple identical or nonidentical chains, suggesting a homodimeric or heterodimeric signature in the recognition site. A variation of this structural theme often seen in prokaryotic transcription factors is the *helix-loop-helix* signature. In this case, the basic DNA-binding helices are connected into a dimer by a short loop.

The zinc finger domain (Figure 2.4b) is also common in eukaryotic TFs and regulates gene expression by binding to extended DNA sequences. A zinc finger grips a specific region of DNA, binds to the major groove of DNA and wraps part of the way around the double helix. Each finger

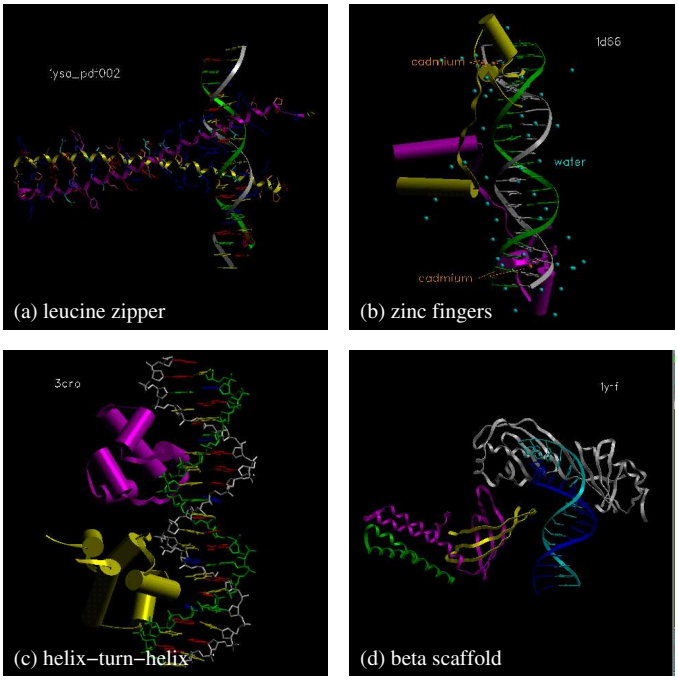


Figure 2.4: DNA binding domains in TFs.

makes contact with a short stretch of the DNA, and residues from the amino-terminal part of the α-helix form hydrogen bonds with the exposed bases in the major groove. Zinc-finger DNA binding proteins are highly versatile and can have various numbers of zinc fingers in the binding domain. Arrays of zinc fingers are well suited for combinatorial recognition of DNA sequences.

The helix-turn-helix domain (Figure 2.4c) contains two α-helices separated by 34 Å - the pitch of a DNA double helix. Molecular modeling studies showed that these two helices would fit into two successive major grooves. This domain, common in bacterial DNA-binding proteins, such as the bacteriophage λ Cro protein, also occurs in the eukaryotic homeobox proteins controlling development in insects and vertebrates.

The beta-scaffold factors (Figure 2.4d) are somewhat unusual in that they bind to the minor groove of DNA. The binding domain is globular rather than elongated, suggesting extensive contact between the DNA sequence and the protein binding domain.

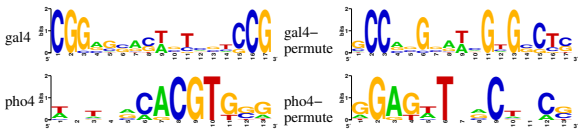


Figure 2.5: Conservation-coupling of a zinc-finger motif *gal4* and a helix-loop-helix motif *pho4*. Since typical conservation-couplings are often reflected in the "contour shape" (e.g., *U- or bell-shape*) of the motif *logo* (a graphical display of the *spatial* pattern of information content over all sites), we can understand this property as a "shape bias".

These class-specific protein-binding mechanisms suggest the existence of features that are characteristic of different families of DNA motifs, and shared by different motifs in the same family. It is evident that the positions within the motifs are not necessarily uniformly conserved, nor are the conserved positions randomly distributed. Since only a subset of the positions inside the motif are directly involved in protein binding, the degree of conservation of positions inside the motif is likely to be spatially dependent, and such dependencies may be typical for each motif family corresponding to a TF class due to structural complementarity between motifs and the corresponding TFs. It is also possible that due to different degrees of variability-tolerance for different TF classes, each family of motifs may require a different selection of prototypes for the distributions of possible nucleotides at the positions within the motifs. Note that such regularities are less likely to be preserved in a non-functional recurring pattern, thus they also provide important clues to distinguishing genuine from false motif patterns during *de novo* motif finding. Figure 2.5 provides two examples for the so-called *conservation-coupling* property of the position dependencies in functional motifs. On the left-hand side are two genuine motifs from two different families. On the right are artificial patterns resulting from a column permutation of the original motifs. Although the two patterns will receive the same likelihood score under conventional PWM representations, clearly the patterns on the left are biologically more plausible because of the complementarity of their patterns of conserved positions to the structures of their binding proteins. Again, it is important to remember that the conservation-coupling property and nt-distribution prototypes are only associated with the generic biophysical properties of a motif family, but **not** with any specific consensus sequence of a single motif; thus, they are called *meta-sequence features*.

2.4.2 HMDM: a Bayesian Profile Model for Motif Families

The goal is to build a statistical model to capture the generic properties of a motif family so that it can generalize to novel motifs belonging to the same family. In the following we develop such a model using a hierarchical Bayesian approach.

The column of nucleotides at each position in a motif can be modeled by a *position specific multinomial distribution* (PSMD). A multinomial distribution over K symbols can be viewed a point in a regular $(K-1)$-dimensional simplex; the probabilities of the symbols are the distances from the point to the faces of the simplex (an example of a 2-dimensional simplex is shown in Figure 2.6a). A Dirichlet distribution is a particular type of distribution over the simplex, hence a distribution over the multinomial distributions. Each specific Dirichlet is characterized by a vector of K parameters. It can impose a bias toward a particular type of PSMD in terms of how strongly it is conserved, and to what nucleotide it is conserved. For example, in Figure 2.6a, the center of probability mass is near the center of the simplex, meaning that the multinomial distributions that define a near uniform probability for all possible nucleotides will have a higher prior probability. But for a Dirichlet density whose center of mass is close to a corner associated with a particular nucleotide, say, "A" (Figure 2.6b), the multinomial distributions with high frequencies for "A" have high prior probabilities. Therefore, we can regard a Dirichlet distribution as a "prototype" for the PSMDs of motifs.

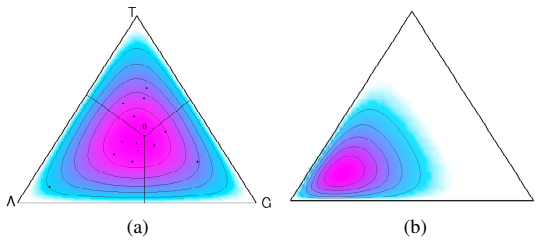


Figure 2.6: Dirichlet densities over a three-nucleotide simplex.

We propose a generative model that generates a multi-alignment $\mathbf{A}$ containing M instances of a motif of length L, in the following way (as illustrated in Figure 2.7). 1) Sample a sequence of states

$s = (s_1, \ldots, s_L)$ from a first-order Markov chain with initial distribution υ and transition matrix Υ. The states in this sequence can be viewed as prototype indicators for the columns (positions) of the motif. Associated with each state, is a corresponding Dirichlet distribution specified by the value of the state. For example, if $s_l = i$, then column l is associated with a Dirichlet distribution $\alpha_i = [\alpha_{i,1}, \ldots, \alpha_{i,4}]^t$. 2) For each $l \in \{1, \ldots, L\}$, sample a multinomial distribution θ_l according to $p(\theta|\alpha_{s_l})$, the probability defined by the Dirichlet component α_{s_l}. 3) Generate all the nucleotides in column l *iid* according to the multinomial distribution parametrized by θ_l.

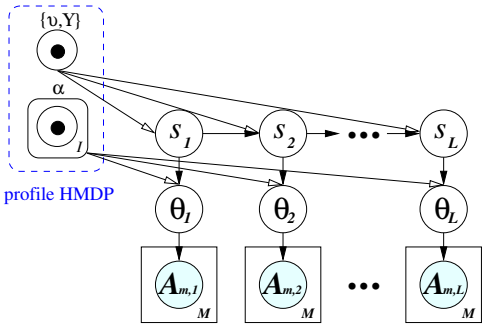


Figure 2.7: The graphical model representation of a MotifPrototyper. Empty circles represent random variables associated with a single motif and the boxes are plates representing *iid* replicates (i.e., M observed instances of the motif). Black arrows denote dependencies between the variables. Parameters of the MotifPrototyper are represented by the center-dotted circles, and the round-cornered box over the α parameter denotes I sets of Dirichlet parameters. The round-cornered dashed box denotes plate of parameters of a single HMDM model, and hence represent a possible mixture of HMDMs.

Thus, the complete likelihood of a motif alignment $\mathbf{A}_{M\times L}$ characterized by a nucleotide-count matrix h is:

$$p(\mathbf{A}, s, \theta|\alpha, \upsilon, \Upsilon) = p(\mathbf{A}|\theta)p(\theta|s, \alpha)p(s|\upsilon, \Upsilon). \tag{2.17}$$

where (using the update properties of the Dirichlet distribution and letting $s_l^i = 1$ if s_l is at state i and 0 otherwise)

$$p(h|x, \theta)p(\theta|s, \alpha) = \prod_{l=1}^{L}\prod_{i=1}^{I} \mathrm{Dir}(\alpha_i + h_l)^{s_l^i}, \tag{2.18}$$

$$p(s|\upsilon, \Upsilon) = \prod_{i=1}^{I}[\upsilon_i]^{s_1^i} \prod_{l=1}^{L-1}\prod_{i,j=1}^{I} [\Upsilon_{i,j}]^{s_l^i s_{l+1}^j}. \tag{2.19}$$

Technically, such a model, which is named a *MotifPrototyper*, is a *hidden Markov Dirichlet-multinomial model*. It defines a structured prior for the PWM of a motif.

With the availability of a categorization for motifs, each family of motifs can be associated with a family-specific profile HMDM model that imposes PSMD prototypes and positional-dependencies unique to this family.

What do we gain from a MotifPrototyper? First, a MotifPrototyper introduces prior information about the joint distribution of the nt-distribution in different positions of a motif of the corresponding family, and gives high probabilities to those commonly found distributions possibly compatible with the degree of variability-tolerance intrinsic to the class of TFs corresponding to the motif family. Under a MotifPrototyper, *a posteriori*, each PSMD in a motif follows a family-specific mixture of multiple Dirichlet distributions, which blends the different prototypes that might dictate the nt-distribution at that position. Furthermore, a MotifPrototyper stochastically imposes family-specific spatial dependencies for different columns within a motif. As Figure 2.7 makes clear, a MotifPrototyper is *not* a simple HMM for sequence data. In an HMM model the transitions would be between the emission models (i.e., multinomials) themselves, and the output at each step would be a single monomer in the sequence. In MotifPrototyper, the transitions are between different prior components for the emission models, and the direct output of this HMM is the parameter vector of a generative model, which will be sampled multiple times at each position to generate *iid* instances. This approach is especially useful when we have prior knowledge about motif properties, such as *conservation-coupling* or other positional dependencies. In contract to the IC profile used in the constrained PM model, due to the stochastic nature of a probabilistic model, MotifPrototyper will in general not be rigidly confined to any particular motif shape (unless we explicitly forbid certain transitions in the transition matrix Υ of the hidden Markov chain). These properties relieve our motif model from the restricted, often brittle constraints needed in other models, such as exactly what shape to look for, the widths of the conserved and unserved patches in a motif, the length of the whole motif [1], etc., and as a result provide desirable flexibility and robustness under practical

[1] In an HMDM model, the length of the motif pattern to be modeled does not have to be rigorously defined but only

motif detection environment.

Secondly, rather than using a maximum likelihood (ML) approach to estimate the PWM, which considers only the relative frequency of nucleotides but is indifferent to the actual number of instances observed, MotifPrototyper facilitates a Bayesian estimation of the PWM under a family-specific prior, thus taking into consideration the actual number of observations available for PWM estimation along with the biological prior. It is possible with only a few instances to obtain a robust estimation of the nucleotide frequency at each position of a motif.

Note that a MotifPrototyper defines a family-specific structured prior for the PWMs without committing to any specific consensus motif sequence.

2.4.2.1 Training a MotifPrototyper

Given biologically identified instances of motifs of a particular family, we can compile a multiple-alignment for each motif and write down the joint likelihood of the training data under a single profile model (i.e., a MotifPrototyper) by marginalizing out the PWMs (i.e., θ's) and the hidden Markov states (i.e., s) of each motif in Eq. (2.17). This likelihood is a function of the model parameters. Thus we can compute the empirical Bayes estimation of the model parameters, $\Theta_l := \{\alpha, \upsilon, \Upsilon\}$, by maximizing the likelihood over each parameter using a EM algorithm with a quasi-Newton procedure for the parameter update step [Sjölander *et al.*, 1996] (see Appendix A.2 for details). The result is a set of parameters intrinsic to the training data.

Note that this training process also involves a model selection issue of how many Dirichlet components should be used. As in any statistical model, a balance must be struck between the complexity of the model and the data available to estimate the parameters of the model. Empirically, we found that 8 components appears to be a robust choice and also provides good interpretability.

needs a rough specification (e.g., a length, say, L =20 bp, that is unlikely to be exceeded by most of the plausible motifs). Because under the HMDM model, both conserved and heterogeneous sites are allowed in a candidate motif pattern; a motif whose length is smaller than L can still be picked up by the model with a over-specified length with high probability by allowing heterogeneous sites padded at the ends of the true motif pattern to make up the total length.

2.4.3 Mixture of MotifPrototypers

Now we have built a model that captures the meta-sequence features of structurally but not textually related motifs. The model is a Bayesian profile model that is defined on each motif family rather than each individual motif; thus we call it a MotifPrototyper. To estimate the PWM of a novel motif, since we do not know which family-specific MotifPrototyper is corresponds best to the novel pattern, we can assume that the motifs are generated from a weighted combination of several MotifPrototypers. Statistically, this defines a mixture of MotifPrototypers as the prior distribution of the PWM of a motif,

$$p(\theta|\{\alpha, \upsilon, \Upsilon\}_k, k = 1, \ldots, K) = \sum_{k=1}^{K} w_k p(\theta|\{\alpha, \upsilon, \Upsilon\}_k), \tag{2.20}$$

where w_k is the mixing weight of each family-specific MotifPrototyper.

Under this setting, one can perform several important probabilistic computations regarding motif detection, such as classifying motifs in terms of their preferred binding protein family by identifying the most likely MotifPrototyper for a given motif alignment; computing the Bayesian estimations of the motif parameters; and biasing the *de novo* motif detection to solutions that are structurally more consistent with biologically genuine motifs. In other words, we effectively turn the originally unsupervised *de novo* motif detection into a semi-unsupervised learning problem that integrates the observed sequences with prior knowledge about motif structures.

2.4.3.1 Classifying motifs

Identifying that a motif belongs to a family, and relating it to other members of the family, often allows inference about its functions. Given multiple profile models each corresponding to a distinct motif family, we can compute the conditional likelihood of a set of aligned instances of an unlabeled motif under each profile model by integrating out the hidden variables (i.e., θ and s) in each resulting complete likelihood function. The posterior probability of each possible assignment of class membership to the motif under test is proportional to the magnitude of the conditional likelihood multiplied by the prior probabilities of the respective motif families (which can be computed from the empirical frequency of each motif family). Letting Z denote the family membership indicator,

the posterior probability of $Z = k$ is proportional to the magnitude of the conditional likelihood under the kth MotifPrototyper multiplied by the prior probability of $Z = k$:

$$p(Z = k|\mathbf{A}) \quad \propto \quad p(Z = k)p(\mathbf{A}|\{\alpha, \upsilon, \Upsilon\}_k)$$

Thus, we can estimate the family membership by a *maximum a posteriori* (MAP) scheme. It is noteworthy that, here, we are classifying a set of aligned instances of a motif as a whole, rather than a single sequence substring as in a standard classification task, such as predicting the function or structure of a protein based on its amino acid sequence [Karchin *et al.*, 2002; Moriyama and Kim, 2003].

2.4.3.2 Bayesian estimation of PWMs

Given a set of aligned instances of a motif, if we know the family membership of this motif, we can directly compute the posterior distribution of its PWM, using the family-specific MotifPrototyper as a prior according to Bayes rule. The Bayesian estimate of a PWM is defined as the expectation of the PWM w.r.t. this posterior.

If the family membership is not known *a priori* (i.e., we do not pre-specify what family of motif to look for, but allow the motif to come from any family), then we can simply assume that the PWM admits a mixture of profile models. The posterior distribution of a PWM under a mixture prior is only slightly more complex:

$$\begin{aligned} p(\theta|\mathbf{A}, \{\alpha, \upsilon, \Upsilon\}_{k=1}^K) &= \sum_k p(\theta|\mathbf{A}, \{\alpha, \upsilon, \Upsilon\}_k, Z = k)p(Z = k|\mathbf{A}, \{\alpha, \upsilon, \Upsilon\}_{k=1}^K) \\ &\propto \sum_k p(\theta|\mathbf{A}, \{\alpha, \upsilon, \Upsilon\}_k)p(\mathbf{A}|\{\alpha, \upsilon, \Upsilon\}_k)p(Z = k), \end{aligned} \tag{2.21}$$

where Z denotes the family membership indicator. A useful variant of this mixture model is to replace the mixture with the maximal-likelihood component:

$$p(\theta|\mathbf{A}, \{\alpha, \upsilon, \Upsilon\}_{k=1}^K) \quad \equiv \quad p(\theta|\mathbf{A}, \{\alpha, \upsilon, \Upsilon\}_{k^*}), \text{ where } k^* = \arg\max_k \; p(\mathbf{A}|\{\alpha, \upsilon, \Upsilon\}_k) \tag{2.22}$$

It is straightforward to generalize the current formulation of the MotifPrototyper model to family-specific prior distributions over more sophisticated motif representations, such as trees or

mixture of trees, by slightly reparameterizing the MotifPrototyper model. The training procedure and the usage for classification and *de novo* motif detection require little modification.

2.4.3.3 Semi-unsupervised *de novo* motif detection

In *de novo* motif detection where locations of motif instances are not known, the motif matrix **A** is an unobserved random variable. One can iterate between predicting motif locations based on the current Bayesian estimate of the motif PWM, and updating the Bayesian estimate based on newly predicted motif instances. We will elaborate on this point in §2.5, where we describe a **LOGOS** model that uses MotifPrototyper as the local model and an expressive hidden Markov model to be developed in the next section, CisModuler, as the global model, for *de novo* motif detection in higher eukaryotic genomic sequences. But as a "prove-of-concept" demonstration of the influence of MotifPrototyper on the performance of *de novo* motif detection, in this section, we only use a simple oops model as the global model. It can be proved that the iterative procedure we described is guaranteed to converge to a locally optimal solution (cf. Chapter 4). But unlike the standard EM algorithm for estimating a PWM, since we can compute the Bayesian estimate based on a trained profile motif prior, we essentially turn *de novo* motif detection from an originally unsupervised learning problem into a semi-unsupervised learning problem that can make use of biological training data without committing to any particular consensus motif pattern.

2.4.4 Experiments

Under the **LOGOS** framework, the MotifPrototyper and mixture of MotifPrototypers are both structured Bayesian upgrades of the standard PM local models for motifs. These models can be learned from categorized training motifs to define family-specific priors for the PWMs, and when coupled with a global model, can be used to introduce useful bias during *de novo* motif detection.

In this sub-section, we present results of learning MotifPrototyper models from categorized families of motifs, and demonstrate applications of the learned MotifPrototypers with three experiments, each addressing a typical issue of interest in *in silico* motif analysis. (1) Given instances of a (computationally) identified motif, assign the motif to a motif family that corresponds to a

particular class of transcription factors. (2) Provide a Bayesian estimate of a PWM which may be more informative than a maximum likelihood estimate. (3) Improve *de novo* motif detection by casting the problem as a *semi-supervised learning* task that makes use of biological prior knowledge incorporated in the family-specific MotifPrototypers.

2.4.4.1 Parameter estimation

The TRANSFAC database (version r6.0) contains 336 nucleotide-count matrices of aligned motif sequences. These matrices summarize a significant portion of the biologically identified transcription regulatory motifs reported in the literature, and are well categorized and curated. (Although the original aligned sequences corresponding to the count matrices are not provided.) We used 271 of the matrices as training data, each derived from at least 10 recognition sites of a TF in one of the 4 well-represented superclasses (Table 2.2), to compute the empirical Bayes estimates of the parameters of 4 profile Bayesian models of motif families.

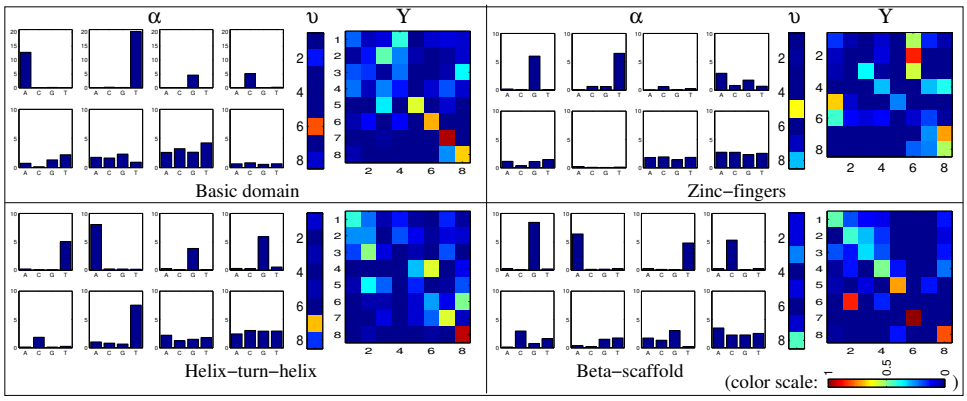


Figure 2.8: Parameters of 4 profile models learned from training motifs. Each of the 8 panels under α represents the 4-dimensional parameter vector of a Dirichlet component (the height of the bar represents the magnitude of the corresponding element in the vector); vector υ and matrix Υ are represented by color images, of which each element of υ or Υ specifies the color of a rectilinear patch in the image.

We performed 50 random restarts for the quasi-Newton algorithm for parameter estimation and picked the solutions corresponding to the highest log likelihood achieved at convergence. Figure 2.8 illustrates the parameters of the 4 resulting profile models pictorially. Here we do not intend to fully

interpret these numerical representations of each profile model in terms of their biological implications. But based on a rough inspection, it is not difficult to read off some interesting high-level biological characteristics. For example, for the *basic-domain* profile model, the transition probabilities between the 4 conserved nt-distribution prototypes [2] (the first four mixture components of the Dirichlet mixture) appear to be rather high (evident from the bright diagonal block at the upper left corner of the Υ matrix), as are the self-transition probabilities of all of the 4 non-conserved Dirichlet components (evident from the bright diagonal stripe at the lower right corner of the Υ matrix). The transition probabilities between the conserved and non-conserved Dirichlet components are relatively low (dark off-diagonal areas in Υ). Furthermore, it appears that the initial probability is high for the 6th Dirichlet component, a fairly non-conserved one. This suggests a general meta-sequence feature, namely that motifs of the basic-domain family are likely to begin with a consecutive run of mostly non-conserved positions, followed by a consecutive stretch of mostly conserved positions, and possibly followed by another consecutive run of mostly non-conserved positions, reminiscent of the bell-shaped signature in Figure 2.5. Although it is possible to find many other similar high-level characteristics, some of which may even reveal previously unnoticed biological features (*e.g.*, characteristic PSMD prototypes of motif families), here we refrain from such elaborations, but simply maintain that MotifPrototyper is a formal mathematical abstraction of the meta-sequence properties intrinsic to a motif profile represented by the training examples.

To evaluate the training quality of the profile models, we define the *training error* as the percentage of misclassification of the superclass-identities of the training motif matrices using profile models learned from the full training set. As Table 2.3 shows, our training errors range from 10-28%, with the beta-scaffold MotifPrototyper having the best fit. Given that "motif family" is rather loosely defined based on TF superclasses, and that each superclass still has very diverse and ambiguous internal structures, these training errors indicate that family-specific regularities can be captured

[2] Note that the parameter vector of a Dirichlet component can be regarded as a vector of pseudo-counts of the nucleotides. Thus a Dirichlet parameter vector with a dominant element implies a conserved nt-distribution prototype, whereas a Dirichlet parameter vector without a dominant elements implies a heterogeneous, or non-conserved nt-distribution prototype.

reasonably well by MotifPrototyper.

Table 2.3: Learning MotifPrototyper

	Basic domains	Zinc-fingers	Helix-turn-helix	beta-scaffold
training error	0.168	0.173	0.276	0.100

2.4.4.2 Motif classification

To examine the generalizability of MotifPrototyper to newly encountered motif patterns, we performed a 10-fold cross-validation (CV) test for motif classification, in which the profile models are learned from 90% of the training motif matrices, and their classification performance is evaluated on the remaining 10% of the motif matrices. We do so 10 times so that each motif pattern corresponding to a particular TF will be classified exactly once as a test case. The performances over each family of motifs are summarized in Table 2.4. Classification error rates for both the entire dataset and the reduced dataset that contains only the major motif subclasses (i.e., those with at least 10 different motifs) under each superclass are presented. Not surprisingly, performance on the dataset with only major subclasses is significantly better, suggesting that the minor classes in each superclass are possibly more ambiguous and less typical with respect to the overall characteristics of the superclass. In fact, some minor classes are unanimously assigned to a different superclass by our classifier, for example, all 6 members of class 1.6 (bHSH) and all 7 members of class 3.4 (heat shock factors) are assigned to superclass 4 (beta-scaffold), whereas all 5 members of class 4.7 (HMG) are assigned to superclass 3 (helix-turn-helix). Whether such inconsistencies reflect a deficiency of our classifier or possible true biological ambiguity of these motif patterns is an interesting problem to be investigated further.

Table 2.4: Motif classification using MotifPrototyper

	Basic domains	Zinc-fingers	Helix-turn-helix	Beta-scaffold
CV error (whole set)	0.256	0.423	0.443	0.403
CV error (major classes)	0.217	0.373	0.379	0.178

To our knowledge, there has been no algorithm that classifies aligned sets of motif instances as

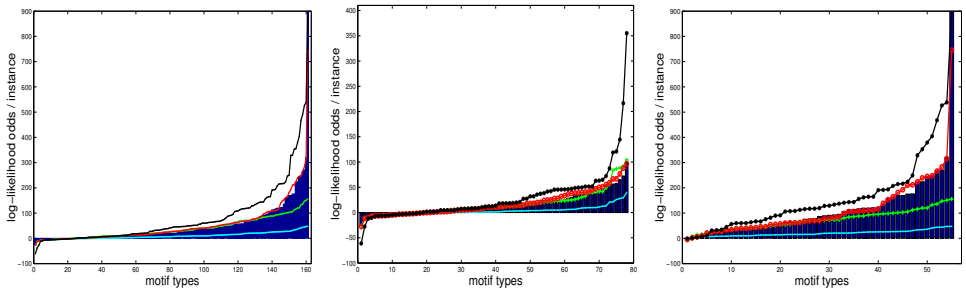
collective objects based on meta-sequence features shared within motif families. The closest counterpart in sequence analysis is the profile HMM (pHMM) model for protein classification [Krogh *et al.*, 1994], but pHMM is based on the assumption that proteins of the same family share sequence-level similarities, and the objects classified are single sequences. Thus, no direct comparison can be made between pHMM and MotifPrototyper. Nevertheless, note that although pHMM is based on much more stringent features at the sequence level and aimed at the relatively simpler task of evaluating single sequences, the typical accuracy of pHMM is around 20-50% for short polypeptides (i.e., $<$ 100 aa) [Karchin *et al.*, 2002; Moriyama and Kim, 2003], comparable to the performance of motif classification using MotifPrototyper. Thus we believe that MotifPrototyper exhibits a reasonable performance given that the labeling of motif family membership is more ambiguous than that of single protein sequences, the meta-sequence features we use are far less stringent than sequence similarities, and motif patterns are much shorter than polypeptides.

2.4.4.3 PWM estimation and motif scoring

A major application of MotifPrototyper is to serve as an informative prior for Bayesian estimation of the PWM from a set of aligned instances of a novel motif. Since in a realistic *de novo* motif detection scenario, one has to evaluate many substrings corresponding to either a true motif, or random patterns in the background, it is expected that the Bayesian estimate of a PWM resulting from MotifPrototyper is more reliable than the maximum likelihood estimate in discriminating between true motifs and background sequences. We demonstrate this ability by comparing the likelihood of a true motif substring with the likelihoods of background substrings, all scored under the estimated PWM of the motif. To get an objective evaluation for this comparison, the following experiments were performed: 1) for a set of aligned instances of a motif, compute the Bayesian estimate of the PWM from 66% of the instances, and then use it to score (*i.e.*, compute the likelihood of) the remaining 34% of the instances in terms of their joint log likelihood; 2) use the same PWM to score M sets of background strings, each having the same length and number of instances as the motif instances being scored in step 1; 3) compute the mean log-likelihood-odds between the motif and

the background substrings (over M sets of randomly sampled background substrings). For each motif, we repeat this procedure 3 times so that each motif substring will be scored exactly once. The performance on each motif is summarized by the average log-likelihood-odds per motif instance. (Larger odds means that the background substrings are less likely to be mistakenly accepted as motif instances, and thence, the false positive rate is smaller).

Since the original aligned motif sequences corresponding to the count matrices used for MotifPrototyper training are not provided in TRANSFAC and are hard to retrieve from the original literature, we compiled an independent collection of aligned motif instances for 161 TFs in TRANSFAC, each of which has at least 6 binding sites whose sequence information is available (Table 2.2). Background substrings from a uniform and random model were simulated [3].



all motifs | motifs with > 10 instances | motifs with <10 instances

Figure 2.9: Evaluation of PWM estimation by 4 different schemes. Cyan:symmetric-Dirichlet smoothing; green: ML; red: mixture of profile models; black: maximal-likelihood profile model out of the mixture. Motifs are listed along the x-axis, ordered by the log-likelihood-odds of their PWM based on the "true" (according to their original family label) profile prior model.

The results of the evaluation are highlighted in Figure 2.9. We compared 4 PWM estimation schemes: maximum likelihood estimation (*i.e.*, plain relative frequencies); Bayesian smoothing using a single symmetric Dirichlet prior; Bayesian estimation using a mixture of profile models; and Bayesian estimation using the maximal-likelihood profile model from the mixture. Depicted as the bars in Figure 2.9 for reference are the results for Bayesian estimation using a single profile model corresponding to the original family label of each motif, an unrealistic scenario in *de novo*

[3]This corresponds to examining the log-likelihood-odds under a motif model w.r.t. a uniform and random null hypothesis. Sampling of background substrings from a genuine genomic sequence as the null hypothesis was also done at a small scale (for some motifs) and yields largely the same results. But since the motifs we studied are from diverse genomic sources, a comprehensive evaluation in this manner is tedious and hence was omitted.

motif detection.

As evident from Figure 2.9 and 2.10, the discriminative power of the Bayesian estimate of the PWM, measured by the log-likelihood-odds (of motif vs. background substrings), is indeed better than that of the maximum likelihood estimate for most of the motifs we tested. In particular, in cases where only a small number of instances are available for estimation, the mixture of profile models still leads to a good estimate that generalizes well to new instances and results in high log-likelihood-odds, whereas the ML estimation does not generalize as well (Fig. 2.9).

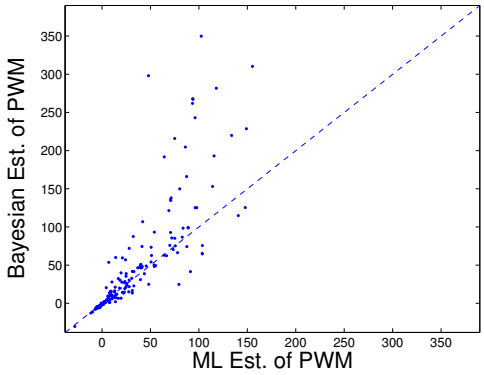


Figure 2.10: A comparison of Bayesian and ML estimates of the PWM. Each point represents a motif being tested, the x-coordinate (resp. y-coordinate) represents the log-likelihood-odds due to the ML (resp. Bayesian) estimation.

These results give strong support to the claim that in many cases, a MotifPrototyper-based approach can significantly improve the sensitivity and specificity for novel motifs, and provide a robust estimation of their PWMs under few observations. These are very useful properties for *de novo* motif detection in complex genomic sequences.

2.4.4.4 *De novo* motif discovery

Now we present a comparison of the profile Bayesian motif model – MotifPrototyper – with the conventional PM model for *de novo* motif detection, using semi-realistic test data for which the ground truth (i.e., full annotation of motif types and locations) is known for evaluating the prediction results. Note that the experiments described here are a small excerpt from a large suite of *de novo* motif detection experiments under various scenarios. We will return to the bulk of these experiments

in §2.7, after the development of a more powerful global model.

We tested on 28 well-represented yeast motifs from the *Promoter Database of Saccharomyces cerevisiae* (SCPD). Each motif has 5 to 32 recorded instances, all of which have been identified/verified via biological experiments and hence are considered "authentic". For each motif, a test dataset is created by planting each of the "authentic" instances of that motif at a random position in a 500bp simulated background sequence (i.e., one motif per sequence). To further increase the difficulty of the motif detection task, a "decoy" signal, which is an artificial pattern obtained by randomly permuting the positions in the motif, was inserted into the study sequence [4]. Since each sequence has only one true motif occurrence, prediction was made by finding the position with the maximal log-likelihood ratio (for the substring that begins with that position) under the estimated motif PWM (obtained at the convergence point of a procedure that iterates between computing the posterior distribution of motif locations based on current estimate of the PWM, and computing the Bayesian estimate of the PWM based on the current posterior distribution of motif locations), and under the background nt-distribution (assumed to be the nt-frequencies estimated from the entire sequence). This scenario frees us from modeling the global distribution of motif occurrences, as needed for more complex sequences (cf. the LOGOS model), and therefore demonstrates the influence of different models for motif patterns on *de novo* detection. We evaluate the performance based on *hit-rate*, the ratio of correctly identified motif instances (within $\pm$ 3bp offset with respect to the locations of the authentic instances) to the total number of instances to be identified. To obtain robust estimation, for each motif 40 experiments were performed, each with a different test dataset (i.e., with different background sequences, motif and decoy locations, and decoy patterns).

Specificity of a single MotifPrototyper. Before presenting the full-scale test of the mixture of MotifPrototypers trained on four categories of motifs from the TRANSFAC database on the yeast motifs from the SCPD database, here we first examine whether the motif properties captured in a MotifPrototyper effectively bias the posterior prediction of motif presence toward the desired

[4] By permutation we mean that the same permuted order is applied to all the instances of a motif so that the multinomial distribution of each position is not changed but their order is changed.

pattern represented in the training set. For this purpose we trained a MotifPrototyper from the 28 yeast motifs in the SCPD database described above, and examined the resulting MotifPrototyper for its ability to detect motifs present in this training set in the presence of a "decoy". Figure 2.11 shows the Boxplot (which shows the median, lower quartile, upper quartile, outliers, etc.) of the hit (i.e., finding the genuine motif) and mishit (i.e., finding the decoy) rate of MotifPrototyper on *abf1* and *gal4*. Note the dramatic contrast of the specificity of the MotifPrototyper to true motifs compared to that of the PM model.

It is noteworthy that the MotifPrototyper model actually does not contain any explicit information about the consensus sequences of the training motifs; it merely captures the dependencies between general heterogeneous and homogeneous motif sites whose nucleotide distributions are not fixed, but instead are drawn from specified priors over the space of nucleotide distributions. Thus, the high specificity of MotifPrototyper to a genuine motif pattern under the interference of a false motif pattern suggests its remarkable ability to implicitly capture sensible "motif shapes".

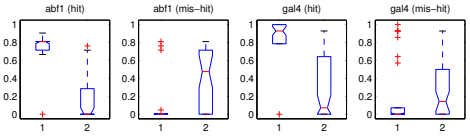


Figure 2.11: Boxplots of hit and mishit rate of MotifPrototyper (1) and PM (2) on two motifs used during MotifPrototyper training.

Generalizability of a single MotifPrototyper. How well does a MotifPrototyper generalize to motifs not present in the training set? Here we use the MotifPrototyper learned from 20 of the 28 SCPD motifs to detect motifs from an independent test set containing the rest of the 8 SCPD motifs. In the first motif finding task, we use synthetic sequences each having only one true motif instance at a random position. Figure 2.12 summarizes the results over 40 experiments. As shown in the figure, the MotifPrototyper significantly outperforms the PM model for motifs *abf1*, *gal4* and *crp*, and achieves comparable performance for motifs *gcn4* and *mig1*. It does poorly for motifs *mat-a2* and *mcb*. Note that these two motifs are quite short and somewhat uniformly "conserved," which is

in fact "atypical" in the training set. The smallish sizes of the motifs also diminish the utility of the Markov model in MotifPrototyper.

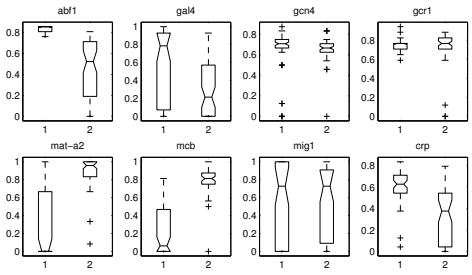


Figure 2.12: Boxplots of hit rate of MotifPrototyper (1) and PM (2) on sequences each embedded with one motif instance from the test dataset.

In the foregoing motif detection task, the PM model shows decent performance, especially for those more-or-less uniformly conserved motifs such as *gcn1*, *mat* and *mcb*. But it already shows signs of failure for motifs with more complex shapes (e.g. *gal4*). The second task is more challenging and biologically more realistic, where we have both the true motifs and the permuted "decoys." Figure 2.13 shows the boxplot of the hit-rate as well as the mishit-rate for motif detection over 40 experiments. As expected, under the interference of the decoys, the PM model apparently gets confused and often decides to pick the permuted false motifs. Only two of the eight motifs are correctly detected by the PM model with high hit-rate. In contrast, the MotifPrototyper model exhibits remarkable robustness under this more difficult situation, and maintains a high hit-rate in six of the eight motifs. But for two of the motifs (again, *mat-a2* and *mcb*), MotifPrototyper biases toward the permuted version, which suggests that indeed the original *mat-a2* and *mcb* patterns are not captured by MotifPrototyper, consistent with the result from the first task.

***De novo* motif detection using a mixture of MotifPrototypers.** Now we conclude this section with an evaluation of a mixture of MotifPrototypers trained from the TRANSFAC database. We test this model on all the 28 motifs from the SCPD database. As shown in Figure 2.14, the mixture of MotifPrototypers significantly outperforms the PM model (i.e. with $> 20\%$ margin) on 11 of the 28 motifs, and is comparable to the PM model (within $\pm 10\%$ difference) for the remaining

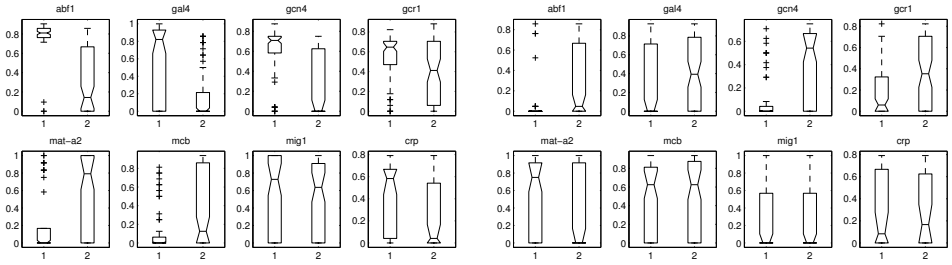


Figure 2.13: Boxplots of hit (left panel) and mishit (right panel) rates of MotifPrototyper (1) and PM (2) on sequences each containing one motif instance from the test dataset together with a permuted decoy.

17 motifs. Overall, the mixture of MotifPrototypers correctly identifies 50% or more of the motif instances for 16 of the 28 motifs, whereas the PM model achieves 50% hit-rate for only 8 of the 28 motifs. Note that the mixture of MotifPrototypers is fully autonomous and requires no user specification of which particular profile motif model to use. If we are willing to introduce a manual post-processing step, in which we use each of the 4 profile motif models described before separately for *de novo* motif finding, and generate 4 sets of motif predictions instead of one (as for the mixture of MotifPrototypers) for visual inspection, it is possible to obtain even better predictions (diamond symbols in Figure 2.14).

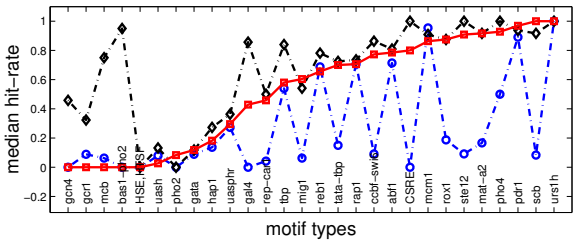


Figure 2.14: Median hit-rates of *de novo* detection of 28 yeast motifs using MotifPrototyper (square), PM (circle), and the best outcome out of 4 single-profile-based predictions using MotifPrototyper (diamond). Motifs are listed along the x-axis, ordered by the hit-rates of MotifPrototyper for each motif.

The ability to provide multiple candidate solutions, each corresponding to a specific TF category, manifests a key advantage of the profile motif model. It allows a user to capture different types of prior knowledge about motif structures and bias motif prediction toward a particular meta-sequence structure in a well-controlled way. A human observer given a visual presentation of the most likely motifs suggested by different profile motif models could easily pick out the best one

from these candidates, whereas the PM model can yield only a single *most likely* answer.

2.4.5 Summary and Discussion

We have presented MotifPrototyper, a novel profile Bayesian motif model that captures generic *meta-sequence features* shared by motifs corresponding to common transcription factor superclasses. It is a probabilistic graphical model that captures the positional dependencies and nucleotide distribution prototypes typical to each motif family, and defines a prior distribution of the position weight matrices of motifs for each family. We demonstrated how MotifPrototyper can be trained from biologically identified motif examples, and its applications for motif classification, Bayesian estimation of PWMs, and *de novo* motif detection.

To the best of our knowledge, all extant motif models are intended to be motif-specific, emphasizing the ability to characterize sequence-level features unique to a particular motif pattern. Thus when one defines such a model for a novel motif not biologically characterized before, one needs to solve a completely unsupervised learning problem to identify the possible instances and fit the motif parameters simultaneously. Under this unsupervised framework, there is little explicit connection between the novel motif to be estimated from the unannotated sequences and the rich collection of biologically identified motifs recorded in various databases. It is reasonable to expect that the fruitful biological investigations of gene regulatory mechanisms and the resulting large number of known motifs could contribute more information to the unraveling of novel motifs. MotifPrototyper represents an initial foray into the development of a new framework that turns *de novo* motif detection into a semi-unsupervised learning problem. It provides more control during the search for novel motif patterns by making use of prior knowledge implied in the known motifs, helps to improve sensitivity to biologically plausible motifs, and potentially reduces spurious solutions often occurred in a purely unsupervised setting.

It may be possible to build a stronger motif classifier using discriminative approaches such as neural networks or support vector machines, and we are currently pursuing this direction. But since the goal of this chapter is not merely to build a classifier, but to develop a model that can easily

be integrated into a more general architecture for *de novo* motif detection, a generative framework, especially via a Bayesian prior model, provides the desired generalizability and flexibility for such tasks. As discussed in §2.2.4, a graphical model formalism of the motif detection problem allows a modular combination of heterogeneous submodels each addressing a particular component of the overall problem. The design of MotifPrototyper aligns with this principle, and serves as an advanced "local" submodel under the **LOGOS** framework.

2.5 CisModuler: Modeling the Syntactic Rules of Motif Organization

As discussed in previous sections, the transcription regulatory sequences in higher eukaryotic genomes often consist of multiple CRMs. Each CRM contains locally enriched occurrences of binding sites for a certain array of regulatory proteins, capable of integrating, amplifying or attenuating multiple regulatory signals via combinatorial interaction with these proteins. The architecture of CRM organization is reminiscent of the grammatical rules underlying a natural language, and provides the potential for implementing sophisticated regulatory circuits directing temporally/spatially coordinated expression of genes during development and differentiation. It also presents a particular challenge to computational motif and CRM identification in higher eukaryotes. In this section, we present CisModuler, a Bayesian hidden Markov model that attempts to capture the stochastic syntactic rules of CRM organization and integrates over (and thus draws influence from) all possible values of the Markov transition probabilities weighted by their corresponding prior probabilities that reflect general knowledge of the CRM structure. Under the CisModuler model, all candidate sites are evaluated based on a posterior probability measure that takes into consideration their similarity to known binding sites, their contrasts against local genomic context, and their first-order dependencies on upstream sequence elements. We compare this approach to the standard window-based likelihood scoring approach described previously, and demonstrate superior results on large scale analysis of *Drosophila* early developmental enhancers. This model provides a useful and arguably superior alternative for CRM/motif detection given motif PWMs, and can be also used as a submodel (i.e., the global model) for *de novo* motif/CRM detection in higher eukaryotic genomes.

2.5.1 The *CisModuler* Hidden Markov Model

Hidden Markov models have been widely used in computational biology to capture simple sequence structures (e.g., segmentations) inherent in bio-polymer sequences. Despite their limited expressive power compared to more complex models such as stochastic context free grammars (SCFGs) [Lari and Young, 1990] or hierarchical hidden Markov models (hHMMs) [Fine *et al.*, 1998], they have enjoyed remarkable success in problems such as gene-finding in DNAs [Burge and Karlin, 1997] and domain modeling in proteins [Krogh *et al.*, 1994], and in many cases appear to strike the right balance between simplicity and expressiveness.

We propose to use an HMM to model the global distribution of motif instances in genomic sequences, by encoding a set of stochastic syntactic rules presumably underlying the CRM organization and motif dependencies using a discrete first-order Markov process. We call this specialized HMM a *CisModuler*. The CisModuler HMM defines a probability distribution over possible functional states of each single position in a DNA sequence. The space of allowed functional states is constructed in a way that captures detailed architectural features of genomic sequences bearing CRMs.

More precisely, let $X = (X_1, \ldots, X_T)$ be a chain of "hidden" state variables associated with an "observed" DNA sequence $y = (y_1, \ldots, y_T)$, specifying which functional state (e.g., a background, the l-th position of motif k, etc.) is responsible for generating the observed nucleotide at each position. By definition, $x_t \in \mathbb{S}$, where the state space $\mathbb{S}$ includes all possible functional states of a position in a CRM-bearing DNA sequence. Specifically, $\mathbb{S} = \mathbb{M} \cup \mathbb{M}' \cup \mathbb{B}_p \cup \mathbb{B}_d \cup \{b_g, b_c\}$, where $\mathbb{M} = \{1^{(1)} \ldots L_1^{(1)}, 1^{(2)} \ldots L_2^{(2)}, \ldots, 1^{(k)} \ldots L_k^{(k)}\}$ is the set of all possible sites within a motif on the forward DNA strand (i.e., states $1^{(1)}$ to $L_1^{(1)}$ correspond to the sites in motif type 1 on the forward strand, and so on); $\mathbb{M}'$ is the set of all possible sites within a motif if it is on the reverse complementary DNA strand; $\mathbb{B}_p = \{b_p^{(1)}, \ldots, b_p^{(k)}\}$ denotes the set of *proximal-buffer* states associated with each type of motif [5]; $\mathbb{B}_d = \{b_d^{(1)}, \ldots, b_d^{(k)}\}$ denotes the set of *distal-buffer* states associated

[5] Here, proximal-buffer refers to the background sites immediately next to the proximal-end of the motif. For consistency, orientations are defined with respect to the initial position of the input sequence. That is, the 1st position of the input sequence corresponds to the proximal end, and the last position corresponds to the distal end.

with each type of motif; b_c represents the intra-cluster (local) background state; and b_g represents the inter-cluster (global) background state. Permissible transitions between these states are illustrated in Figure 2.15. The distribution of X follows a first-order Markov process according to this transition scheme, with state-transition probabilities parameterized as shown in the state-transition diagram. Note that we have not included functional states related to gene annotation and basal promoters, but such extensions are straightforward if co-identification of CRMs and genes is desired.

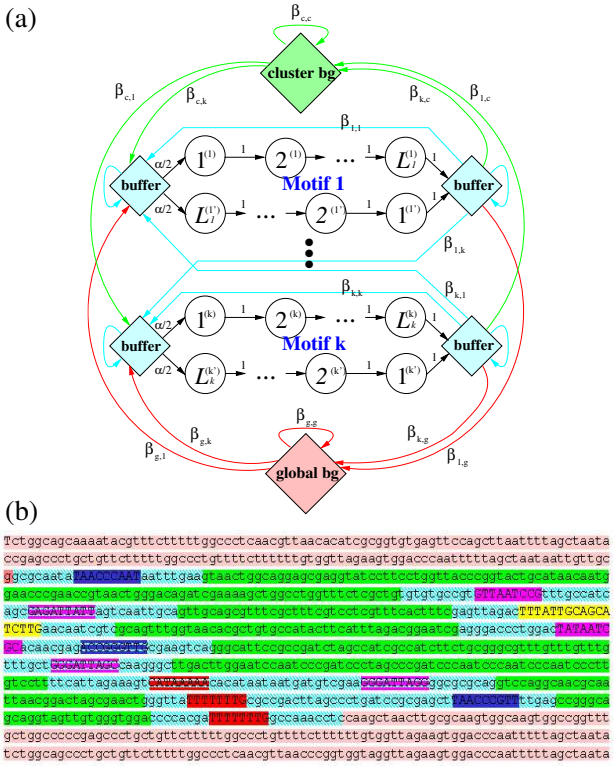


Figure 2.15: The CisModuler HMM. (a) The state-transition diagram. Labeled circular and diamond nodes represent the functional states in DNA sequences; arrows between nodes represent permissible state-transitions; numbers and parameter symbols accompanied the arrows (with the parameter subscripts denoting the source and target of the transitions) denote the corresponding transition probabilities. (b) A typical segmentation of a piece of DNA sequence induced by CisModuler. Background sites are colored as in the state-transition diagram; the blue, magenta, yellow and red segments represent motifs of 4 different kinds; segments with parallel stripes denote motifs in reverse-complementary orientation.

The motivation for this Markov model is that generally one expects to see occasional motif clusters in a large ocean of global background sequences (represented by state b_g). Each motif instance

in a cluster is like an island in a sea of intra-cluster background sequences (b_c's), with surrounding coastal water of motif-specific buffer sequences (i.e., $b_p^{(i)}$'s and $b_d^{(i)}$'s for motif i) (Fig. 2.15). We refer to a motif instance together with its surrounding buffers as a *motif envelope*. The CisModuler model assumes that the distance between clusters is geometrically distributed with mean $1/(1-\beta_{g,g})$, and the span of the intra-cluster sea is also geometrically distributed with mean $1/(1-\beta_{c,c})$. However, the distances between motifs admit a much richer distribution, because the widths of the motif envelopes are modeled on a motif-specific basis, and the transitions between envelopes can occur either by sailing through the intra-cluster sea or by bypassing it. These modeling choices are intended not only to reflect uncertainty about the CRM structure, but also to offer substantial flexibility to accommodate potential richness of CRM structures. As shown in Figure 2.15a and 2.15b, one can begin with a global background state, then either loop over this state, or with some probability $\beta_{g,i}$, move into the proximal-buffer state of a motif i; with equal probability $\alpha_{i,m}/2$, a proximal-buffer state $b_p^{(i)}$ reaches the start states $1^{(i)}$ (resp. $L_i^{(i')}$) of motif i on the forward (resp. reverse) strand, deterministically passes through all internal sites of motif i, and transitions to the distal-buffer state $b_d^{(i)}$, thereby stochastically generating a non-empty motif envelope [6]; each $b_d^{(i)}$ has some probability $\beta_{i,j}/2$ of transitioning to the proximal-buffer state of another motif j (or of the same motif when $j = i$) to concatenate another motif envelope, or with probability $\beta_{i,c}$ to pad with some intra-cluster background before adding more envelopes; all distal-buffer states also have probability $\beta_{i,g}$ of returning to the global background state, terminating a CRM stretch. It is not difficult to see that a path in such a state space according to this HMM grammar bears a structure similar to a genomic sequence containing motif modules (Figure 2.15b). Note that the HMM model does not impose rigid constraints on the number of motif instances or modules; the actual number of instances is determined by the posterior distribution of the sequence of functional states, $p(x|y)$.

The use of an HMM to model the CRM distribution has been previously described by Frith *et al.* in the Cister program [Frith *et al.*, 2001]. But the CisModuler model we present here uses a

[6] Note that the distinction between the proximal and distal buffers avoids generating empty envelopes (because otherwise, a single buffer state would not be able to remember whether a motif has been generated beyond k positions prior to the current position under a kth order Markov model.)

much more sophisticated design of the functional state space that allows couplings between motifs within the CRMs to be captured, and models inter-motif distances with more flexible distributions (rather than a simple geometric distribution). Furthermore, as will be detailed in the following sections, we provide a Bayesian treatment for the state transition probabilities, which in previous models are regarded as fixed parameters and rely on empirical default values or user specification. We also combine the newly designed HMM with a more expressive kth-order Markov model for the background, which turns out to contributes to significantly improving the specificity for CRM detection.

2.5.2 Bayesian HMM

One caveat of the standard HMM approach for CRM modeling is the difficulty of fitting the model parameters, such as the state-transition probabilities, due to the scarcity of fully annotated CRM-bearing genomic sequences. In principle, one can learn the maximal likelihood estimates of the model parameters in an unsupervised fashion, using the Baum-Welch algorithm, directly from the unannotated sequences while analyzing them. But in practice, such a completely likelihood-driven approach tends to result in spurious results, such as over-estimation of the motif and CRM frequencies and poor stringency of the learned models of potential motif patterns. Previous methods tried to overcome this by reducing as much as possible the number of parameters needed, and setting them according to some best guesses of the motif/CRM frequencies or CRM sizes [Frith *et al.*, 2001]. But as a result, such remedies compromise the expressive power of the already simple HMM, and risk misrepresenting the actual CRM structures. In the following, we propose a Bayesian approach that introduces the desired "soft constraints" and smoothing effect for an HMM of rich parameterization, using only a small number of *hyper-parameters*. Essentially, this approach defines a posterior probability distribution over all possible value-assignments for the HMM parameters, given the observed unannotated sequences and empirical prior distributions of the parameters that reflect general knowledge of CRM structures. The resulting model allows probabilistic queries (i.e., estimating the probability of a functional state) to be answered based on the aforementioned posterior distribution

rather than on fixed given values of the HMM parameters.

We assume that the self-transition probability of the global background state $\beta_{g,g}$, and the total probability mass of transitioning into a motif-buffer state $\sum_{k\in\mathbb{B}_p}\beta_{g,k}$ (note that $\beta_{g,g} = 1 - \sum_{k\in\mathbb{B}_p}\beta_{g,k}$), admit a beta distribution, $Beta(\xi_{g,1}, \xi_{g,2})$, where a small value was chosen for $\frac{\xi_{g,2}}{\xi_{g,1}+\xi_{g,2}}$, corresponding to a prior expectation of a low CRM frequency. Similarly, a beta prior $Beta(\xi_{c,1}, \xi_{c,2})$ is defined for the self- and total motif-buffer-going transition probabilities $[\beta_{c,c},\ \sum_{k\in\mathbb{B}_p}\beta_{c,k}]$ associated with the intra-cluster background state; and another beta prior $Beta(\xi_{p,1}, \xi_{p,2})$ for the self- and motif-going transition probabilities $[\alpha_{i,i},\ \alpha_{i,m}]$ associated with the proximal-buffer state of a motif. Finally, it is assumed that, for the distal-buffer state, the self-transition probability, the total mass of transition probabilities into a proximal-buffer state, the probability of transitioning into the intra-cluster background, and the probability of transitioning into the global background, $[\beta_{i,i},\ \sum_{k\in\mathbb{B}_p}\beta_{i,k},\ \beta_{i,c},\ \beta_{i,g}]$, admit a 4-dimensional gamma distribution, $Gamma(\xi_{d,1}, \xi_{d,2}, \xi_{d,3}, \xi_{d,4})$.

Note that due to conjugacy between the prior distributions described above and the corresponding transition probabilities they model, the hyper-parameters of the above prior distributions can be understood as *pseudo-counts* of the corresponding transitioning events, which can be roughly specified according to empirical guesses of the motif and CRM frequencies. But unlike the standard HMM approach, in which the transition probabilities are fixed once specified, the hyper-parameters only lead to a soft enforcement of the empirical syntactic rules of CRM organization in terms of prior distributions, allowing controlled posterior updating of the HMM transition probabilities during analysis of the unannotated sequences. For the CisModuler HMM, we specify the hyperparameters (i.e., the pseudo-counts) using estimated frequencies of the corresponding state-transition events, multiplied by a "prior strength" N, which corresponds to an imaginary "total number of events" from which the estimated frequencies are "derived". That is, for the beta priors, we let $[\xi_{[\cdot,1]}, \xi_{[\cdot,2]}] = [1 - \omega_{[\cdot]},\ \omega_{[\cdot]}] \times N$, where the "$\cdot$" in the subscript denotes either the g, c, or p state, and $\omega_{[\cdot]}$ is the corresponding frequency. For the gamma prior, we let $[\xi_{d,1}, \xi_{d,2}, \xi_{d,3}, \xi_{d,4}] = [\omega_{d,1},\ 1 - \sum_j \omega_{d,j},\ \omega_{d,2},\ \omega_{d,3}] \times N$. Overall, 7 hyper-parameters need to be specified (of course one can use different "strengths" for different prior, with a few additional

parameters), a modest increase compared to the three needed in Cister [Frith *et al.*, 2001].

2.5.3 Markov Background Models

Several previous studies have stressed the importance of using a richer background model for the non-motif sequences [Liu *et al.*, 2001; Huang *et al.*, 2004]. In accordance with these results, CisModuler uses a global kth-order Markov model for the emission probabilities of the global background state. For the emission probabilities of the intra-cluster background state and the motif-buffer states, we used two *local* Markov models of order m and m', respectively. Since the models are defined to be *local*, the conditional probability of a nucleotide at a position t is now estimated from all $(m+1)$- (resp. $(m'+1)$-) tuples from a window of $2d$ (resp. $2d'$) centered at t. These probabilities can also be computed off-line and stored for subsequent use. With a careful bookkeeping scheme (i.e., using a "sliding window" to compute the local Markov model of each successive position, each with a constant "update cost" based on the previous one, except for the initial window that needs a cost quadratic in the window size), this computation takes only $O(T)$ time. For the emission probabilities of the motif states, we directly use the appropriate columns of nucleotide frequencies in the PWM of the corresponding motif.

2.5.4 Posterior Decoding Algorithms for Motif Scan

2.5.4.1 The baseline algorithm

Given the initial state distribution and transition probability matrix of the HMM, the background probabilities of each nucleotide, and the PWMs of the motifs to be searched for, the posterior probability distribution of the functional states at each position of the sequences, $p(x_t|y), \forall t$, can be computed using the forward-backward algorithm. One can read off the functional annotation (or segmentation) of the input sequences from $p(x_t|y)$ according to a *maximal a posteriori* (MAP) scheme, that is, the predicted functional state of position t is:

$$x_t^* = \arg\max_{s \in \mathbb{S}} p(X_t = s|y) \tag{2.23}$$

Note that by using such a posterior decoding scheme (rather than the Viterbi algorithm), one

integrates the contributions of all possible functional state paths for the input sequence (rather than a single "most probable" path), into the posterior probability for each position. Therefore, although the HMM architecture does not explicitly model overlapping motifs, the inference procedure does take into account possible contributions of DNA binding sites interacting with competing TFs.

2.5.4.2 Bayesian inference and learning

Under the Bayesian framework described in § 2.5.2, the parameters in the HMM are treated as continuous random variables (collectively referred to as Ω) with a prior distribution. Now to compute the posterior probability of functional states needed in Eq. (2.23), one needs to marginalize out these parameter variables:

$$p(x_t|y) = \int p(x_t|y,\Omega)p(\Omega|y)d\Omega \tag{2.24}$$

This computation is intractable in closed form. One approach to obtaining an approximate solution is to use Markov chain Monte Carlo methods (e.g., a Gibbs sampling scheme). Here we use a more efficient, deterministic approximation scheme based on *generalized mean field* (GMF) inference, also referred to as *variational Bayesian learning* [Ghahramani and Beal, 2001] in the special scenario that is applicable to our problem setting. We will discuss the theoretical and algorithmic details of GMF inference at length in Chapter 4. Operationally, a posterior decoding algorithm under the Bayesian HMM setting can be understood as replacing the single-round posterior decoding with an iterative procedure consisting of the following two steps:

- Compute the expected counts for all state-transition events (i.e., sufficient statistics) using the forward-background algorithm, using **current** values of the HMM parameters.
- Compute the Bayesian estimate (to be detailed shortly) of the HMM parameters based on their prior distribution and the expected sufficient statistics from the last step. **Update** the HMM parameters with these estimations.

This procedure is different from the standard EM algorithm which alternates between inference about the hidden variables (the E step) and maximal likelihood estimation of the model parameters

(the M step). In a GMF algorithm, the "M" step is a Bayesian estimation step, in which one computes the posterior expectation of the HMM parameters, which will lead to an optimal lower bound on the true likelihood of the data (which is intractable to compute exactly for a Bayesian HMM) (see Chapter 4).

Now we outline the formulas for Bayesian estimation of the HMM parameters. Note that because the state-transition probability distributions (which are multinomial) and the prior distributions of the transition parameters (which are either beta or gamma) are conjugate-exponential [Beal *et al.*, 2001], we have to compute the Bayesian estimate of the logarithms of the transition parameters (referred to as the *natural parameters*) rather than of the parameters themselves. For example, for the state-transition parameter $\beta_{g,g}$, we have:

$$\begin{aligned} E[\ln(\beta_{g,g})] &= \int_{\beta_{g,g}} \ln \beta_{g,g} p(\beta_{g,g}|\xi_{g,1}, \xi_{g,2}, E[n_{g,g}]) d\beta_{g,g} \\ &= \Psi(\xi_{g,1} + E[n_{g,g}]) - \Psi(\sum_j \xi_{g,j} + \sum_{k \in \mathbb{B}_p} E[n_{g,k}]), \end{aligned} \tag{2.25}$$

where $\Psi(x) = \frac{\partial \log \Gamma(x)}{\partial x} = \frac{\dot{\Gamma}(x)}{\Gamma(x)}$ is the digamma function; $E[\cdot]$ denotes the expectation with respect to the posterior distribution of the argument; and $n_{g,g}$ refers to the sufficient statistic of the parameter $\beta_{g,g}$ (i.e., the counts of the transition event $g \to g$). The Bayesian estimate of the original parameter is simply $\beta^*_{g,g} = \exp(E[\ln(\beta_{g,g})])$. (In fact we keep using the natural parameterization in the actual forward-background inference algorithm to avoid numerical underflow caused by a long product of probability terms.) The individual "motif-buffer-going" probability $\beta_{g,i}$ can be estimated similarly.

The initial state probability of the the CisModuler HMM is not important for CRM prediction as it only directly determines the functional state of the first position of the input sequences and its influence diminishes quickly along the sequence. One can simply fix the initial state to be a global background state with probability 1.

2.5.5 Experiments

Although the literature on transcription regulation mechanisms in higher eukaryotes is very rich, there still exist great biological ambiguities in motif and CRM annotations in many metazoan

genomes. To evaluate our model using relatively unambiguous criteria, we focus on 14 loci in the *Drosophila* genome that are well known to be involved in regulating the transcription of *Drosophila* early developmental genes (Table 2.5). Papatsenko *et al.* [2002] have carefully curated these loci based on an extensive study of the literature. Their compilation delineates 19 best-known early *Drosophila* developmental enhancers from these loci, where reside binding sites for 3 maternal transcription factors: Bicoid (Bcd), Caudal (Cad), Dorsal (DI), as well as the zygotic gap gene factors Hunchback (Hb), Kruppel (Kr), Knirps (Kni), Tailless (Tll), and Gaint (Gt). To mimic the typical motif/CRM search scenario in metazoan genomic analysis (i.e., using a single long sequence potentially containing numerous motifs rather than multiple short promoter regions from co-regulated genes of simple organisms such as yeast), for each locus we extract a 5000 to 20000 bp long genomic region surrounding the enhancers as input data. Note that it is possible that there may exist additional unknown motifs/CRMs in these extended regions.

Table 2.5: Developmental regulatory loci in *Drosophila* genome.

locus (target gene)	regulators	length	# of CRMs
Abdominal-A	Hb, Kr, Gt,	10000	1
Buttonhead	Bcd, Hb	5000	1
Engrailed	Cad, Ftz	10000	1
Even-skipped	Hb, Kni, Bcd, Kr, Gt	20000	3
Fushi-Tarazu	Ftz, Ttk, Cad	10000	2
Gooseberry	Eve, Prd (HD)	10000	1
Hairy	Kr, Hb, Kni, Cad, Gt	10000	3
Kruppel	Bcd, Hb, Gt, Kni	10000	1
Orthodentcile	Bcd	5000	1
Runt	Kr, Gt, Hb, Kni	10000	1
Spalt	Bcd, Hb, Kr, Cad	10000	1
Tailless	Bcd, Cad	8227	1
Ultrabithorax BRE	Hb, Ftz, Tll	10000	1
Ultrabithorax PBX	Hb, Ftz, Tll	10000	1

As discussed above, the hyperparameters of the CisModuler model reflect prior beliefs about the architectural features of the CRM structure, such as rough spans of the inter- or intra-module background and distances between motif instances. We specify these hyperparameters as follows: for the global background, $\omega_g = 0.0002$; for the intra-module background, $\omega_c = 0.01$; for the proximal motif buffer, $\omega_p = 0.1$; for the distal buffer hyperparameters, $\omega_{d,1} = 0.1$, $\omega_{d,2} = 0.4$, $\omega_{d,3} = 0.4$; and for the strength of the hyperparameters, $N = 500$. The background probability of the nucleotide at each position was computed locally using a 3rd-order Markov model from a

sliding window of 600 bp centered at the corresponding position. Since we are scanning for known motifs, the PWMs of the motifs to be found are taken from [Papatsenko *et al.*, 2002] and [Berman *et al.*, 2002].

2.5.5.1 MAP prediction of motifs/CRMs

The locations of individual motif instances and CRMs in a DNA sequence can be determined from its associated state sequence that corresponds to the MAP states of all the DNA sites. Figure 2.16a shows the MAP states and the associated posterior probabilities of these states in a 5000 bp region at the *Drosophila* buttonhead locus. As shown in the graphical illustration below the MAP plot, this region contains a CRM between positions 330 and 1504, and part of the coding sequence of the buttonhead gene. Fine-grained annotations indicate that a core subregion at the proximal end of this CRM (positions 447-660) harbors 5 Bcd motifs. Another 4 Bcd instances are clustered at the distal end of this CRM (positions 1150-1354). Three additional motifs (2 Bcds and 1 Hb) are scattered in the middle of this CRM, but they appear to be weaker matches to the motif consensus compared to the ones in the core subregions [Papatsenko *et al.*, 2002]. Using MAP estimation under CisModuler, 7 of the 12 motifs, 3 in the proximal and 4 in distal subregions of the CRM, are identified and the core regions of the buttonhead CRM are correctly identified. Overall, among the 335 motif instances (of 11 different regulatory proteins) and 19 CRMs contained in the loci we analyzed, 80 motifs and 16 CRMs are correctly identified, out of a total prediction of 316 motifs and 51 CRMs.

Under the aforementioned parameterization of CisModuler, the sensitivity measure of our prediction (i.e., correct predictions/total annotated motifs) is about 25%. But it is worth pointing out that this result is obtained at a very low noise-to-signal ratio (i.e., incorrect predictions/correct predictions) of less than 3. Most extant algorithms report a list of predictions ranked by the score and provide no quantitative measure of prediction accuracy suitable for a comparison. A few extant algorithms reported higher sensitivity, but at an extremely high N/S ratio. For example, the log-odds-based MATCH program [Quandt *et al.*, 1995] achieves a $\sim 90\%$ sensitivity with a N/S ratio of

1379 and 784 for Ap-1 and NEAT sites, respectively; the more sophisticated comparative-genomics-based rVista program [Loots *et al.*, 2002] achieves a similar sensitivity with N/S ratios of about 69 and 38, respectively. Such a high N/S ratio can make experimental verification extremely hard or even infeasible, significantly compromising the value of the predictions. Also worth mentioning is that our CRM-prediction using CisModuler is even more reliable, with an 84% sensitivity, and 2.18 N/S. Thus it is possible to first identify the CRMs using a coarser-grained model, and then zoom in to find motifs within the CRMs using a finer-grained model.

Note that unlike many other scoring schemes for motif/CRM detection, such as the log odds or likelihood score regularized by word frequencies, our MAP prediction does not require a cutoff value for the scores, nor a window to measure the local concentration of motif instances, both of which are difficult to set optimally. To show the potential advantage of the MAP approach, Fig 2.16b shows the log odds of all sites in the buttonhead locus. Simply from inspection, it is apparent that, even though we compute the log odds based on a more discriminating Markovian background model together with the motif PWMs, we end up with too many positive signals (i.e., peaks with log odds > 0). Exponentiating the log odds of all sites and thus transforming them to likelihood ratios (the lower small panel in the graph) can significantly improve the contrast, but compared to the MAP plot and the sketch of the genomic structure of this region, pruning away noises via a good cutoff value and scoring window is still a non-trivial task.

2.5.5.2 Motif/CRM prediction via thresholding posterior probability profile

The MAP prediction described in the previous section considers only a single (i.e. the *a posteriori* most probable) functional state for each site in a DNA sequence, and to some degree underuses the posterior probabilities of all possible functional states for each site. An alternative approach is to use the full posterior probability distribution at each site as a score function, and analyze the score profile of the whole sequence using strategies conventionally applied to log odds or likelihood profiles, such as thresholding motif scores with a cutoff value (to qualify a motif instance) and measuring local motif concentrations with a sliding window (to qualify a CRM).

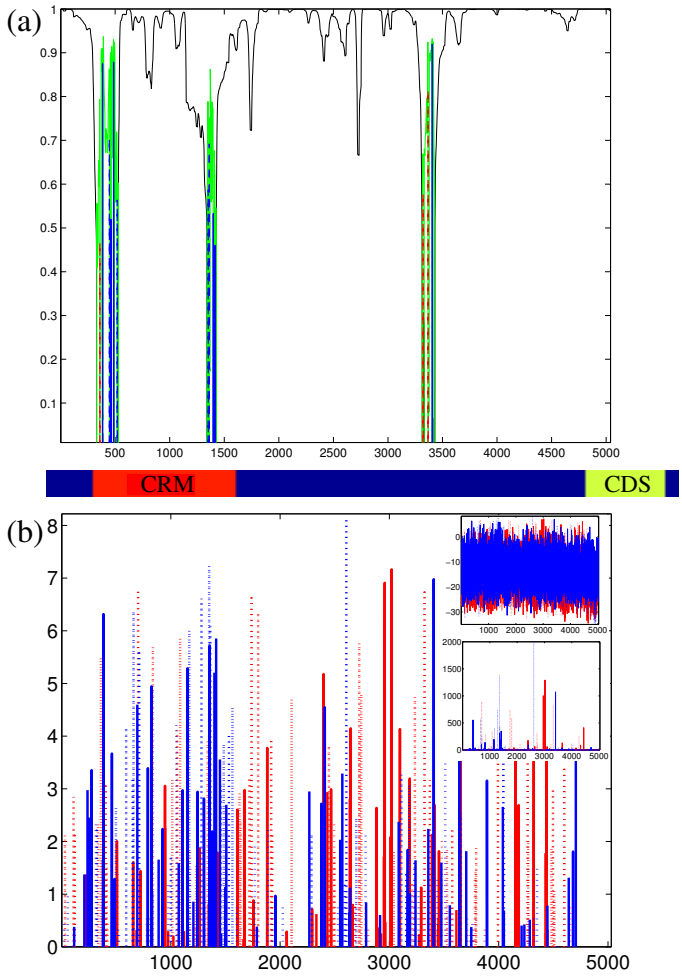


Figure 2.16: Motif- and CRM-scan using CisModuler. (a) MAP plot of the buttonhead locus under the CisModuler model. The y axis denotes posterior probability, and the x axis represents sites in the sequence. The black curve corresponds to the global background state, the green curve corresponds to the intra-cluster background and buffer states, and other color curves correspond to various motif states (red:Hb, blue:Bcd, and dotted curves correspond to the state of a reverse oriented motif represented by the same color). For each site, only the posterior probability of the MAP state is plotted. (b) Log odds of each site under the motif PWM versus a 3rd-order local Markov background model. Only positive scores (i.e., higher motif prob. than background prob.) are shown in the large panel. Complete log odds profiles (including the negative scores that indicate the background) are shown in the upper small panel for reference. The likelihood ratio scores derived from the log odds are shown in the lower small panel. Between panels (a) and (b) is a graphical illustration of the biological annotation of this region.

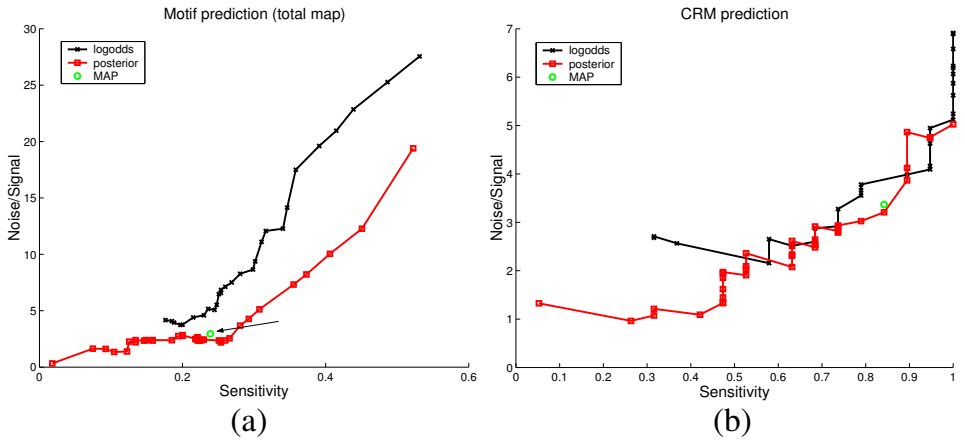


Figure 2.17: Validation of the posterior score under CisModuler and the likelihood ratio score: trade-off between sensitivity and noise. (a) Motif detection. (b) CRM detection. The performance curves record sensitivity and N/S achieved at a wide range of score cutoff values.

Fig. 2.17a shows the trade-off between sensitivity and noise during motif detection, in terms of the proportion of the known binding sites detected and the amount of concomitant noise generated. Following [Huang *et al.*, 2004], Fig. 2.17a traces the balance of sensitivity versus N/S ratio achieved at a wide range of score cutoffs. Two score profiles were analyzed, the posterior probability profile computed using the CisModuler model for the 14 *Drosophila* loci described before (red curve in Fig. 2.17a), and a likelihood ratio profile for the same dataset computed using motif PWMs and a 3rd-order local Markov model (black curve in Fig. 2.17a). Overall, the CisModuler posterior probability score outperforms the likelihood ratio score over the entire range of noise-to-signal ratio. Although not directly comparable (since different datasets are used), the performance curve is similar to that of [Huang *et al.*, 2004] (or arguably better because of the longer input sequences used and the presence of CRMs that complicate motif identifications.) It is interesting to note that the MAP prediction seems to be trying to pick the best possible sensitivity in the low noise-to-signal ratio region.

Fig. 2.17b shows the trade-off between sensitivity and N/S ratio for CRM (rather than motif) detection. The following scheme were used to identify a CRM based on the score profile: under a given cutoff value of motif score, if the motif density within a sliding window of length W is at least c, the corresponding sequence stretch is regarded as covered by a CRM. A contiguous region swept

by a sliding window that meets this criteria is regarded as a CRM. Following typical characteristics of CRMs reported in the literature, we set $W = 500$ and $c = 2\%$. The sensitivity of CRM detection is defined to be the ratio of the number of correctly predicted CRMs to the total number of CRMs; and the N/S ratio as the ratio of the total length of all predicted CRMs over the length of correctly predicted CRMs. From Fig. 2.17b, it appears that CisModuler slightly outperform the the likelihood ratio scheme in the high N/S region, and is significantly better in the low N/S region. As mentioned earlier, the MAP prediction finds a good trade-off between the sensitivity and N/S ratio.

From the experiments reported above, we are optimistic that CisModuler is superior in motif and CRM detection in a complex genomic context. But to predict based on the full posterior probability profile, a cutoff value is needed to qualify the possible presence of motif instances, and a window size will be used to infer CRMs based on within-window concentration. Usually, such values have to be carefully determined from a training dataset, or via a statistical significance criterion as in [Huang *et al.*, 2004]. As a reward, we can take advantage of both the motif dependencies and syntactic architecture of motif distributions explicitly captured in the CisModuler, the flexibility of the thresholding scheme, and the nice statistical guarantee provided by a significance test.

2.5.6 Summary and Discussion

In this section, we presented a model-based Bayesian approach for CRM and motif prediction, which combines many of the desirable features provided by extant methods, and introduces several important novel elements that overcome some of the shortcomings of extant methods. The extensions and contributions includes: a more sophisticated HMM model that is intended to capture, to a reasonable degree, the detailed syntactic structure of CRM and *cis*-regulatory regions containing CRMs; Bayesian priors for various state-transition parameters of the HMM grammar, which in principle alleviate user specification of model parameters [7]; and several kth-order Markov models for various types of background sequences.

We compared our approach to the standard likelihood-ratio (or log odds) scoring approach,

[7] Although sophisticated users could choose to decide the "strength" of the priors, or define their own priors.

and demonstrated superior results on large scale *Drosophila* early developmental enhancer analysis. CisModuler provides a useful and arguably superior alternative approach to detect CRM and motif occurrences based on a given PWM, and can be also used as a subroutine in *de novo* motif/CRM detection from higher eukaryotic genomes.

2.6 LOGOS: for Semi-unsupervised *de novo* Motif Detection

Recall that under the **LOGOS** framework, the local, global and background submodels jointly define the likelihood of an observed DNA sequence that contains unspecified motifs. Each submodel can be designed separately to address different aspects of the biological characteristics of a transcriptional regulatory sequence, and combination of submodels each from a wide spectrum of possible designs is possible. Therefore, **LOGOS** facilitates a flexible trade-off between expressiveness and complexity for motif modeling.

Most extant models for *de novo* motif detection fall into the most basic submodel combination, namely, a PM local model plus a UI global model (denoted by **LOGOS**$_{pu}$ in the sequel). Examples of **LOGOS**$_{pu}$ include the basic models underlying the MEME [Bailey and Elkan, 1995a] and AlignACE [Hughes *et al.*, 2000] programs (although both programs have more sophisticated and efficient implementation, e.g., more careful initiation schemes for over-represented words, which in practice improve their performance over a basic **LOGOS**$_{pu}$ model).

Having both the MotifPrototyper model for local motif structure and the CisModuler model for global motif organization (which also includes the kth-order Markov model for the background), one can envisage a novel generative model for transcriptional regulatory sequences that is significantly more expressive than any extant motif detection models. A graphical representation of such a model, which is referred to as **LOGOS**$_{hh}$ (standing for HMDM + HMM), is depicted in Figure 2.18. (For simplicity, in the sequel we abbreviate **LOGOS**$_{hh}$ with the unsubscripted "**LOGOS**" when no confusion arises in the context, e.g., no comparison with other variations of **LOGOS** is being made.) Specifically, in such a **LOGOS** model, the functional annotations of a DNA sequence that determine the motif locations and modular structures are determined by a CisModuler HMM

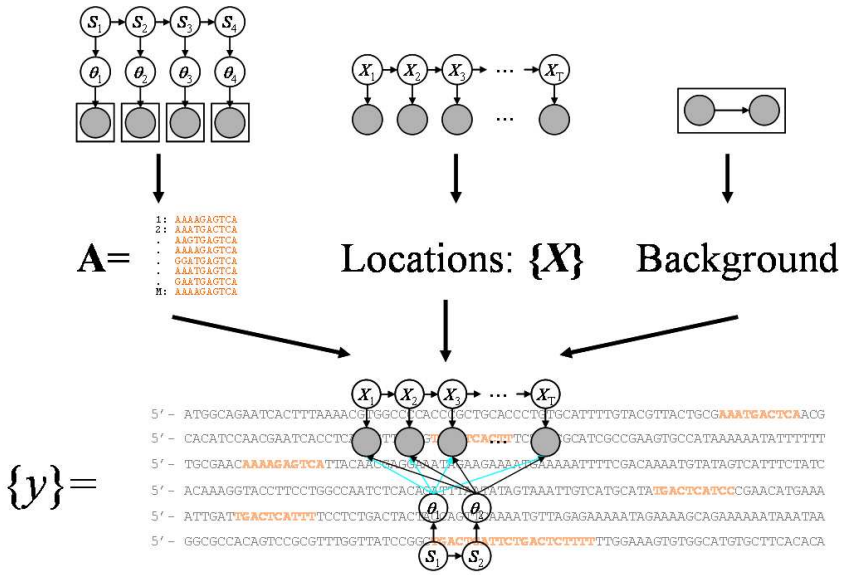


Figure 2.18: A modular construction of the **LOGOS**$_{hh}$ model for motif-bearing sequences. For simplicity, only one possible motif and one study sequence are included in this model.

model; but the emission probabilities of the motif states, or the PWMs of the motifs, are generated from a MotifPrototyper model, whereby prior knowledge regarding both global motif organization and local motif structure is incorporated for *de novo* motif detection, making it a *semi-unsupervised learning* problem. Note that the context-sensitive dependency induced by the latent motif indicator x couples each sequence variable with all existing motif parameters. Thus the nucleotide identity of any position in a sequence is determined by a complex dependency structure that captures intra and inter motif dependencies reflecting both intrinsic structural properties and higher-level correlations of the motifs (detailed in §4.7).

The expressiveness of **LOGOS** comes at the cost of resulting in a very expensive computational problem for probabilistic inference in this model. Specifically, the inference and learning problem associated with **LOGOS** is that of computing the maximum *a posteriori* prediction of motif locations and the Bayesian estimate of motif PWMs. But unlike the situations we discussed when dealing with only the MotifPrototyper or the CisModuler submodels of **LOGOS**, which essentially intend to solve only one of the two aforementioned queries assuming the solution to the other is known, now one needs to solve the two queries jointly, and this turns out to be intractable due to the

exploded state space of the joint model.

Since no off-the-shelf exact algorithm works for **LOGOS**, some approximation schemes have to be used. One option is to pursue a stochastic approximation using MCMC techniques such as Gibbs sampling. In Chapter 5, we describe a Gibbs sampler algorithm for posterior inference on **LOGOS**. But as demonstrated in a preliminary experiment on modest-sized input sequences (see §4.7.2), the Gibbs sampler converges very slowly and appears impractical for supporting a realistic motif detection program. In Chapter 4, we develop a deterministic approximation method called generalized mean field inference. Essentially, a GMF algorithm alternates between solving one of the two sub-problems mentioned before in the respective submodel of **LOGOS**, conditioning on the approximate solution of the other sub-problem, and then updating the approximate solutions using the newly obtained solutions, which yields a better approximation. It can be shown that this algorithm is guaranteed to converge to a locally optimal solution, and defines a lower bound on the likelihood of the study sequences. A full description of the theory and algorithm of GMF inference in general graphical models, and specifically the fixed-point equations for the **LOGOS** model, is deferred to Chapter 4. In the following, we present an extensive validation of the **LOGOS** model on fully annotated semi-realistic datasets and real genomic sequences from yeast, and a preliminary test on a small set of unannotated *Drosophila* genomic sequences.

2.6.1 Experiments

2.6.1.1 Performance on semi-realistic sequence data

Recall that in §2.5, we validated the utility of the MotifPrototyper model for *de novo* motif detection in conjunction with a trivial global model — oops, which assume one motif per sequence. Now we consider a more realistic scenario, in which each study sequence contains multiple motifs. We compare three variants of the **LOGOS** model for this setting, ordered by decreasing model expressiveness, HMDM+HMM (**LOGOS**$_{hh}$), PM+HMM (**LOGOS**$_{ph}$) and PM+UI (**LOGOS**$_{pu}$). Specifically, a slightly simplified **LOGOS**$_{hh}$ is used for the task herein, where the global submodel is a simpler HMM containing only a single global background state in addition to the motif states

(meaning that no CRM structure is modeled), rather than using the highly elaborated CisModuler Bayesian HMM tailored for higher eukaryotic sequences.

Single motif, and multiple instances per sequence. Under a realistic motif detection condition, the number of motif instances is unknown. Rather than trying all possible numbers of occurrences suggested by the user or decided by the algorithm and reporting a heuristically determined plausible number, **LOGOS** uses the global HMM model to describe a posterior distribution for motif instances, which depends on both the prespecified indicator state transition probabilities and the actual sequence y to be analyzed. In this experiment, the transition probabilities are empirically set at a default value to reflect our rough estimates of motif frequencies (i.e., 5%). But as more training data of annotated regulatory sequences are collected, these parameters can be fit in a genome-specific fashion.

Table 2.6: Performance of **LOGOS** for single motif detection, with unknown number of instances per sequence.

motif	**LOGOS**$_{hh}$		**LOGOS**$_{ph}$		**LOGOS**$_{pu}$	
name	FP	FN	FP	FN	FP	FN
abf1	**0.3115**	**0.2116**	0.6774	0.1957	0.7917	0.9123
gal4	**0.1569**	**0.1569**	0.1895	0.1534	0.2917	0.7939
gcn4	**0.1820**	**0.2355**	0.6142	0.2821	0	0.9594
gcr1	**0.1962**	**0.2134**	0.3371	0.2038	0.3333	0.9437
mat	**0.0723**	**0.0337**	0.3563	0	0.5000	0.9643
mcb	0.3734	0.0910	**0.3628**	**0.0792**	0.3333	0.9431
mig1	**0.0774**	**0**	0.0854	0	0.9764	0.1000
crp	**0.3768**	**0.3398**	0.2727	0.5294	0	0.9487

Table 2.6 summarizes the performance of three variants of **LOGOS** for single motif detection, with an unknown number of instances per sequence. We present the median false positive (FP) and false negative (FN) rates (in terms of finding each instance of the motifs within an offset of 3 bp) of motif detection experiments over 20 test datasets. Each test dataset consists of 20 sequences, each generated by planting (uniformly at random) 0–7 instances of a motif (real sites from SCPD), together with its permuted "decoy," in a 300–400 bp random background sequence. As Table 2.6 shows, **LOGOS**$_{pu}$ yields the weakest results, losing in all 8 motif detections (in terms of (FP+FN)/2), suggesting that the conventional PM+UI model, which is used in MEME, and with

slight variation, in AlignACE and BioProspector, is not powerful enough to handle non-trivial detection tasks as posed by our test set. **LOGOS**$_{ph}$ improves significantly over **LOGOS**$_{pu}$, even yielding the best performance in one case (for *mcb*), suggesting that the HMM global model we introduced indeed strengthens the motif detector. Finally, as hoped, **LOGOS**$_{hh}$ yields the strongest results, performing best on 7 of the 8 motifs, convincingly showing that capturing the internal structures of motifs and making use of prior knowledge from known motifs, combined with the use of the HMM global model, can yield substantially improved performance. Our results are reasonably robust under different choices of the global HMM parameters.

Simultaneous detection of multiple motifs. Detecting multiple motifs simultaneously is arguably a better strategy than detecting one at a time and then deleting or masking the detected motifs, especially when motif concentrations are high, because the latter strategy mistakenly treats the other motifs as background, causing potentially suboptimal estimation of both motif and background parameters. The global HMM model we propose readily handles simultaneous multiple motif detection (say, finding K motifs at a time): we only need to encode all motif states into the state space $\mathbb{S}$ of the motif indicator X, and perform standard HMM inference. The locations of all motifs can be directly read off from the state configuration of x. Table 2.7 summarizes the results on 20 test sets each containing 20 sequences harboring motifs *abf1*, *gal4* and *mig1* (0–6 total instances/seq). The upper panels show the predictive performance based on the optimal (in terms of maximal log-likelihood of y from 50 independent runs of the GMF algorithm) posterior expectation of X. Note that with a MotifPrototyper local model, **LOGOS**$_{hh}$ exhibits better performance. In the lower panels, we show the best FP-FN results in the top three predictions (i.e., top 3 PWMs for each of the K motifs we look for) made by **LOGOS** (note that 'K-at-a-time' prediction yields a total of $3K$ possibly redundant motif patterns). This is close to the stochastic dictionary scenario where the predicted motif is to be identified from the optimal dictionary of the patterns resulting from the motif detection program [Gupta and Liu, 2003]. It is expected that a human observer could easily pick out the biologically more plausible motifs when given a visual presentation of the most likely

motifs suggested by a motif finder.

Table 2.7: Simultaneous multiple motif detection (median FP-FN rate over 20 test sets containing three motifs.)

		LOGOS$_{hh}$		**LOGOS**$_{ph}$	
		FP	FN	FP	FN
MAP prediction	abf1	0.3591	0.3274	0.7778	0.7434
	gal4	0.1259	0.1714	0.3751	0.1491
	mig1	0.3849	0.2243	0.3481	0
best of top 3 prediction	abf1	0.3841	0.2400	0.4721	0.3972
	gal4	0.0926	0.0986	0.2609	0.1255
	mig1	0.1250	0.0333	0.2318	0

Detecting motifs of uncertain lengths. A useful property of the MotifPrototyper submodel is that it actually does not need to know the exact lengths of the motifs to be detected, since the MotifPrototyper allows a motif to start (and end) with consecutive heterogeneous sites. Thus, a blurred motif boundary is permissible, especially when the resulting window is large enough to cover at least the entire length of the motif. As a result, we do not have to know the exact length of the motif, but just need to roughly guess it conservatively, during *de novo* motif detection. This is another appealing feature of **LOGOS**, which extends its flexibility. As shown in Table 2.8, even in simultaneous multiple motif detection, with improperly specified motif lengths, **LOGOS**$_{hh}$ performs nearly as well as when motif lengths are precisely specified, whereas **LOGOS**$_{ph}$ is not as good.

Table 2.8: Simultaneous detection of three motifs, with lengths improperly specified (18, 22, and 20 bp, respectively, instead of the actual 13, 17, and 11 bp).

		LOGOS$_{hh}$		**LOGOS**$_{ph}$	
		FP	FN	FP	FN
MAP prediction	abf1	0.7295	0.6667	0.8021	0.7680
	gal4	0.1167	0.2042	0.2357	0.1325
	mig1	0.4183	0.2128	0.8150	0.8381
best of top 3 prediction	abf1	0.3310	0.2804	0.5742	0.4821
	gal4	0.0955	0.1222	0.1882	0.1250
	mig1	0.2124	0.1327	0.3218	0.1623

2.6.1.2 Motif detection in yeast promoter regions

In this section we report a performance comparison of **LOGOS** (HMM+HMDM) with two popular motif detection programs, MEME and AlignACE, on 12 yeast genomic sequence sets gathered

from the SCPD database (the selection is based on having at least a total of 5 motif instances in all sequences and the motif being independent of our training set). Each sequence set consists of multiple yeast promoter regions each about 500 bp long and containing on both strands an unknown number of occurrences of a predominant motif (but also possibly other minor motifs) as specified by the name of the dataset (Table 2.9, where the rightmost column gives the number of sequences in each dataset). Note that both the relatively large sizes of the input sequences and the possible presence of motifs other than what has been annotated make the motif finding task significantly more difficult than a semi-realistic test data or small, well curated real test data. We use the following command to run MEME: " meme $efile -p 2 -dna -mod tcm -revcomp -nmotifs 1. " In practice, this means that it searches for a DNA sequence on both strands for at most one motif, which can occur zero or more times in any given sequence. AlignACE is run with default command-line arguments nearly identical to those for MEME, with the only difference that AlignACE can return multiple predicted motifs (of which we select the best match from the top five MAP predictions). **LOGOS** is set in the multiple-detection mode and is used to make two motif predictions simultaneously. As shown in Table 2.9, for this non-trivial *de novo* motif detection task, **LOGOS** outperforms the other two programs by a significant margin.

Table 2.9: Comparison of motif detectors on yeast promoter sequences.

set	**LOGOS**		MEME		AlignACE		seq
name	FP	FN	FP	FN	FP	FN	no.
abf1	0.7949	0.6522	1.0000	1.0000	**0.5294**	**0.6087**	20
csre	**0.4444**	**0.1667**	0.7778	0.5000	0.8000	0.5000	4
gal4	**0.1333**	**0.0714**	0.1667	0.2857	0.3333	0.1429	6
gcn4	**0.3529**	**0.1852**	1.0000	1.0000	0.3333	0.5556	9
gcr1	**0.2859**	**0.6154**	1.0000	1.0000	0.4545	0.4615	6
hstf	0.8571	0.5556	**0.6000**	**0.5556**	0.8500	0.6667	6
mat	**0.4194**	**0**	0.3750	0.5625	0.2500	0.2500	7
mcb	0.4706	0.2500	0.2000	0.3333	**0.2500**	**0.2500**	6
mig1	**0.8077**	**0.2857**	1.0000	1.0000	0.8333	0.7857	22
pho2	**0.9024**	**0.5000**	1.0000	1.0000	1.0000	1.0000	3
swi5	**0.7647**	**0.5000**	1.0000	1.0000	0.9412	0.7500	2
uash	**0.8250**	**0.6818**	1.0000	1.0000	0.9231	0.9545	18

2.6.1.3 Motif detection in *Drosophila* regulatory DNAs

In this section, we report on a preliminary *de novo* motif discovery analysis of the regulatory regions of the 9 *Drosophila* genes involved in body segmentation. The input data consists of 9 DNA

sequences ranging from 512 to 5218 bp, as described in [Berman *et al.*, 2002]. Biologically identified motifs include *bcd*, *cad*, *hb*, *kni* and *kr*. For comparison, we provide the PWMs postulated by Berman *et al.* for these five motifs, which were used in their motif scan analysis (Figure 2.19). The sources of all PWMs are biologically identified sequence segments from the literature (which are unaligned, ranging from 5 to 93 instances per motif, and about $20 \sim 40$ bases in length). The PWMs are derived from an alignment of all these identified motif sequences.

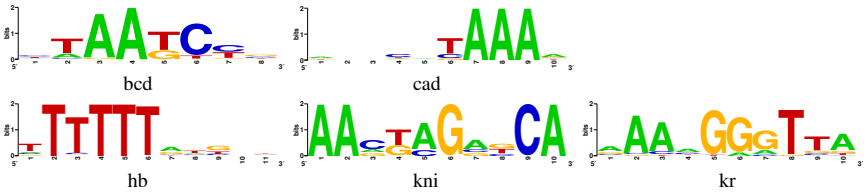


Figure 2.19: Berman *et al.*'s *Drosophila* motif patterns derived from multi-alignments of biologically identified motif instances.

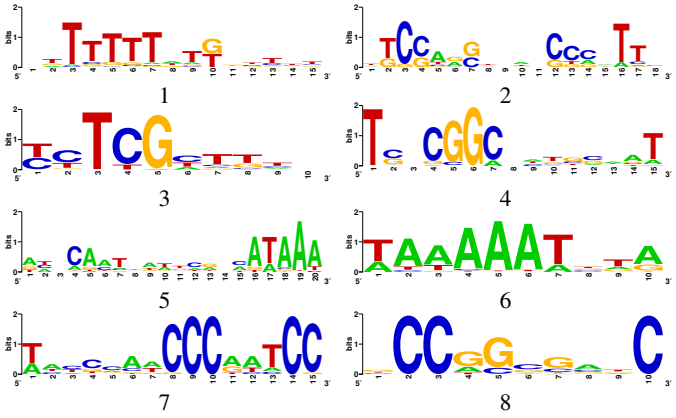


Figure 2.20: Motif patterns detected by **LOGOS** in the regulatory regions of 9 *Drosophila* genes.

We applied **LOGOS** (which is set to identify 4 motifs at a time) to the *Drosophila* dataset; Figure 2.20 gives a partial list of the top-scoring motif patterns (of the top three runs out of a total of 50 runs, evaluated by the likelihood under the **LOGOS** model at convergence). Note that the *logos* shown here are not the conventional sequence logos based on counts of aligned nucleotides; instead we use the logo visualization software to graphically present the **Bayesian estimate** of the position-specific multinomial parameters θ of each motif, so they are not necessarily equal to the usual nt frequencies of aligned sequences, but represent a more robust probabilistic model of the

motif sequences. A visual inspection reveals that patterns 1 and 5 correspond to the *hb* and *cad* binding sites, respectively (as confirmed by the matching locations of our results and the sequence annotations). Part of pattern 2 agrees with the reverse complement of the *kr* motif (containing -CCCxTT-), but this motif seems to be actually a "two-block" motif because the pattern we detected under a longer estimated motif length contains an additional co-occurring conserved pattern a few bases upstream. Part of pattern 7 is close to the *bcd* motif (containing -AATCC-) but also contains additional sites (i.e., the three highly conserved C's upstream), which turned out to result from a number of false positive substrings picked up together with the true *bcd* motifs. A careful examination of pattern 6 suggests that it may be actually derived from putative motif subsequences that correspond to the *kni* binding site. This is not obvious at first because it appears quite different from the *kni* logo in Figure 2.19. But after seeing an example *kni* site in stripe 2/7: 5'agaaaactagatca3', starting at position 35, we realized that this answer might be plausible. The discrepancy is likely due to artifacts in the original generation of the alignment data supporting the *kni* logo: only 5 biologically identified instances were used and they are quite diverse; the resulting multiple alignment is visually sub-optimal in that homogeneous sites are severely interspersed with heterogeneous sites. Patterns 3, 4, and 8 are putative motifs not annotated in the input sequences. We also ran the same dataset through MEME (also 4 patterns to be found a time) and the output is in general weaker and harder to interpret. Figure 2.21 shows the best three patterns, from which one could recognize a *hb* (pattern 1) and a *cad* (pattern 3). Note that the motif logos given in Figure 2.19 are based on the nucleotide-frequency profiles of biologically identified instances from many sources. Thus it is not surprising that some of the patterns we found are similar to but do not match the logos in Figure 2.19 exactly since our logos are derived from Bayesian estimates of the motif parameters and our data source consists of a small number of regulatory regions of the *Drosophila* genome, which might be smaller and less representative compared to the data source underlying Figure 2.19 (except for *kni*).

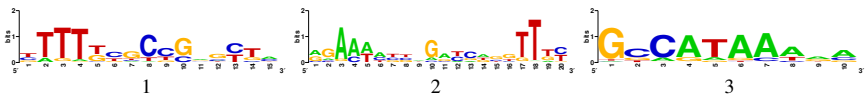


Figure 2.21: Motif patterns detected by MEME in the regulatory regions of the *Drosophila* eve-skipped gene.

2.7 Conclusions

In this chapter, we presented a modular, parametric Bayesian model, **LOGOS**, to capture various aspects of the characteristics of DNA motifs in the transcriptional regulatory sequences, including canonical structures of motif families, syntax of motif organization, and the distribution of background sequences. Using a graphical model formalism, **LOGOS** manifests a modular architecture for the motif model, which consists of a local submodel for the sequence composition of motif sites, a global submodel for the locational distribution of motif sites in the genomic sequences, and a background submodel for non-motif sequences — addressing different aspects of motif properties in a divide-and-conquer fashion.

We developed a MotifPrototyper model for local motif alignment, which captures site dependencies inside motifs and incorporates learnable prior knowledge from known motifs for Bayesian estimation of the PWMs of novel motifs in unseen sequences. We also developed a CisModuler HMM model for the global motif distribution, which introduces dependencies among motif instances and allows efficient and consistent inference of motif locations. A deterministic algorithm based on generalized mean field approximation will be described in Chapter 4 to solve the complex missing value and Bayesian inference problems associated with the **LOGOS** model. As will be explained shortly, GMF allows probabilistic inference in the local alignment and the global distribution submodels to be carried out virtually separately with a proper Bayesian interface connecting the two processes. This divide and conquer strategy aligned with the modular architecture of **LOGOS** makes it much easier to develop more sophisticated models for various aspects of motif analysis without being overburdened by the daunting complexity of the full motif problem.

Due to the functional diversity of the DNA motifs, it is expected that there could exist more complex dependencies and regularities in the structures of motifs. Thus, further investigations into these properties and more powerful local models for motifs are needed. Similarly, the HMM-based global model we proposed is only a first step beyond the conventional UI model, and is only able to capture dependencies between motifs and motif clusters at a very limited level (e.g., it cannot model

higher-order dependencies such as hierarchical structures and long-distance influence between motifs). More expressive models are needed to achieve these goals. Nevertheless, under the **LOGOS** architecture, extensions from baseline models are modular and the probabilistic computations involved can also be handled in a divide-and-conquer fashion via generalized mean field inference. We are optimistic that **LOGOS** can serve as a flexible framework for motif analysis in biopolymer sequences.

Chapter 3

Modeling Single Nucleotide Polymorphisms for Haplotype Inference

— A Nonparametric Bayesian Approach

In addition to unveiling the genetic code underlying the structure, localization, and regulation of biopolymer macromolecules such as proteins and RNAs that are essential for biological activities, and thereby facilitating mechanistic analysis of the function and evolution of various organisms, the availability of nearly complete genome sequences for organisms such as humans also makes it possible to begin to explore individual differences between DNA sequences on a genome-wide scale, and to search for associations of such genotypic variations with diseases and other phenotypes [Risch, 2000].

The largest class of individual differences in DNA are the *single nucleotide polymorphisms*, or SNPs. Millions of SNPs have been detected thus far out of an estimated total of ten million common SNPs [Sachidanandam *et al.*, 2001; Venter *et al.*, 2001]. SNPs are promising markers for population genetic studies and for localizing genetic variations potentially responsible for complex diseases due to their high density, low mutation rate, and amenability to automated genotyping [Patil *et al.*, 2001]. However, each individual SNP only yields limited information regarding populational variation and disease association [Akey *et al.*, 2001]. It is known that studies using haplotype information of multiple linked SNPs generally outperform those using single-marker analysis [Weiss and Clark, 2002; Clark, 2003]. Thus it is important to know the haplotype structure of the genome in the

population under study. In this chapter, we present a novel nonparametric Bayesian approach for haplotype inference from SNP genotype data. Some of the material in this thesis has appeared before in [Xing *et al.*, 2004c].

3.1 Biological Foundations and Motivation

Recall that a chromosome is a complete strand of DNA in the genome. For diploid organisms such as humans, each individual has two physical copies of each chromosome in his/her somatic cells. One copy is inherited from the mother, and the other from the father.

A SNP commonly has two variants, or *alleles*, at a single chromosomal locus in the population, corresponding to two specific nucleotides chosen from $\{A, C, G, T\}$. [1] Essentially, SNPs are genetic variations in the same chromosomal locus among different individuals in a population, which are usually neutral nucleotide substitutions that are not necessarily functionally essential and do not substantially affect the fitness of their bearers [Kruglyak and Nickerson, 2001]. Thus, they are believed to result from ancient neutral mutations that took place in the ancestors of the modern population, and may carry important information about tribal or ethnic group formation, evolution and migration [Stoneking, 2001].

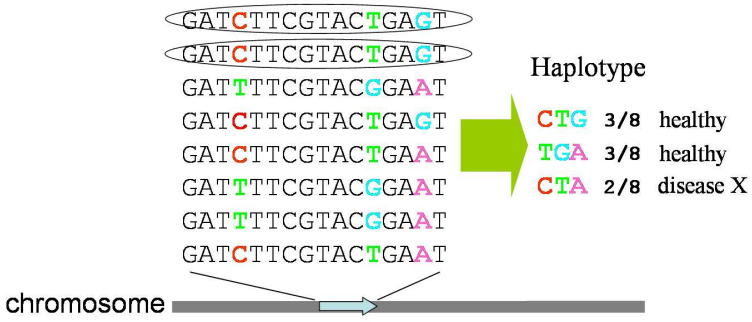


Figure 3.1: SNP haplotypes and possible phenotype associations.

A *haplotype* is a list of alleles at consecutive sites in a local region of a single chromosome.

[1] An *allele* is a variant of a SNP, a gene, or some other entity associated with sites in DNA. In our case (SNPs), the sites are single nucleotides, and the alleles can generally be assumed to be binary, reflecting the fact that lightning (mutation) doesn't tend to strike twice in the same place.

It can be regarded as a state configuration of a particular chromosome (Fig. 3.1). Although in general the individual SNPs are themselves not related to functionality, the SNPs haplotypes may co-occur with some disease-related phenotypes due to physical proximity of the haplotype to possible causal regions on the DNA genome, which could lead to co-inheritance [Akey *et al.*, 2001; Daly *et al.*, 2001; Pritchard, 2001]. Therefore, haplotypes can be used for inferring the chromosomal locations of the genes underlying diseases. Assuming no recombination in a local region containing multiple SNPs, a haplotype is inherited as a unit. Recall that for diploid organisms (such as humans) the chromosomes come in pairs. Thus two haplotypes go together to make up a *genotype*, which is the list of *unordered* pairs of alleles in a region. That is, a genotype is obtained from a pair of haplotypes by omitting the specification of the association of each allele with one of the two chromosomes—its *phase*. Phase information can be critical to the mapping of a disease gene, by allowing a more precise and robust localization of it within a target area via a linkage analysis which assesses the level and significance of statistical associations between disease phenotypes and genetic markers [Akey *et al.*, 2001; Clark, 2003]. To date, haplotype mapping has been successfully employed for a number of monogenic diseases, such as cystic fibrosis and Huntington's [Lazzeroni, 2001]; and has appeared valuable in locating susceptibility genes in complex multigenic disorders [Puffenberger *et al.*, 1994; Hugot *et al.*, 2001; Rioux *et al.*, 2001]. In these cases, exploiting haplotype information can greatly reduce the number of assays necessary to genotype a subject's genome and thus facilitate comprehensive whole-genome association studies for mapping complex diseases.

Common biological methods for assaying genotypes typically do not provide phase information for individuals with heterozygous genotypes at multiple autosomal loci (Fig. 3.2); phase can be obtained at a considerably higher cost via molecular haplotyping [Patil *et al.*, 2001]. In addition to being costly, these methods are subject to experimental error and are low-throughput. Alternatively, phase can also be inferred from the genotypes of a subject's close relatives [Hodge *et al.*, 1999]. But this approach is often hampered by the fact that typing family members increases the cost and does not guarantee full informativeness. It is desirable to develop automatic and robust methods for

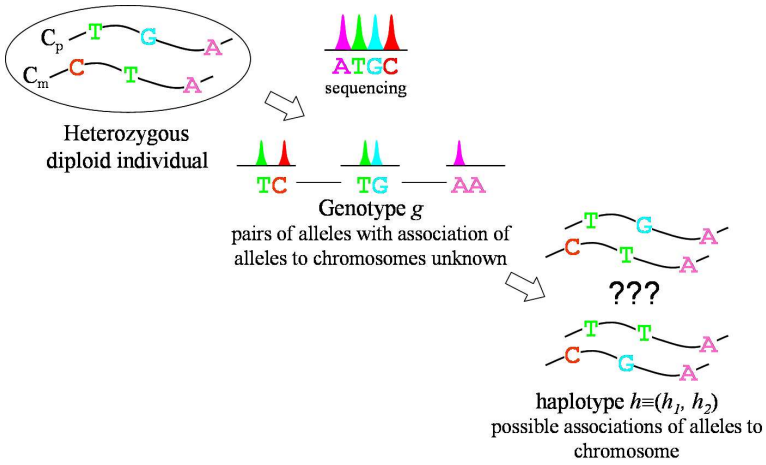


Figure 3.2: Phase ambiguity. For a heterozygous individual, who has different SNP alleles on the pair of chromosomes at multiple loci, a standard sequencing experiment only yields the joint identity of both alleles of each SNP locus (i.e., the genotypes), whereas the exact chromosomal association of the alleles (i.e., the haplotype, or phase) of the SNP sequence is lost. (This is because that it is technically difficult to sequence the paired chromosomes in a cell separately in a standard sequencing experiment, which simply blends all cell extracts in the same test tube.) It is often the case that for given genotypes of multiple SNPs, there exist multiple consistent haplotype reconstructions. For example, the genotypes shown here can be consistently explained by either one of these two possible associations of alleles to chromosomes.

inferring haplotypes from genotypes and possibly other data sources (e.g., pedigrees). As pursued in this chapter, *in silico* phasing programs based on explicit statistical models are a feasible approach to meet these goals.

3.2 Problem Formulation and Overview of Related Work

From the point of view of population genetics, the basic model underlying the haplotype inference problem is a finite mixture model. That is, letting $\mathcal{H}$ denote the set of all possible haplotypes associated with a given region (a set of cardinality 2^k in the case of binary polymorphisms, where k is the number of heterozygous SNPs), the probability of a genotype is given by:

$$p(g) = \sum_{h_1,h_2 \in \mathcal{H}} p(h_1, h_2) p(g|h_1, h_2), \tag{3.1}$$

where the likelihood model (i.e., 2nd term on the r.h.s.), which defines the probability of an (observed) genotype pattern given a pair of (latent) haplotype patterns, is referred to as a *genotype model*; the mixing proportion (i.e., 1st term on the r.h.s.), which defines the joint probability of a

pair of haplotypes, is referred to as a *haplotype model*; and the space of all possible haplotypes in this region in a population is called the *population haplotype pool*. In the standard setting (e.g., [Excoffier and Slatkin, 1995]), one usually assumes that:

- The genotype model is deterministic (i.e., typing is considered as noiseless),

$$p(h_1, h_2)p(g|h_1, h_2) = \mathbb{I}(h_1 \oplus h_2 = g)$$

  where $\mathbb{I}(h_1 \oplus h_2 = g)$ is the indicator function of the event that haplotypes h_1 and h_2 are consistent with g.

- The pair of haplotypes of an individual are subject to Hardy-Weinberg equilibrium (HWE) (i.e., the pair of haplotypes are independently inherited) [Lange, 2002], an assumption that is standard in the literature and will also be made here,

$$p(h_1, h_2) = p(h_1)p(h_2)$$

- The size of the the population haplotype pool is set fixed (to a manageable integer) to avoid exhaustive enumeration,

$$\mathcal{H} = K \ll 2^k$$

Given this basic statistical structure, the haplotype inference problem can be viewed a *missing value inference* and *parameter estimation* problem. Numerous statistical models and statistical inference approaches have been developed for this problem, which will be briefly reviewed shortly. There is also a plethora of combinatorial algorithms based on various deterministic models of haplotypes. While recognizing their effectiveness in a number of occasions and important insights they provide to the problem, we choose to forego an extensive discussion of this literature (but see [Gusfield, 2004] for an overview) and focus on statistical methods in this chapter. It is our view that the statistical approaches provide more flexibility in handling missing values (e.g., occasional missing genotyping outcomes), typing errors, evolution modeling and more complex scenarios on the horizon in haplotype modeling (e.g., recombinations, gene linkage, etc.).

3.2.1 Baseline Finite Mixture Model and the EM Approach

Given the statistical structure illustrated in Eq. (3.1), the simplest methodology for haplotype inference is maximum likelihood via the EM algorithm, treating the haplotype identities as latent variables and estimating the parameters $p(h)$, usually referred to as *population haplotype frequencies* f_h, assuming that the individual haplotypes are *iid* following a multinomial distribution parameterized by $\{f_h : h \in \mathcal{H}\}$ [Excoffier and Slatkin, 1995]. This methodology has rather severe computational requirements, in that a probability distribution must be maintained on the (large) set of possible haplotypes, but even more fundamentally it fails to capture the notion that small sets of haplotypes should be preferred. This notion derives from an underlying assumption that for relatively short regions of the chromosome there is limited diversity due to population bottlenecks and relatively low rates of recombination and mutation.

The key shortcoming of the aforementioned EM-based finite mixture model lies in its inability to take into account uncertainty about the the number of haplotypes (i.e., the number of mixture components), and to impose appropriate statistical bias. This problem is, up to a terminological mapping, closely related to clustering problems that are commonly studied in machine learning and data mining literature. In particular, collaborative filtering involves the clustering of sets of choices made by sets of individuals, and this clustering problem is closely related to the clustering of sets of alleles in sets of chromosomes. In these domains, the perennial problem of "how many clusters?" is well known, and is particularly salient in large data sets where the number of clusters needs to be relatively large and open-ended. At one time, an EM algorithm can only handle a pre-fixed integer number of mixture components (e.g., 2^k or a smaller number K of possible haplotypes). In haplotype phasing, such an approach does not return any estimate of uncertainty about the specific number of haplotypes that it finds, and heuristics such as cross-validation needs to be used to empirically pick a favorable K.

3.2.2 Bayesian Methods via MCMC

One approach to dealing with the issue of the unknown number of mixture components, and the desirable bias for more compact phase reconstruction, is to formulate a notion of "parsimony," and to develop algorithms that directly attempt to maximize parsimony. Several important papers have taken this approach [Clark, 1990; Clark *et al.*, 1998; Gusfield, 2002; Eskin *et al.*, 2003] and have yielded new insights and algorithms. Another approach is to elaborate the probabilistic model, in particular by incorporating priors on the parameters. Different priors have been discussed by different authors as outlined in the following. These models provide implicit notions of parsimony, via the implicit "Ockham factor" of the Bayesian formalism [Bernardo and Smith, 1994].

3.2.2.1 Simple Dirichlet priors

The PL model proposed by Niu *et al.* [Niu *et al.*, 2002], which was implemented in the software HAPLOTYPER, incorporates simple Dirichlet priors to the haplotype frequencies, $\{f_h\}$, to be estimated (no prior for the haplotypes themselves are introduced):

$$p(\{f_h\}) = \frac{\Gamma(\sum_h \beta_h)}{\prod_h \Gamma(\beta_h)} \prod_h \left[f_h\right]^{\beta_h - 1} \tag{3.2}$$

As indicated by Stephens *et al.* [2003], the Dirichlet priors correspond to a simple, but highly unrealistic assumption about the genetic processes underlying the evolution of the study population — that the genetic sequence of a mutant offspring does not depend on the progenitor sequence.

As is standard in Bayesian inference, an MCMC algorithm, specifically, a novel Gibbs sampling scheme, was used to compute the Monte Carlo estimates, i.e., the haplotype frequencies and the individual haplotypes. In particular, two computational tricks — *prior annealing* and *partition-ligation* (from which comes the name of the model) — appeared to significantly reduce the computational effort required to obtain a good approximation to the true posterior distribution of the aforementioned estimators of interest.

3.2.2.2 The coalescent prior

The model introduced by Stephens *et al.* [2001] (referred to as the SSD model after its authors) is based on a more elaborate Bayesian framework, which assumes that the unobserved haplotypes are subject to a prior that considers how randomly sampled individuals are related genealogically via a neutral coalescent [Stephens and Donnelly, 2000]. For computational feasibility (i.e., not having to marginalize over a space of all valid genealogical trees), they devised a Gibbs sampler that samples individual haplotypes from a conditional distribution that approximates the coalescent:

$$\pi(h = \beta|H) = \sum_{\alpha \in \mathcal{H}} \sum_{s=0}^{\infty} \frac{r_\alpha}{r} \left(\frac{\theta}{r+\theta}\right)^s \frac{\theta}{r+\theta} (T^s)_{\alpha\beta}, \tag{3.3}$$

where r_α is the number of haplotypes of type α in the set H (the set of all haplotypes in the study population excluding the next sampled haplotype, h), r is the cardinality of H, θ is a scaled mutation rate, and T is the substitution probability matrix between all pairs of haplotypes.

The coalescent prior is arguably more realistic than the "parent-independent" mutation model underlying the simple Dirichlet prior in the PL model, and favors mutant offspring that differ only slightly from the progenitor sequences, hence implicitly introducing a parsimonious bias. A recent paper by Lin *et al.* [2002] also described a number of modifications to the SSD model, which appears to slightly compromise the approximation to the coalescent prior but on the other hand improves the efficiency of the sampling algorithm in the original implementation of SSD. The latest version of the software PHASE, where the coalescent prior is used, also assimilates the computational tricks (i.e. prior annealing and ligation) contributed by Niu *et al.* [2002], and represents the state of the art haplotype inference program, significantly and constantly beating other extant methods on real and simulated data.

3.2.3 Bayesian Network Prior

Note that the coalescent model does not readily generalize to more complex scenarios such as possible recombinations within a stretch of SNPs. A recombination could reduce the linkage disequilibrium, or in other words, decouple the subsets of SNPs on the two sides of the recombination spot

on the chromosome. Exploiting this phenomenon, or finding SNP blocks with high internal linkage (even though they do not necessarily arise from the presence of true recombinations hotspots that define their boundaries), could help to optimally decompose the difficult problem of phasing long stretches of SNPs into multiple subproblems of manageable sizes, i.e., phasing each block of SNPs separately, and then trying to stitch together the sub-solutions. Greenspan *et al.* [2003] attempted to model the events of recombination on a chromosome as a 1st-order hidden Markov process, defining a segmentation of the whole sequence of SNPs. Associated with each block of consecutive SNPs resulting from the segmentation is a block-specific distribution of *ancestral haplotypes*. For each chromosome, the choice of ancestral haplotype at each block is determined by the latent *recombination variables* associated with each block. The configuration of the recombination variable at each block is 1st-order Markovian with respect to the recombination variables of the previous blocks. Within each block, each individual haplotype is a possibly corrupted (via mutations) version of the ancestral haplotype under a stochastic mutation model. This model readily handles missing values and mis-typings in SNP data acquisition, and elegantly facilitates a divide-and-conquer strategy for large phasing problem. Note that the number of ancestral haplotypes at each block and the boundaries of the blocks are unknown model parameters. A minimum description length (MDL) criterion is used for model selection in conjunction with an EM algorithm.

Identifying and interpreting haplotype blocks is a standing-along problem that has received much attention in recent years due to its relevance to understanding the linkage disequilibrium structures of chromosomes and the evolution history of the genome. In addition to the work of Greenspan *et al.* [2003], numerous methods outside the context of haplotype phasing (i.e., focusing on empirically phased data), such as dynamic programming [Zhang *et al.*, 2002], HMM [Daly *et al.*, 2001], and MDL [Anderson and Novembre, 2003], have been reported. We view these as a complementary issue to the problem we are interested in here, and forego an extensive review.

3.2.4 Summary and Prelude to Our Approach

Extant approaches for phasing rely on the plausible assumption that, locally, haplotype data has limited diversity. This constraint is modeled in different ways by the different methods, often leading to "guessing" in advance a parameter that represents the size of the genetic pool in the population. No current approach suggests an explicit probabilistic model for this quantity, but rather an empirical estimate is used. Such an approach fails to take into account uncertainty in this important quantity. Moreover, to ensure success, this quantity has to be set large so that no or few individual haplotype configurations will be missed. This heuristic causes a computational burden and may bias the algorithm toward non-parsimonious solutions with a large number of rare haplotypes.

In the following we also take a Bayesian statistical approach, but we attempt to provide more explicit control over the number of inferred haplotypes than has been provided by the statistical methods proposed thus far. The resulting inference algorithm has commonalities with the parsimony-based schemes.

The approach to be presented is based on a nonparametric prior known as the *Dirichlet process* [Ferguson, 1973]. In the setting of finite mixture models, the Dirichlet process — not to be confused with the Dirichlet distribution — is able to capture uncertainty about the number of mixture components [Escobar and West, 2002]. The basic setup can be explained in terms of an urn model, and a process that proceeds through data sequentially. Consider an urn which at the outset contains a ball of a single color. At each step (i.e., for each data point) we either draw a ball from the urn and replace it with two balls of the same color, or we are given a ball of a new color which we place in the urn, with a parameter defining the probabilities of these two possibilities. The association of data points to colors defines a "clustering" of the data.

To make the link with Bayesian mixture models, we associate with each color a draw from the distribution defining the parameters of the mixture components. This process defines a *prior distribution* for a mixture model with a random number of components. Multiplying this prior by a likelihood yields a *posterior distribution*. Markov chain Monte Carlo algorithms have been developed to sample from the posterior distributions associated with Dirichlet process priors [Escobar

and West, 2002; Neal, 2000].

The usefulness of this framework for the haplotype problem should be clear—using a Dirichlet process prior we in essence maintain a pool of haplotype candidates that grows as observed genotypes are processed. The growth is controlled via a parameter in the prior distribution that corresponds to the choice of a new color in the urn model, and via the likelihood, which assesses the match of the new genotype to the available haplotypes.

To expand on this latter point, an advantage of this probabilistic formalism is its ability to elaborate the observation model for the genotypes to include the possibility of errors. In particular, the indicator function $\mathbb{I}(h_1 \oplus h_2 = g)$ in Eq. (3.1) is suspect—there are many reasons why an individual genotype may not match with a current pool of haplotypes, such as the possibility of mutation or recombination in the meiosis for that individual, and/or errors in the genotyping or data recording process. Such sources of small differences should not lead to the inference procedure spawning new haplotypes.

In the following we present a statistical model for haplotype inference based on a Dirichlet process prior and a likelihood that includes error models for genotypes. A Markov chain Monte Carlo procedure, in particular a procedure that makes use of both Gibbs and Metropolis-Hasting updates, for posterior inference, will be described in Chapter 5.

3.3 Haplotype Inference via the Dirichlet Process

The input to a phasing algorithm can be represented as a *genotype matrix* G with columns corresponding to SNPs in their order along the chromosome and rows corresponding to genotyped individuals. $G_{i,j}$ represents the information on the two alleles of the i-th individual for SNP j. we denote the two alleles of a SNP by 0 and 1, and $G_{i,j}$ can take on one of four values: 0 or 1, indicating a homozygous site; 2, indicating a heterozygous site; and '?', indicating missing data.[2]

We will describe the model in terms of a pool of ancestral haplotypes, or *templates*, from which

[2] Although we focus on binary data here, it is worth noting that our methods generalize immediately to non-binary data.

each individual haplotype originates [Greenspan and Geiger, 2003]. The haplotype itself may undergo point mutation with respect to its template. The size of the pool and its composition are both unknown, and are treated as random variables under a Dirichlet process prior. We begin by providing a brief description of the Dirichlet process and subsequently show how this process can be incorporated into a model for haplotype inference.

3.3.1 Dirichlet Process Mixture

Rather than present the Dirichlet process in full generality, we focus on the specific setting of mixture models, and make use of an urn model to present the essential features of the process. For a fuller presentation, see, e.g., Ishwaran and James [2001]. Assume that data x arise from a mixture distribution with mixture components $p(x|\phi)$. Also assume the existence of a *base measure* $G(\phi)$, which is one of the two parameters of the Dirichlet process. (The other is the parameter τ, which we present below). The parameter $G(\phi)$ is not the prior for ϕ, but is used to generate a prior for ϕ, in the manner that we now discuss.

Consider the following process for generating samples $\{x_1, x_2, \ldots, x_n\}$ from a mixture model consisting of an unspecified number of mixture components, or *equivalence classes*:

- The first sample x_1 is sampled from a distribution $p(x|\phi_1)$, where the parameter ϕ_1 is sampled from the base measure $G(\phi)$.
- The ith sample, x_i, is sampled from the distribution $p(x|\phi_{c_i})$, where:
  - The equivalence class of sample i, c_i, is drawn from the following distribution:

$$p(c_i = c_j \text{ for some } j < i | c_1, \ldots, c_{i-1}) = \frac{n_{c_j}}{i - 1 + \tau} \tag{3.4}$$

$$p(c_i \neq c_j \text{ for all } j < i | c_1, \ldots, c_{i-1}) = \frac{\tau}{i - 1 + \tau}, \tag{3.5}$$

    where n_{c_i} is the *occupancy number* of class c_i—the number of previous samples belonging to class c_i.

– The parameter ϕ_{c_i} associated with the mixture component c_i is obtained as follows:

$$\phi_{c_i} = \phi_{c_j} \quad \text{if } c_i = c_j \text{ for some } j < i \text{ (i.e., } c_i \text{ is a populated equivalence class)}$$
$$\phi_{c_i} \sim G(\phi) \quad \text{if } c_i \neq c_j \text{ for all } j < i \text{ (i.e., } c_i \text{ is a new equivalence class)}$$

Eqs. (3.4) and (3.5) define a conditional prior for the equivalence class indicator c_i of each sample during a sequential sampling process. They imply a self-reinforcing property for the choice of equivalence class for each new sample—previously populated classes are more likely to be chosen.

It is important to emphasize that the process that we have discussed will be used as a *prior distribution*. We now embed this prior in a full model that includes a likelihood for the observed data. In Section 5.3 we develop Markov chain Monte Carlo inference procedures for this model.

3.3.2 DP-Haplotyper: a Dirichlet Process Mixture Model for Haplotypes

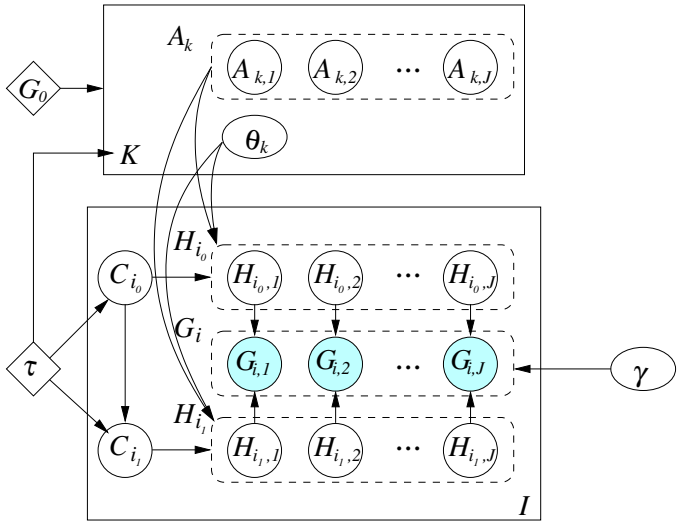


Figure 3.3: The graphical model representation of the haplotype model with a Dirichlet process prior. Circles represent the state variables, ovals represent the parameter variables, and diamonds represent fixed parameters. The dashed boxes denote sets of variables corresponding to the same ancestral template, haplotype, and genotype, respectively. The solid boxes correspond to i.i.d. replicates of sets of variables, each associated with a particular individual, or ancestral template, respectively.

Now we present a probabilistic model, *DP-Haplotyper*, for the generation of haplotypes in a population and for the generation of genotypes from these haplotypes. We assume that each

individual's genotype is formed by drawing two random *templates* from an ancestral pool, and that these templates are subject to random perturbation. To model such perturbations we assume that each locus is mutated independently from its ancestral state with the same error rate. Finally, assume that we are given noisy observations of the resulting genotypes. The model is displayed as a graphical model in Figure 3.3.

Let J be an ordered list of loci of interest. For each individual i, denote his/her paternal haplotype by $H_{i_0} := [H_{i_0,1}, \ldots, H_{i_0,J}]$ and maternal haplotype by $H_{i_1} := [H_{i_1,1}, \ldots, H_{i_1,J}]$. We denote a set of ancestral templates by $\mathbf{A} = \{A_1, A_2, \ldots\}$, where $A_k := [A_{k,1}, \ldots, A_{k,J}]$ is a particular member of this set. The set $\mathbf{A}$ is a random variable whose cardinality and composition are not fixed, but rather vary with realizations of the Dirichlet process and vary with the observed data.

In our framework, the probability distribution of the haplotype variable H_{i_t}, where the subsubscript $t \in \{0, 1\}$ indexes paternal or maternal origin, is modeled by a mixture model with an unspecified number of mixture components, each corresponding to an equivalence class associated with a particular ancestor. For each individual i, we define the equivalence class variables C_{i_0} and C_{i_1} for the paternal and maternal haplotypes, respectively, to specify the ancestral origin of the corresponding haplotype. The C_{i_t} are the random variables corresponding to the equivalence classes of the Dirichlet process. The base measure G of the Dirichlet process is a joint measure on ancestral haplotypes A and mutation parameters θ, where the latter captures the probability that an allele at a locus is identical to the ancestor at this locus. We let $G(A, \theta) = p(A)p(\theta)$, and we assume that $p(A)$ is a uniform distribution over all possible haplotypes. We let $p(\theta)$ be a beta distribution, $\mathrm{Beta}(\alpha_h, \beta_h)$, and we choose a small value for $\beta_h/(\alpha_h + \beta_h)$, corresponding to a prior expectation of a low mutation rate.

Given C_{i_t} and a set of ancestral templates, we define the conditional probability of the corresponding haplotype instance $h := [h_1, \ldots, h_J]$ to be:

$$\begin{aligned} p(H_{i_t} = h | C_{i_t} = k, \mathbf{A} = \mathbf{a}, \boldsymbol{\theta}) &= p(H_{i_t} = h | A_k = a, \theta_k = \theta) \\ &= \prod_j p(h_j | a_j, \theta), \end{aligned} \tag{3.6}$$

where $p(h_j|a_j,\theta)$ is the probability of having allele h_j at locus j given its ancestor. Eq. (3.6) assumes that each locus is mutated independently with the same error rate. For haplotypes, $H_{i_t,j}$ takes values from a set B of alleles. We use the following *single-locus mutation model*:

$$p(h_j|a_j,\theta) = \theta^{\mathbb{I}(h_j=a_j)}\Big(\frac{1-\theta}{|B|-1}\Big)^{\mathbb{I}(h_j\neq a_j)} \tag{3.7}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

The joint conditional distribution of haplotype instances $\mathbf{h} = \{h_{i_t} : t \in \{0,1\}, i \in \{1,2,\ldots,I\}\}$ and parameter instances $\boldsymbol{\theta} = \{\theta_1,\ldots,\theta_K\}$, given the ancestor indicator $\mathbf{c}$ of haplotype instances and the set of ancestors $\mathbf{a} = \{a_1,\ldots,a_K\}$, can be written explicitly as:

$$p(\mathbf{h},\boldsymbol{\theta}|\mathbf{c},\mathbf{a}) \propto \prod_k \theta_k^{m_k+\alpha_h-1}\Big(\frac{1-\theta_k}{|B|-1}\Big)^{m'_k}\big[1-\theta_k\big]^{\beta_h-1} \tag{3.8}$$

where $m_k = \sum_j\sum_i\sum_t \mathbb{I}(h_{i_t,j} = a_{k,j})\mathbb{I}(c_{i_t} = k)$ is the number of alleles that were not mutated with respect to the ancestral allele, and $m'_k = \sum_j\sum_i\sum_t \mathbb{I}(h_{i_t,j} \neq a_{k,j})\mathbb{I}(c_{i_t} = k)$ is the number of mutated alleles. The count $\mathbf{m}_k = \{m_k, m'_k\}$ is a sufficient statistic for the parameter θ_k and the count $\mathbf{m} = \{\mathbf{m}_k, \mathbf{m}'_k\}$ is a sufficient statistic for the parameter $\boldsymbol{\theta}$. The marginal conditional distribution of haplotype instances can be obtained by integrating out θ in Eq. (3.8):

$$p(\mathbf{h}|\mathbf{c},\mathbf{a}) = \prod_k R(\alpha_h,\beta_h)\frac{\Gamma(\alpha_h+m_k)\Gamma(\beta_h+m'_k)}{\Gamma(\alpha_h+\beta_h+m_k+m'_k)}\Big(\frac{1}{|B|-1}\Big)^{m'_k} \tag{3.9}$$

where $\Gamma(\cdot)$ is the gamma function, and $R(\alpha_h,\beta_h) = \frac{\Gamma(\alpha_h+\beta_h)}{\Gamma(\alpha_h)\Gamma(\beta_h)}$ is the normalization constant associated with $\mathrm{Beta}(\alpha_h,\beta_h)$. (For simplicity, we use the abbreviation R_h for $R(\alpha_h,\beta_h)$ in the sequel).

We now introduce a *noisy observation model* for the genotypes. We let $G_i = [G_{i,1},\ldots,G_{i,J}]$ denote the *joint genotype* of individual i at loci $[1,\ldots,J]$, where each $G_{i,j}$ denotes the genotype at locus j. We assume that the observed genotype at a locus is determined by the paternal and maternal alleles of this locus as follows:

$$p(g_{i,j}|h_{i_0,j},h_{i_1,j},\gamma) \quad = \quad \gamma^{\mathbb{I}(h_{i,j}=g_{i,j})}[\mu_1(1-\gamma)]^{\mathbb{I}(h_{i,j}\overset{1}{\neq}g_{i,j})}[\mu_2(1-\gamma)]^{\mathbb{I}(h_{i,j}\overset{2}{\neq}g_{i,j})}$$

where $h_{i,j} \triangleq h_{i_0,j} \oplus h_{i_1,j}$ denotes the unordered pair of two actual SNP allele instances at locus j; "$\overset{1}{\neq}$" denotes set difference by exactly one element (i.e., the observed genotype is heterozygous,

while the true one is homozygous); "$\overset{2}{\neq}$" denotes set difference of both elements (i.e., the observed and true genotypes are different and both are homozygous); and μ_1 and μ_2 are appropriately defined normalizing constants [3]. We place a beta prior $\mathrm{Beta}(\alpha_g, \beta_g)$ on γ. Assuming independent and identical error models for each locus, the joint conditional probability of the entire genotype observation $\mathbf{g} = \{g_i \,:\, i \in \{1, 2, \ldots, I\}\}$ and parameter γ, given all haplotype instances is:

$$\begin{aligned} p(\mathbf{g}, \gamma|\mathbf{h}) &= \prod_i p(g_i, \gamma|h_{i_0}, h_{i_1}) \\ &= \big[\gamma\big]^{u+\alpha_g-1}\big[\mu_1(1-\gamma)\big]^{u'}\big[\mu_2(1-\gamma)\big]^{u''}\big[1-\gamma\big]^{\beta_g-1} \\ &= \gamma^{\alpha_g+u-1}\big[1-\gamma\big]^{\beta_g+u'+u''-1}\mu_1^{u'}\mu_2^{u''}, \end{aligned} \tag{3.10}$$

where the sufficient statistics $\mathbf{u} = \{u, u', u''\}$ are computed as $u = \sum_{i,j} \mathbb{I}(h_{i,j} = g_{i,j})$, $u' = \sum_{i,j} \mathbb{I}(h_{i,j} \overset{1}{\neq} g_{i,j})$, and $u'' = \sum_{i,j} \mathbb{I}(h_{i,j} \overset{2}{\neq} g_{i,j})$, respectively. Note that $u + u' + u'' = IJ$. To reflect an assumption that the observation error rate is low we set $\beta_g/(\alpha_g + \beta_g)$ to a small constant (0.001). Again, the marginal conditional distribution of $\mathbf{g}$ is computed by integrating out γ.

Having described the Bayesian haplotype model, the problem of phasing individual haplotypes and estimating the size and configuration of the latent ancestral pool can be solved via posterior inference given the genotype data. In Chapter 5, we describe Markov chain Monte Carlo (MCMC) algorithms for this purpose.

3.3.3 Haplotype Modeling Given Partial Pedigree

For diploid organisms such as humans, a subject has two physical copies of each chromosome in his/her somatic cells, which carry the two haplotypes of the SNP sequence in a specific region. When an offspring is to be produced, each of the parents donates a haploid *gamete* (i.e., a sperm for the male and an egg for the female), which carries only one of the two copies of every chromosome

[3] For simplicity, we may let $\mu_1 = \mu_2 = 1/V$, where V is the total number of ways a single SNP haplotype $h_{i,j}$ and a single SNP genotype $g_{i,j}$ can differ (i.e., 2 for binary SNPs). When different μ_1 and μ_2 are desired to penalize single- and double-disagreement differently, one must be careful to treat the case of homozygous $h_{i,j}$ and heterozygous $h_{i,j}$ differently, because they are related to noisy genotype observations in different manners. For example, a heterozygous $h_{i,j}$ (e.g., 01) cannot be related to any genotype with a double disagreement, whereas a homozygous $h_{i,j}$ (e.g., 00) can (e.g., w.r.t. $g_{i,j} = 11$).

of a parent (i.e., one of the two haplotypes). The two gametes of opposite sex then fuse (after mating) to produce a diploid fertilized egg and re-pair the paternal and maternal copies of the chromosome (and therefore, their respective associated haplotypes). The fertilized egg eventually grow into an adult offspring which can be typed.

When the parent-offspring triplet (or even other close biological relatives) are (geno)typed, the ambiguity of haplotypes of an individual can sometimes be resolved by exploiting the dependencies among the haplotypes of family members induced by genetic inheritance and segregation just described. For example, if both parents are homozygous, i.e., $g_1 = a \oplus a$, $g_0 = b \oplus b$, and the offspring are heterogeneous, i.e., $g_{\lambda_{10}} = a \oplus b$, where λ_{10} denotes the offspring of subjects "1" and "0", then we can infer that the haplotypes of the offspring are $h_{\lambda_{10}} = (a, b)$. This special case of triplet genotypes is regarded as *fully informative*. Clearly, not all genotypes are fully informative, and inheritance of haplotypes may be more than mere faithful copying. In particular, chromosomal inheritance could be accompanied by single-generation mutations, which alter single or multiple SNPs on the chromosomes; and recombinations, which disrupt and recombine some chromosome pairs in gamete donors to generate novel (i.e., mosaic) haplotypes. Although genotypes of this nature do not directly lead to full resolution of each individual's haplotypes, undoubtedly the strong dependencies that exist among the genotype data (in contrast to the *iid* genotypes we studied in the last section) could be exploited to reduce the ambiguity of the phasing.

Given the genotypes from a population and partial pedigrees that relate members of various subsets of a population, in order to apply the pedigree constraints in haplotype inference, we need to introduce a few new ingredients into the basic DP-haplotyper model described in the last section to model the distribution of individual haplotypes in a population consisting of now partially coupled (rather than conditionally independent) individuals (Fig. 3.4). We refer to this expanded model as the *Pedi-haplotyper* model.

Formally, we introduce a segregation random variable, $S_{i_t,j}$, for each one of the two SNP alleles of each locus of an individual, to indicate its meiotic origin (i.e., from which one of the two SNP alleles of a parent it is inherited). For example, $S_{i_t,j} = 1$ indicates that allele $H_{i_t,j}$ is inherited from

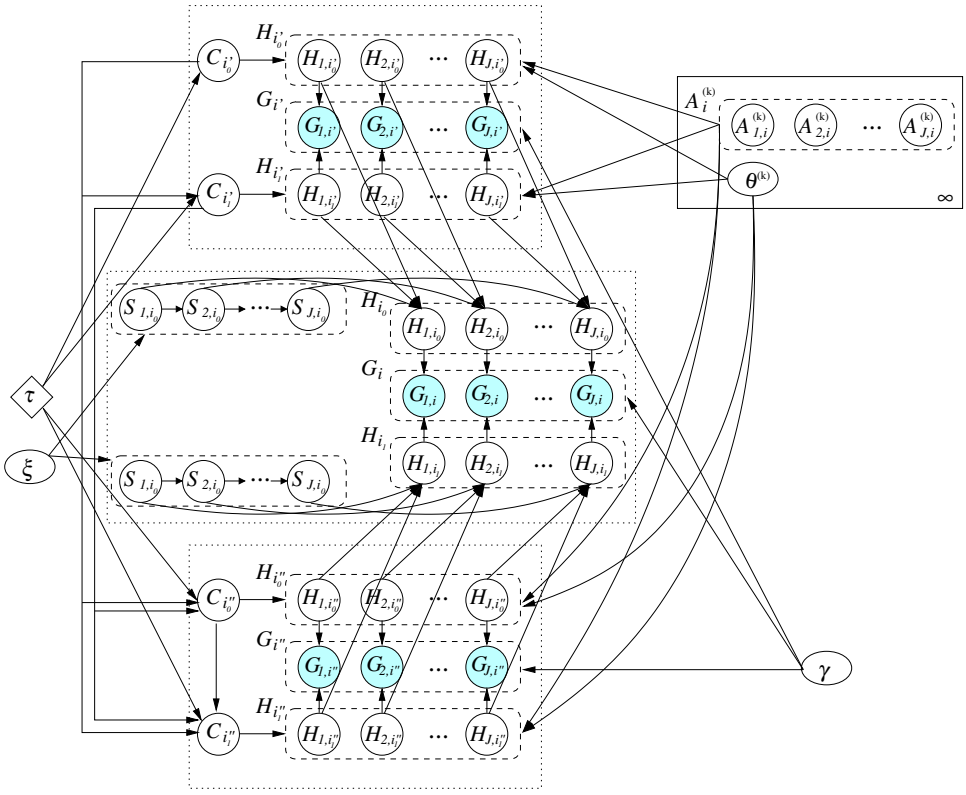


Figure 3.4: The graphical model representation of the Pedi-haplotyper model.

the maternal allele of individual i's t-parent (where $t = 0$ means father and $t = 1$ means mother). We denote the t-parent of individual i by $\pi(i_t)$, and his/her paternal (resp. maternal) allele by $\pi_0(i_t)$ (resp. $\pi_1(i_t)$). We use the following conditional distribution to model possible mutation during single generation inheritance.

$$p(h_{i_t,j}|s_{i_t,j} = r, h_{\pi_0(i_t),j}, h_{\pi_1(i_t),j}, \epsilon_t) = \big[\epsilon_t\big]^{\mathbb{I}(h_{i_t,j}=h_{\pi_r(i_t),j})}\big[\frac{1-\epsilon_t}{|B|-1}\big]^{\mathbb{I}(h_{i_t,j}\neq h_{\pi_r(i_t),j})} \quad (3.11)$$

where $1 - \epsilon_t$ is the mutation rate during inheritance, and $r \in \{0, 1\}$ represents the choice of the paternal or maternal alleles of a parent subject by an offspring. Note that this *single generation inheritance model* allows different mutational rates for the parental and maternal alleles if desired (e.g., to reflect the difference in gamete environment in a male or a female body), by letting ϵ_0 and ϵ_1 take different values, or giving them different beta prior distributions in case we want to model uncertainty of ϵ_t in a Bayesian framework.

To model possible recombination events during single generation inheritance, we assume that the list of segregation random variables, $[S_{i_t,1}, \ldots, S_{i_t,J}]$, associated with individual haplotype H_{i_t}, forms a 1st-order Markov chain, with transition matrix $\boldsymbol{\xi}$:

$$\begin{aligned} p(S_{i_t,j+1} = r'|S_{i_t,j} = r) &= \xi_{rr'} \\ &= \big[\xi\big]^{\mathbb{I}(r=r')}\big[1-\xi\big]^{\mathbb{I}(r\neq r')}, \end{aligned} \quad (3.12)$$

where $1 - \xi$ is the probability of a recombination event (i.e., a swap of parental origin) at position j. This model is equivalent to assuming that the recombination events follow a Poisson point process of rate ξ along the chromosome. If desired, a beta prior $Beta(\alpha_s, \beta_s)$ can be introduced for ξ. Again, the recombination rates in males and females can be different if desired.

Looking back to the overall graphical topology of the Pedi-haplotyper model, as illustrated in Figure 3.4, for founding members in the pedigree (i.e., those without parental information), or half founding members (i.e., those with information from only one of the two parents), we assume that their un-progenitored haplotype(s) are inherited from some ancestors, thus following the basic haplotype model described in §3.3. For the haplotypes of the offspring in the pedigree, we couple

them to their parents using the single generation mutation and recombination model described in the previous paragraphs. Thus, the Pedi-haplotyper model proposed in this section is fully generalizable to any pedigree structure.

Solving the Pedi-haplotyper model is slightly more difficult than for the basic Dirichlet process mixture model, DP-haplotyper, for *iid* populations. But as we show in Chapter 5, most of the methods we developed for the DP-haplotyper can be directly used in this more elaborate framework, with the addition of a few new sampling steps for the newly introduced random variables.

3.4 Experimental Results

We validated our algorithm by applying it to simulated and real data and compared its performance to that of the state-of-the-art PHASE algorithm [Stephens *et al.*, 2001] and other current algorithms. We report on the results of both variants of our algorithm: the Gibbs sampler, denoted DP(Gibbs), and the Metropolis-Hasting sampler, denoted DP(MH). Throughout the experiments, we set the hyperparameter τ in the Dirichlet process to be roughly 1% of the population size, i.e., for a data set of 100 individuals, $\tau = 1$. We used a burn-in of 2000 iterations (or 4000 for datasets with more than 50 individuals), and used the next 6000 iterations for estimation.

3.4.1 Simulated Data

In our first set of experiments we applied our method to simulated data ("short sequence data") from Stephens *et al.* [2001]. This data contains sets of $2n$ haplotypes, randomly paired to form n genotypes, under an infinite-sites model with parameters $\eta = 4$ and $R = 4$ determining the mutation and recombination rates, respectively. We used the first 40 datasets for each combination of individuals and sites, where the number of individuals ranged between 10 and 50, and the number of sites ranged between 5 and 30.

To evaluate the performance of the algorithms we used the following error measures: err_s, the ratio of incorrectly phased SNP sites over all non-trivial heterozygous SNPs (excluding individuals with a single heterozygous SNP); err_i, the the ratio of incorrectly phased individuals over all

	DP(MH)			PHASE			EM
#individuals	err_s	err_i	d_s	err_s	err_i	d_s	err_i
10	0.060	0.216	0.051	0.046	0.182	0.054	0.424
20	0.039	0.152	0.039	0.029	0.136	0.046	0.296
30	0.036	0.121	0.038	0.024	0.101	0.027	0.231
40	0.030	0.094	0.029	0.019	0.071	0.026	0.195
50	0.028	0.082	0.024	0.019	0.072	0.025	0.167

Table 3.1: Performance on data from Stephens *et al.* [2001]. The results for the EM algorithm are adapted from Stephens *et al.* [2001].

non-trivial heterogeneous individuals; and d_s, the *switch distance*, which is the number of phase flips required to correct the predicted haplotypes over the total number of non-trivial heterogeneous SNPs. The results are summarized in Table 3.1. Overall, we perform slightly worse than PHASE on the first two measures, and slightly better on the switch distance measure (which uses 100,000 sampling steps). Both algorithms provide a substantial improvement over EM.

		DP(Gibbs)			DP(MH)			PHASE			HAP	HAPLOTYPER
block id.	length	err_s	err_i	d_s	err_s	err_i	d_s	err_s	err_i	d_s	err_s	err_s
1	14	0.223	0.485	0.229	0	0	0	0.003	0.030	0.003	0.007	0.039
2	5	0	0	0	0.007	0.026	0.007	0.007	0.026	0.007	0.036	0.065
3	5	0	0	0	0	0	0	0	0	0	0	0.008
4	11	0.143	0.262	0.128	0	0	0	0	0	0	0.015	-
5	9	0.020	0.066	0.020	0.011	0.033	0.011	0.011	0.033	0.011	0.027	0.151
6	27	0.071	0.191	0.074	0.005	0.043	0.005	0	0	0	0.018	0.041
7	7	0.005	0.018	0.005	0.005	0.018	0.005	0.005	0.018	0.005	0.068	0.214
8	4	0	0	0	0	0	0	0	0	0	0	0.252
9	5	0.029	0.097	0.029	0.012	0.032	0.012	0.012	0.032	0.012	0.057	0.152
10	4	0.007	0.025	0.007	0.007	0.025	0.007	0.008	0.025	0.008	0.042	0.056
11	7	0.010	0.034	0.005	0.005	0.017	0.005	0.011	0.034	0.011	0.033	0.093
12	5	0.010	0.037	0.020	0	0	0	0	0	0	0	0.077

Table 3.2: Performance on the data of Daly *et al.* [2001], using the block structure provided by Halperin and Eskin [2002]. The results of HAP and HAPLOTYPER are adapted from Halperin and Eskin [2002]. Since the error rate in Halperin and Eskin [2002] uses the number of both heterozygous and missing sites as the denominator, whereas we used only the non-trivial heterozygous ones, we rescaled the error rates of the two latter methods to be comparable to ours.

3.4.2 Real Data

We applied our algorithm to two real datasets and compared its performance to that of PHASE [Stephens *et al.*, 2001] and other algorithms.

The first dataset contains the genotypes of 129 individuals over 103 polymorphic sites [Daly *et al.*, 2001]. In addition it contains the genotypes of the parents of each individual, which allows the

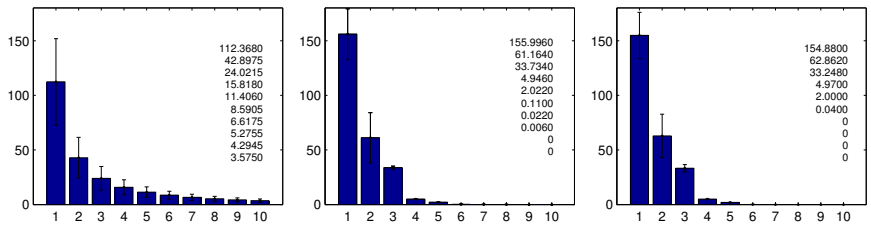


Figure 3.5: The top ten ancestral templates during Metropolis-Hasting sampling for block 1 of the data of [Daly *et al.*, 2001]. (The numbers in the panels are the posterior means of the frequencies of each template). (a) Immediately after burn-in (first 2000 samples). (b) 3000 samples after burn-in. (c) 6000 samples after burn-in.

inference of a large portion of the haplotypes as in Eskin *et al.* [2003]. The results are summarized in Table 3.2. It is apparent that the Metropolis-Hasting sampling algorithm significantly outperforms the Gibbs sampler, and is to be preferred given the relatively limited number of sampling steps ($\sim$ 6000). The overall performance is comparable to that of PHASE and better than both HAP [Halperin and Eskin, 2002; Eskin *et al.*, 2003] and HAPLOTYPER [Niu *et al.*, 2002].

It is important to emphasize that our methods also provide *a posteriori* estimates of the ancestral pool of haplotype templates and their frequencies. We omit a listing of these haplotypes, but provide an illustrative summary of the evolution of these estimates during sampling (Figure 3.5).

The second dataset contains genotype data from four populations, 90 individuals each, across several genomic regions [Gabriel *et al.*, 2002]. We focused on the Yoruban population (D), which contains 30 trios of genotypes (allowing us to infer most of the true haplotypes) and analyzed the genotypes of 28 individuals over four medium-sized regions (see below). The results are summarized in Table 3.3. All methods yield higher error rates on these data, compared to the analysis of the data of Daly *et al.* [2001], presumably due to the low sample size. In this setting, over all but one of the four regions, our algorithm outperformed PHASE for all three types of error measures. A preliminary analysis suggests that our performance gain may be due to the bias toward parsimony induced by the Dirichlet process prior. We found that the number of template haplotypes inferred in our algorithm is typically small, whereas in PHASE, the hypothesized haplotype pool can be very large (i.e., region 7b has 83 haplotypes, compared to 10 templates in our case and 28 individuals overall).

		DP(MH)			PHASE		
region	length	err_s	err_i	d_s	err_s	err_i	d_s
16a	13	0.185	0.480	0.141	0.174	0.440	0.130
1b	16	0.100	0.250	0.160	0.200	0.450	0.180
25a	14	0.135	0.353	0.115	0.212	0.588	0.212
7b	13	0.105	0.278	0.066	0.145	0.444	0.092

Table 3.3: Performance on the data of Gabriel *et al.* [2002].

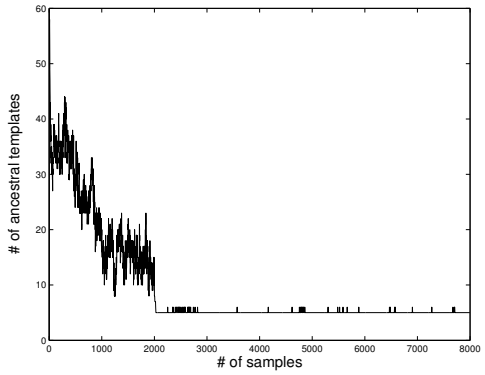


Figure 3.6: Sampling trace of the number of population haplotypes derived from the genotypes. As can be seen, the Markov chain starts from a rather non-parsimonious estimation, and converges to a parsimonious solution after about two thousand samples.

In terms of computational efficiency, we noticed that PHASE typically required 20,000 to 100,000 steps until convergence, while our DP-based method required around 2,000∼6,000 steps to convergence (Fig. 3.6).

3.5 Conclusions and Discussions

In this chapter, we have proposed a Bayesian approach to the modeling of genotypes based on a Dirichlet process prior. We have shown that the Dirichlet process provides a natural representation of uncertainty regarding the size and composition of the pool of haplotypes underlying a population. We will present in Chapter 5 several Markov chain Monte Carlo algorithms for haplotype inference under either a basic DP mixture haplotype model intended for an *iid* population, or, an extended graphical DP mixture model — Pedi-haplotyper model — for a population containing both *iid* subjects and subjects coupled by partial pedigrees. The experiments on the basic DP mixture haplotype model show that this model leads to effective inference procedures for inferring the ancestral pool

and for haplotype phasing based on a set of genotypes. The model accommodates growing data collections and noisy and/or incomplete observations. The approach also naturally imposes an implicit bias toward small ancestral pools during inference, reminiscent of parsimony methods, doing so in a well-founded statistical framework that permits errors.

Our focus here has been on adapting the technology of the Dirichlet process to the setting of the standard haplotype phasing problem. But an important underlying motivation for our work, and a general motivation for pursuing probabilistic approaches to genomic inference problems, is the potential value of our model as a building block for more expressive models. In particular, as in Greenspan and Geiger [2003] and Lauritzen and Sheehan [2002], the graphical model formalism naturally accommodates various extensions, such as segmentation of chromosomes into haplotype blocks and the inclusion of pedigree relationships. In section §3.4, we have outlined a preliminary extension of the basic Dirichlet process mixture model that incorporates pedigree relationships and briefly discussed how to model realistic biological processes that might influence haplotype formation and diversification, such as recombination and mutation during single generation inheritance. We recognize that many other important issues also deserve careful attention, for example, haplotype recombinations among the ancestral haplotype pools (so far, we assume that these ancestral haplotypes relate to modern individual haplotypes only via mutations), aspects of evolutionary dynamics (e.g., coalescence, selection, etc.), and linkage analysis under joint modeling of complex traits and haplotypes. We believe that the graphical model formalism we proposed can readily accommodate such extensions. In particular, it appears reasonable to employ an ancestral recombination hypothesis (rather than single generation recombination) to account for common individual haplotypes that are distant from any single ancestral haplotype template, but can be matched piecewise to multiple ancestral haplotypes. This may be an important aspect of chromosomal evolution and can provide valuable insight into the dynamics of populational genetics in addition to point-mutation-based coalescence theory, and can potentially improve efficiency and quality of haplotype inference.

The Dirichlet process parameterization also provides a natural upgrade path for the consideration of richer models; in particular, it is possible to incorporate more elaborate base measures G into the Dirichlet process framework—the coalescence-based distribution of Stephens *et al.* [2001] would be an interesting choice. In Chapter 5, while developing MCMC algorithms for haplotype inference, we will also briefly discuss a heuristic for constructing an informative base measure for the DP using low-quality but inexpensive haplotype information (e.g., that obtained from a conventional EM algorithm). Note that the partition structure of the Dirichlet process is equivalent to that induced by the Ewens sampling formula (ESF) [Tavare and Ewens, 1998] known to the population genetics community. The ESF represents a non-Darwinian theory of evolution which claims that "the extensive genetic variation observed in natural populations is, on the whole, not due to natural selection, but arises rather as a result of purely stochastic changes in gene (allele) frequencies in a finite population" [Tavare and Ewens, 1998]. The fact that our DP mixture model performed adequately in a number of problems suggests that such non-Darwinian evolution may apply to SNP distribution, which is interesting, yet would appear paradoxical, if we proceed to use haplotypes to map clearly non-neutral genes (say, those that relate to biological disorders) via linkage disequilibrium.

Chapter 4

Probabilistic Inference I: Deterministic Algorithms

The Bayesian graphical models presented in the last two chapters both define high-dimensional, hybrid probability distributions for which important statistical queries may be difficult to compute. For example, in the **LOGOS** model, the sequence variable Y_t at site t of a study sequence depends on all the motif parameters $\{\theta_l^{(k)} \,|\forall l, k\}$, each of which in turn depends on one of the PSMD prototype (i.e., Dirichlet component) indicators $\{S_l^{(k)} \,|\forall l, k\}$ coupled by a first-order Markov chain. Thus, to compute the posterior probability distributions $p(x_t|\mathbf{y})$ and $p(\theta_l^{(k)}|\mathbf{y})$ for MAP prediction of motif locations and Bayesian estimation of motif PWMs, one has to integrate over the Cartesian product of a continuous state space for the PWMs and the discrete spaces for the PSMD prototype (denoted as $\mathbb{D}$) and for the sequence annotation indicators (denoted as $\mathbb{S}$). The complexity of such a state space is on the order of

$$\mathbb{R}^{4\times\sum_k L_k} \times |\mathbb{D}|^{\sum_k L_k} \times |\mathbb{S}|^T,$$

which translates to $O(\mathbb{R}^{120} \times 10^{1000})$ for a 1000 bp sequence harboring only two possible motif patterns each of length 15 bp. Clearly, this computation is in general intractable with any off-the-shelf exact algorithm and some approximation scheme is necessary. In this chapter, we present a general variational approach for computing deterministic approximations to such intractable distributions. In the next chapter, we briefly discuss stochastic approximation methods based on sampling. Some of the materials covered in this chapter have appeared in [Xing *et al.*, 2003b;

Xing *et al.*, 2004a].

4.1 Background

For a multivariate probability distribution $p(\mathbf{x}_H, \mathbf{x}_E)$, where $\mathbf{X}_H$ and $\mathbf{X}_E$ denote the sets of all unobserved (i.e., hidden) and observed (i.e., evidence) variables, respectively (and, following convention, their lower case counterparts denote states or values of the corresponding variables), the general problem of probabilistic inference is that of computing the conditional probabilities $p(\mathbf{x}_F|\mathbf{x}_E)$, where $F \subseteq H$ is the index set of an arbitrary subset of hidden variables.

Probabilistic inference techniques play an important role in any probabilistic methodology for prediction and learning. For example, probabilistic prediction of unobserved events or patterns in real world tasks such as weather forecasting, text segmentation and tagging, robot localization, image analysis, filtering and smoothing of sequential data streams, and various computational biology problems such as motif, haplotype and pedigree inference considered in this thesis, all involve performing probabilistic inference on a domain-specific, high-dimensional, and often hybrid (i.e., comprising both discrete and continuous variables) probability model. Probabilistic inference is also indispensable for the acquisition of probability models from incomplete or partially observed data using statistical learning methods, because many of these methods amount to parameter estimation based on a maximum likelihood or an empirical Bayes principle [Efron, 1996], which employs an inference subroutine to impute the unobserved variable(s) for computing the necessary sufficient statistics.

Solving an inference query can be understood as a *marginalization* computation. To see this, observe that the conditional probability $p(\mathbf{x}_F|\mathbf{x}_E)$ is equal to:

$$p(\mathbf{x}_F|\mathbf{x}_E) = \frac{p(\mathbf{x}_F, \mathbf{x}_E)}{p(\mathbf{x}_E)} = \frac{\sum_{\mathbf{x}_{H\setminus F}} p(\mathbf{x}_{H\setminus F}, \mathbf{x}_F, \mathbf{x}_E)}{\sum_{\mathbf{x}_F} p(\mathbf{x}_F, \mathbf{x}_E)}, \tag{4.1}$$

where the summation (or integration in case of continuous variables) over all possible values of some (or all) hidden variables in the model is called *marginalization*. Typically, an inference query involves computing the conditional probabilities for only small subsets of variables (e.g., that of

singleton hidden variables such as x_t in the **LOGOS** model), and sometimes a large number of such queries need to be processed (e.g., all x_t's for motif detection under **LOGOS**). This is often a computationally expensive operation, as the state space to be swept during marginalization grows exponentially with the number of variables being marginalized. The graphical model formalism provides a systematic and efficient approach to such computation. General exact inference algorithms have been developed, which take advantage of the conditional independencies present in the joint distribution $p(\mathbf{x}_H, \mathbf{x}_E)$, which can be inferred from the pattern of missing edges in the graph, to distribute the high-dimensional combinatorial summation over all hidden variables in a standard marginalization operation into a sequence of low-dimensional local summations each over a (small) subset of hidden variables (Fig. 4.1). We will briefly describe a representative of these algorithms, the junction tree algorithm, in the next section.

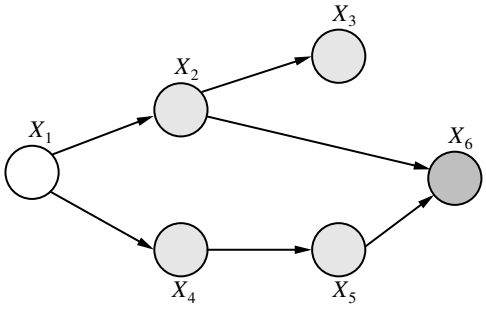


Figure 4.1: Inference on a graphical model. The dark shading indicates the node on which we condition, the unshaded node is the one for which we wish to compute the conditional probability distribution, and the lightly shaded nodes are those that need to be marginalized out in computing the posterior probability $p(x_1|x_6)$. For this graphical model, the summations for computing the joint marginal can be distributed to subsets of variables in the following way (formally known as an *elimination* algorithm): $p(x_1, x_6) = \sum_{x_2,x_3,x_4,x_5} p(x_1)p(x_2|x_1)p(x_4|x_1)p(x_3|x_2)p(x_5|x_4)p(x_6|x_2, x_5) = p(x_1) \sum_{x_2} p(x_2|x_1) \sum_{x_3} p(x_3|x_2) \sum_{x_4} p(x_4|x_1) \sum_{x_5} p(x_5|x_4)p(x_6|x_2, x_5)$
.

Although there are many cases in which the exact algorithms provide a satisfactory solution to the inference and learning problems, large-scale probability models arising from complex real world domains have outgrown the ability of current (and probably future) exact inference algorithms to compute marginals and learn parameters. This is particularly true for models we developed in this dissertation, which concern complex gene regulation elements and genetic polymorphism patterns in the genomic sequences. As illustrated at the beginning of this chapter, the time and

space complexity of the exact algorithms is unacceptable and it is necessary to have recourse to approximation procedures.

For this reason, the development of efficient and broadly applicable approximation algorithms for probabilistic inference is critical to further progress. Two commonly used approximation techniques are *Monte Carlo* methods (such as Markov chain Monte Carlo, or MCMC) and *variational methods*. MCMC techniques are asymptotically exact and easy to apply. The BUGS system uses MCMC within a general-purpose statistical modeling language (see [Gilks *et al.*, 1996]), and the inference process can be set up automatically for a variety of models. Unfortunately, MCMC often converges very slowly. Variational methods, on the other hand, are claimed to exhibit fast convergence and (in some cases) give a deterministic lower bound on the true likelihood. The original belief propagation (BP) method [Pearl, 1988] is now understood as a variational algorithm [Yedidia *et al.*, 2001b] that (if it converges) calculates an optimal approximation to the true posterior distribution among those approximate distributions that include only *pairwise* dependencies among variables. BP can be applied straightforwardly to a wide range of probability models and it has been used for biological classification/clustering problems expressed as complex graphical models [Segal *et al.*, 2001]. A generalized BP (GBP) algorithm can be derived that operates with dependency structures on larger clusters of variables and often gives more accurate results [Yedidia *et al.*, 2001a] [1]. Like BP, GBP sometimes fails to converge. It may also fail to give a lower bound on the true likelihood due to the use of an *ad hoc* approximation to the intractable entropy term in the objective functional it optimizes (to be detailed in §4.3.4.3). Other variational approximation methods based on structured mean field approximation have been developed that are guaranteed to converge to lower bounds on the true likelihood (see, e.g., [Jordan *et al.*, 1999]), but these methods often require model-specific derivation of iteration equations.

In this chapter, we develop a generalized mean field (GMF) theory which leads to a generic variational inference algorithm that is straightforwardly applicable to a wide range of models and is guaranteed to converge to a lower bound on the true likelihood. Given an arbitrary decomposition of

[1] Similar techniques called cluster variational methods (CVMs) have also been developed in the statistical physics community [Kappen and Wiegerinck, 2002].

the original model into disjoint clusters of variables, the algorithm computes the posterior marginal for each cluster given its own evidence and the *expected sufficient statistics*, obtained from its neighboring clusters, of the variables in the cluster's Markov blanket. Optimal clustering of the variables can be obtained in a principled fashion via a graph partition algorithm. The algorithm operates in an iterative, message-passing style until a fixed point is reached. We show that the cluster marginals retain exactly the intra-cluster dependencies of the original model, which means that the inference problem within each cluster can be solved independently of the other clusters (given the Markov blanket messages) by any inference method. This GMF algorithm is applied to the Bayesian motif prediction and learning problem under the **LOGOS** model and shows significant improvement over a sampling-based approach (discussed in the next chapter).

4.1.1 Notation

Before starting the technical sections, here is a summary of some necessary notations and definitions needed in our exposition.

We consider a graph (directed or undirected) $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes (vertices) and $\mathcal{E}$ the set of edges (links) of the graph. Let X_n denote the random variable associated with node n, for $n \in \mathcal{V}$; let $\mathbf{X}_C$ denote the subset of variables associated with a subset of nodes C, for $C \subseteq \mathcal{V}$, and let $\mathbf{X} = \mathbf{X}_\mathcal{V}$ denote the collection of all variables associated with the nodes of the graph. We use upper-case X (resp. $\mathbf{X}$) to denote a random variable (resp. variable set), and lower-case x (resp. $\mathbf{x}$) to denote a certain state (or value, configuration, etc.) taken by the corresponding variable (resp. variable set). We refer to a graph $H = (\mathcal{V}, \mathcal{E}')$, where $\mathcal{E}' \subseteq \mathcal{E}$, as a *subgraph* of G. We use $\mathcal{C} = \{C_1, C_2, \ldots, C_I\}$ to denote a disjoint partition (or, a *clustering*) of all nodes in graph G, where C_i refers to the set of indices of nodes in cluster i; likewise, $\mathcal{D} = \{D_1, D_2, \ldots, D_K\}$ denotes a set of *cliques* (i.e., completely connected subsets of nodes) of G. For a given clustering, we define the *border clique set* $\mathcal{B}_i$ as the set of cliques that intersect with but are not contained in cluster i; and the *neighbor cluster set* $\mathcal{N}_i$ as the set of clusters that contain nodes connected to nodes in cluster i. For undirected graphs, the *Markov blanket* of a cluster i ($\mathcal{MB}_i$) is the set of all nodes

outside C_i that connect to some node in C_i, and, for directed graphs, the Markov blanket is the set of all nodes outside C_i that are parents, children, or co-parents (other than those already in C_i) [2] of some node in C_i (Fig. 4.2). Clusters that intersect with $\mathcal{MB}_i$ are called the *Markov blanket clusters* ($\mathcal{MBC}_i$) of C_i.

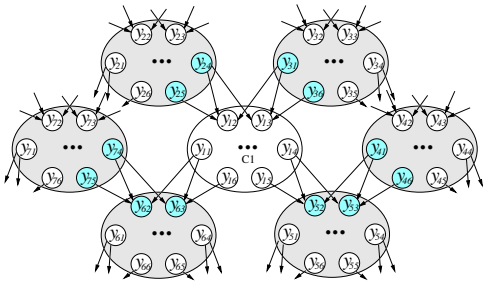


Figure 4.2: The Markov blanket $\mathcal{MB}_1$ (blue-shaded nodes) of cluster 1 in a directed graph. Shaded blobs constitute $\mathcal{MBC}_1$.

4.2 Exact Inference Algorithms

In this section, we give a brief overview of the junction tree algorithm [S. Lauritzen, 1988]. It is a general purpose algorithm which subsumes many other exact inference inference algorithms (e.g., belief propagation for tree models [Pearl, 1988], the forward-backward algorithms for HMMs [Rabiner and Juang, 1986], the peeling algorithm for pedigree models [Thompson, 1981], etc.) as special cases.

4.2.1 The Junction Tree Algorithm

As described in Chapter 1, for a directed graphical model $G(\mathcal{V}, \mathcal{E})$, the joint probability distribution for all the $|\mathcal{V}|$ nodes in the graph can be written as the product of all *local conditional distributions* defined on each node and its parent(s):

$$p(\mathbf{x}) = \prod_{i=1}^{|\mathcal{V}|} p(x_i|\mathbf{x}_{\pi_i}). \tag{4.2}$$

[2] A co-parent of a node, say, X_v, is defined as the parent (other than X_v) of a child node X_u of X_v.

For an undirected graphical model, the joint probability distribution is equal to the product of the *potential functions* associated with each clique of the graph, up to a normalization constant:

$$p(\mathbf{x}) = \frac{\prod_{\alpha=1}^{|\mathcal{D}|} \phi_\alpha(\mathbf{x}_{D_\alpha})}{Z}, \tag{4.3}$$

where $Z = \sum_x \prod_{\alpha=1}^{|\mathcal{D}|} \phi_\alpha(\mathbf{x}_{D_\alpha})$ is referred to as a "partition function".

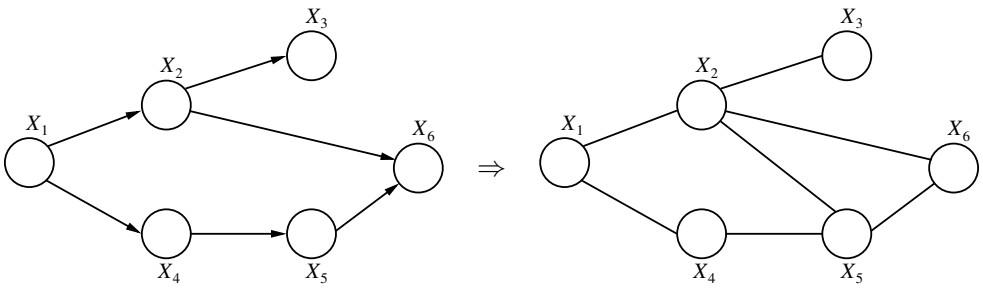


Figure 4.3: Moralization of a directed graph.

A directed graphical model can be converted into an equivalent undirected graphical model via an operation called "moralization," which connects all parents of a common child node pairwise with undirected edges, and then drops the directionality of all other edges in the graph (Fig. 4.3). The resulting graph is called a "moral graph," in which all the nodes originally involved in a local conditional distribution in the directed graph now appear together in a common clique. Thus, local conditional distributions in a directed graph can be thought of as normalized potential functions in the corresponding moral graph, and the product rule (i.e., Eq. (4.2) and Eq. (4.3)) of the joint distribution gives the same outcome for the directed model and its undirected counterpart. Due to the equivalence of the undirected moral graph to the original directed graph in representing a joint probability distribution, the junction tree algorithm concerns only undirected graphs.

The junction tree algorithm starts with the moralized graph. It first chooses an *elimination order* for all nodes in the graph, and applies an operation called *triangulation* to this order as follows: 1) choose the next node in the elimination order, 2) add edges to link all remaining pairs of nodes that are neighbors of this node and, 3) remove the node (and all its incident edges) from the graph. Taking the new edges added in this process and adding them to the original moralized graph yield a *triangulated graph* (Fig. 4.4a).

A triangulated graph allows the creation of a data structure known as a *junction tree* (Fig. 4.4b), on which a generalized message-passing algorithm can be defined. A full discussion of the construction of a junction tree is beyond the scope of this thesis; in short, it is a maximal spanning tree of cliques in the triangulated graph, with weights defined by the cardinality of the intersections between cliques. A key property of the junction tree is the so called *running intersection property*, which says that if a node appears in any two cliques in the tree, it appears in all cliques that lie on the path between the two cliques. As a consequence of this property, in a junction tree, local consistency (i.e., potentials of **adjacent** cliques in the tree agree on marginals of any shared variables) implies global consistency (i.e., potentials of **all** cliques in the tree agree on marginals of common variables).

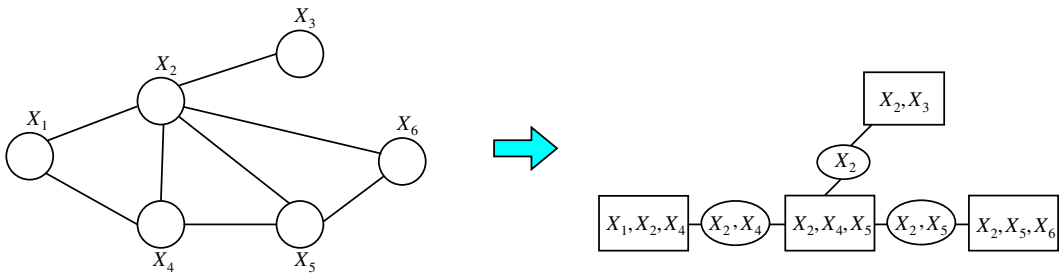


Figure 4.4: Construction of the junction tree. (a) The triangulated graph of the graphical model in Fig. 4.3. (b) The junction tree. Squares represent original cliques in the triangulated graph, ellipsoids represent separators of adjacent cliques.

With the junction tree, the joint probability distribution can now be expressed in the following factored form:

$$p(\mathbf{x}) \quad = \quad \frac{\prod_{C_i \in \mathcal{C}_T} \psi_i(\mathbf{x}_{C_i})}{\prod_{S_j \in \mathcal{S}_T} \phi_j(\mathbf{x}_{S_j})}, \tag{4.4}$$

where $\mathcal{C}_T$ is the set of all cliques in the triangulated graph and $\mathcal{S}_T$ is the set of separators (i.e., clique intersections) spanned by the junction tree.

The clique potentials $\psi(\cdot)$ and separator potentials $\phi(\cdot)$ can be updated by running a message-passing protocol on the junction tree, with the following update rule:

$$\phi_j^*(\mathbf{x}_{S_j}) = \sum_{\mathbf{x}_{C_i \setminus S_j}} \psi_i(\mathbf{x}_{C_i}), \qquad \psi_k^*(\mathbf{x}_{C_k}) = \frac{\phi_j^*(\mathbf{x}_{S_j})}{\phi_j(\mathbf{x}_{S_j})} \psi_k(\mathbf{x}_{C_k}),$$

where $\mathbf{X}_{S_j}$ denotes the set of variables that separates cliques $\mathbf{X}_{C_i}$ and $\mathbf{X}_{C_j}$, and the "message" is now passed from clique i to clique k via separator j (Fig. 4.5). The protocol typically starts by picking a root of the tree, and then first passing messages from root to all leaves along tree branches, and then collecting messages from all leaves to the root, which leads to $\psi_i = p(\mathbf{x}_{C_i})$ and $\phi_j = p(\mathbf{x}_{S_j})$ for all i, j, when the message passing terminates. Note that a single run of the junction tree algorithm yields all clique marginals, not merely that corresponding to a single clique.

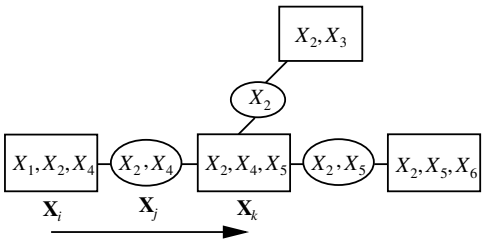


Figure 4.5: Message passing in a junction tree.

It is easy to see that the computational bottleneck of the junction tree algorithm is determined by the size of the maximal clique in the triangulated graph, which is affected by the choice of the elimination order that induces the triangulated graph. The minimum of the maximal clique size among all possible triangulations is know as the *tree width* of the graph. Choosing an elimination order that minimizes the maximal clique size is non-trivial (indeed, it is an NP-hard problem for arbitrary graphs, but can often be effectively approached on special graphs). There are many special-purpose exact inference algorithms for specific families of graphical models (e.g., the forward-backward algorithm for HMMs, Pearl's belief propagation algorithm for trees, etc.), almost all of which are essentially special cases of the the junction tree algorithm applied to special graphs, using a special and often optimal choice of the elimination order for triangulation.

4.3 Approximate Inference Algorithms

As mentioned, for a complex distribution, computing the marginal (or conditional) distributions, as well as the maximum *a posteriori* configurations, of an arbitrary subset of the random variables is intractable. The variational approach to these inference problems involves converting them into an optimization problem, then approximating the feasible set of the solution or the function to be optimized (or both), and solving the relaxed optimization problem. Thus, given a probability distribution $p(\mathbf{x}|\theta)$ that factors according to a graph, the variational methods yield approximations to marginal probabilities via the solution to an optimization problem that generally exploits some of the graphical structure. In the sequel, we describe a general variational principle for inference in probabilistic graphical models, on which a variety of extant deterministic approximate inference techniques are based, and from which we draw the mathematical foundations for the subsequent development of a more general approach for approximate inference called generalized mean field (GMF) inference. We begin with some necessary definitions and algebraic preliminaries.

4.3.1 Cluster-factorizable Potentials

Given a clustering $\mathcal{C}$ of all nodes in $G(\mathcal{V}, \mathcal{E})$, some cliques in $\mathcal{D}$ may intersect with multiple clusters (Fig. 4.6). *Cluster-factorizable potentials* are potential functions which take the form $\phi_\beta(\mathbf{x}_{D_\beta}) = F_\beta(\phi_{\beta_i}(\mathbf{x}_{D_\beta \cap C_i}), \ldots, \phi_{\beta_j}(\mathbf{x}_{D_\beta \cap C_j}))$, where $F(\cdot)$ is a (multiplicatively, or additively) factorizable function over its arguments; i.e., in the case of two clusters, $F(a, b) = a \times b$ or $a + b$. Factorizable potentials are common in many model classes. For example, the classical Ising model is based on singleton and pairwise potentials of the following factorizable form (under the exponential representation, as described shortly): $\phi(x_i) = \theta_i x_i$, $\phi(x_i, x_j) = \theta_{ij} x_i x_j$; higher-order Ising models and many more general discrete models also admit factorizable potentials; conjugate exponential pairs, such as the Dirichlet-multinomial, linear-Gaussian, etc., are also factorizable; finally, for logistic functions and other generalized linear models (GLIMs) that are not directly factorizable, it is often possible to obtain a factorizable variational transformation in the exponential family that

lower bounds the original function [Jaakkola and Jordan, 2000]. In other cases (e.g., tabular potentials over a clustering of variables), a more general treatment based on peripheral marginal potentials can be used (see §4.4.3). We will see that cluster-factorizable potentials allow the decoupling of the computation of expected potentials.

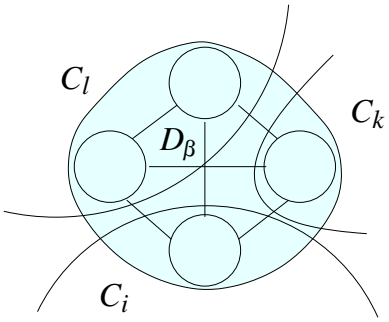


Figure 4.6: A clique D_β intersecting with three clusters $\{C_i, C_j, C_k\}$ in an undirected graph.

4.3.2 Exponential Representations

In order to formulate variational inference as a generic optimization problem, it is convenient to use the following exponential representation for a graphical model.

Similar to the general parameterization of graphical models introduced in Chapter 1, under exponential representations, for undirected graphical models, the family of joint probability distributions associated with a given graph can be parameterized in terms of a set of *potential functions* associated with a set of cliques in the graphs [3]. For a set of cliques $\mathcal{D} = \{D_\alpha | \alpha \in \mathcal{A}\}$ associated with an undirected graph, indexed by a set $\mathcal{A}$, let $\boldsymbol{\phi} = \{\phi_\alpha | \alpha \in \mathcal{A}\}$ denote the set of potential functions defined on the cliques, and $\boldsymbol{\theta} = \{\theta_\alpha | \alpha \in \mathcal{A}\}$ the set of parameters associated with these potential functions (for simplicity, we label ϕ and θ with the corresponding *clique index*, e.g., α, rather than with the clique D_α itself). The family of joint distributions determined by $\boldsymbol{\phi}$ can be

[3] More precisely, these potential functions are now *exponential potential functions* that are semantically different from what we meant by "potential functions" in our early exposition of graphical models. Technically, however, little difference exists in their definitions, except that the range of the exponential potential functions is all real numbers whereas the original potential functions have positive values. For fixed potential weights, there exists a one-to-one correspondence between the two types of potential functions. For simplicity, in the sequel we still use the term "potential functions" in our exposition under the exponential representations.

expressed as follows:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \exp\{\sum_{\alpha \in \mathcal{A}} \theta_\alpha \phi_\alpha(\mathbf{x}_{D_\alpha}) - A(\boldsymbol{\theta})\} \tag{4.5}$$

where $A(\boldsymbol{\theta})$ is the *log partition function*. We also define the *energy*, $E(\mathbf{x}) = -\sum_\alpha \theta_\alpha \phi_\alpha(\mathbf{x}_{D_\alpha})$, for state $\mathbf{x}$.

For directed graphical models, in which the joint probability is defined as $p(\mathbf{x}) = \prod_i p(x_i|\mathbf{x}_{\pi_i})$, we transform the underlying directed graph into a *moral graph*, and set the potential functions $\boldsymbol{\phi}_i(x_i, \mathbf{x}_{\pi_i})$ equal to the logarithms of the local conditional probabilities $p(x_i|\mathbf{x}_{\pi_i})$. In the sequel, we will focus on models based on *conditional exponential families*. That is, the conditional distributions $p(x_i|\mathbf{x}_{\pi_i})$ can be expressed as:

$$p(x_i|\mathbf{x}_{\pi_i}) = u(x_i) \exp\{\theta_i^T \boldsymbol{\phi}_i(x_i, \mathbf{x}_{\pi_i}) - A(\theta_i, \mathbf{x}_{\pi_i})\}, \tag{4.6}$$

where $\boldsymbol{\phi}_i(x_i, \mathbf{x}_{\pi_i})$ is a vector of potentials associated with the variable set $\{x_i, \mathbf{x}_{\pi_i}\}$.

The exponential representation applies to a wide range of models of practical interest, including discrete models, Gaussian, Poisson, exponential, and many others.

4.3.3 Lower Bounds of General Exponential Functions

Now we review some basic results from standard calculus that provide a principled way of constructing higher-order bounds for regular functions. Start from a simple bound for a function $f_0(x)$: $f_0(x) \geq b_0(x), \forall x \in \mathcal{X}$.

Lemma 1 *For anti-derivatives $f_1(x)$ of f_0 and $b_1(x)$ of b_0 such that $f_1(a) = b_1(a)$ for some $a \in \mathcal{X}$:*

$$\begin{array}{ll} f_1(x) \leq b_1(x) & \textit{for } x \leq a \\ f_1(x) \geq b_1(x) & \textit{for } x \geq a \end{array}$$

Proof. Due to the simple bound assumption, for $x \geq a$:

$$\begin{array}{rrcl} & \int_a^x dz f_0(z) & \geq & \int_a^x dz b_0(z) \\ \Rightarrow & f_1(x) - f_1(a) & \geq & b_1(x) - b_1(a) \\ \Rightarrow & f_1(x) - b_1(x) & \geq & f_1(a) - b_1(a) = 0. \end{array}$$

The other direction (i.e., when $x \leq a$) follows similarly. ■

Lemma 2 *For anti-derivatives $f_2(x)$ of f_1 and $b_2(x)$ of b_1 such that $f_2(a) = b_2(a)$, $f_1(a) = b_1(a)$:*

$$f_2(x) \geq b_2(x) \quad \textit{for } x \in \mathcal{X}$$

Proof. Due to Lemma 1, for $x \leq a$:

$$\begin{array}{rcl} \int_x^a dz f_1(z) & \leq & \int_x^a dz b_1(z) \\ \Rightarrow \quad f_2(x) - b_2(x) & \geq & f_2(a) - b_2(a) = 0 \end{array}$$

For $x \geq a$, the same inequality follows similarly. ■

Thus we have the following theorem:

Theorem 1 *Let $f_k(x)$ denote the kth-order anti-derivative of the function $f(x)$. Given a lower bound $b(x)$ of the function $f(x)$, the 2nd-order anti-derivative $b_2(x)$ of the original bound, parameterized by a variational parameter μ such that $b_1(\mu) = f_1(\mu)$ and $b_2(\mu) = f_2(\mu)$, is a lower bound of $f_2(x)$. Likewise (by induction), bounds for higher-order anti-derivatives of f can be successively constructed.*

Since the anti-derivative of the exponential function is just itself, we can easily use Theorem 1 to obtain linear and higher-degree polynomial bounds from bounds of lower order. For example, the well known linear bound of the exponential function, its tangent at $x = \mu$ (see Fig 4.7), can be readily derived from the trivial bound $\exp(x) > 0$ using Theorem 1:

$$f(x) = \exp(x) \geq \exp(\mu)(1 + x - \mu) = b_2(x), \forall x, \mu \tag{4.7}$$

Integrating over both sides twice, and denoting the variational parameters in the new bound as ν (which means that new bound "touches" the original function at ν), we have the following third-order bound:

$$\begin{array}{rcl} f(x) & = & \exp(x) \\ & \geq & \exp(\nu)\{1 + x - \nu + \exp(\xi)(\frac{1-\xi}{2}(x-\nu)^2 + \frac{1}{6}(x-\nu)^3)\}, \\ & = & b_4(x). \end{array} \tag{4.8}$$

where $\xi = \mu - \nu$. When $\xi = 0$, that is, restricting the higher order bound to "touch" the original function at the same point as the lower order bound, we have $b_4(x) = \frac{1}{6}\exp(\mu)((x-\mu)^3 + 3(x-\mu)^2 + 6(x-\mu+1))$. From Figure 4.7, we can see that this bound is much tighter than the linear bound.

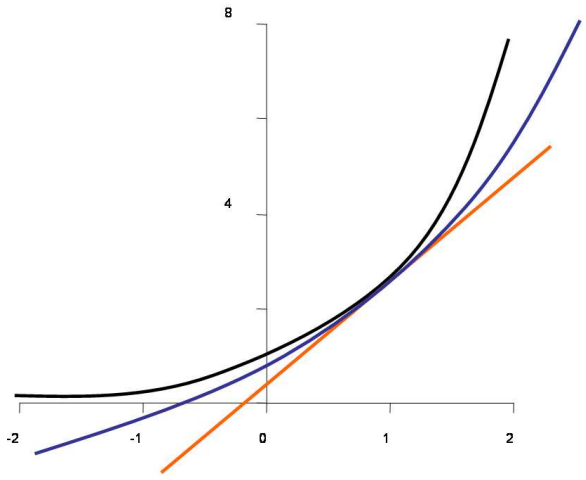


Figure 4.7: The tangent (blue curve) and polynomial (red curve) bounds for an exponential function (black curve).

4.3.3.1 Lower bounding probabilistic invariants

The tangent and polynomial bounds of exponential functions can be used to define objective functionals underlying the variational principle for probabilistic inference by introducing bounds for the probabilistic invariants associated with a distribution and/or data, such as the likelihood and the partition function. Let $q(\mathbf{x}_H) = \exp\{-E'(\mathbf{x}_H)\}$ represent an arbitrary probability distribution (written in an exponential representation) over the hidden variables of a model to be approximated. A bound for the likelihood can be characterized by the following lemma.

Lemma 3 *Every marginal distribution* $q(\mathbf{x}_H) = \exp\{-E'(\mathbf{x}_H)\}$ *defines a lower bound of likelihood:*

$$p(\mathbf{x}_E) \geq \int d\mathbf{x}_H \exp\Big\{-E'(\mathbf{x}_H)\Big\}\Big(1 - A(\mathbf{x}_E) - \big(E(\mathbf{x}_H, \mathbf{x}_E) - E'(\mathbf{x}_H)\big)\Big), \quad (4.9)$$

where $\mathbf{x}_E$ *denotes observed variables (evidence).*

Proof. Using the tangent bound of the exponential function (Eq. 4.7), for a joint distribution $p(\mathbf{x}_H, \mathbf{x}_E) = \exp\{-E(\mathbf{x}_H, \mathbf{x}_E) - A(\mathbf{x}_E)\}$ (where $A(\mathbf{x}_E)$ is the original log-partition function plus the constant evidence potentials), we replace x in Eq. (4.7) with $-(E(\mathbf{x}_H, \mathbf{x}_E) + A(\mathbf{x}_E))$ and lower bound the joint distribution $p(\mathbf{x}_H, \mathbf{x}_E)$ as follows:

$$p(\mathbf{x}_H, \mathbf{x}_E) \geq q(\mathbf{x})(1 - A(\mathbf{x}_E) - (E(\mathbf{x}_H, \mathbf{x}_E) - E'(\mathbf{x}_H))), \tag{4.10}$$

where $E'(\mathbf{x}_H)$ defines a *variational marginal distribution*. Integrating over $\mathbf{x}_H$ on both sides, we obtain the first-order lower bound in Eq. (4.9). ■

This bound is similar to the well-known Jensen bound on the *log-likelihood*: $\log p(\mathbf{x}_E) \geq \int d\mathbf{x}_H q(\mathbf{x}_H) \log \frac{q(\mathbf{x}_H)}{p(\mathbf{x}_H, \mathbf{x}_E)}$, and has the same maximizer, but it is more general in that it can be further upgraded to higher order bounds for tighter approximation using Eq. (4.8).

Rearranging terms on the right hand side of inequality (4.9), we have the following compact form of the lower bound on the likelihood:

$$\begin{aligned} p(\mathbf{x}_E) &\geq C - \left\langle E(\mathbf{x}_H, \mathbf{x}_E) \right\rangle_{q(\mathbf{x}_H)} + \left\langle \log q(\mathbf{x}_H) \right\rangle_{q(\mathbf{x}_H)} \\ &= C - \left\langle E \right\rangle_q - H_q, \end{aligned} \tag{4.11}$$

where the first term C is a constant related to the log-partition function of the original distribution, the second term $\left\langle E \right\rangle_q$ is the *expected energy* under distribution q, and the third term H_q is the *entropy* of distribution q. Note that when no variable in a model is observed, the foregoing exposition can lead to a lower bound on the log-partition function:

$$A \geq 1 - \left\langle E \right\rangle_q - H_q. \tag{4.12}$$

For simplicity, we focus on the likelihood in the sequel, but the exposition applies readily to the bound on the log-partition function.

4.3.4 A General Variational Principle for Probabilistic Inference

The likelihood bound derived in the previous section plays a pivotal role in formulating a probabilistic inference problem variationally, because it makes explicit an objective functional that can be

optimized over the space of all distributions, and leads to a variational representation of a probability distribution.

4.3.4.1 Variational representation

Let $\mathcal{Q}$ denote the set of all distributions on $\mathcal{X}^n$. Given any distribution p represented in the form (4.5), from Eq. (4.9), it is apparent from our discussion so far that the associated likelihood function $p(\mathbf{x}_E)$ can be recovered as a solution of the following optimization problem:

$$\begin{aligned} p(\mathbf{x}_E) &= \max_{q \in \mathcal{Q}} \Big\{ -\big\langle E \big\rangle_q - H_q \Big\} \\ &= \min_{q \in \mathcal{Q}} \Big\{ \big\langle E \big\rangle_q + H_q \Big\}. \end{aligned} \tag{4.13}$$

Moreover, the optimum is uniquely attained when $q = p$. Note that here the optimization problem is defined on a first-order lower bound of the likelihood, and an equivalent result can also be obtained from the well-known minimal KL problem: $\min_{q \in \mathcal{Q}} \mathrm{KL}(q\|p) = 0$, attained at $q = p$, where $\mathrm{KL}(q\|p) \equiv \int_{\mathbf{x}} \log q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})}$ is the Kullback-Leibler divergence from q to p. But for higher-order bounds of $p(\mathbf{x}_E)$, although the solution (i.e., the optimizer) remains the same, a different optimization problem needs to be solved, whose relaxation may lead to better approximation.

Consider exponential family graphical models. In this case, the optimization problem described above takes place over a space that includes all choices of potential functions $\boldsymbol{\phi}$ and all valid weight parameters $\boldsymbol{\theta}$ associated with these potential functions. It should be clear that depending on the choice of canonical parameterization for the density functions $q(\cdot)$, the formal definition of the optimization space varies significantly. For example, under the *exponential* parameterization as we used here for exponential families, $\boldsymbol{\theta}$ belongs to the set $\Theta \equiv \{\boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{D}|} \,|A(\boldsymbol{\theta}) < \inf\}$; under the *mean parameterization* for discrete distributions, one needs to optimize over a *marginal polytope* [Wainwright and Jordan, 2003], $\mathcal{M} \equiv \{\mu \in \mathbb{R}^{|\mathcal{D}|} \mid \exists p(\cdot) \text{ s.t. } \int \boldsymbol{\phi}(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \mu\}$, where $\boldsymbol{\phi}(\mathbf{x})$ denotes the vector of all potential functions associated with the graphical model. Wainwright *et al.* [2003] pointed out that if and only if the exponential representation is minimal (i.e., no affine combination of $\boldsymbol{\phi}(\mathbf{x})$ is equal to a constant), there is a one-to-one mapping from Θ to $\mathcal{M}$.

In general, computing the entropy for an arbitrary distribution q, and hence the objective function in Eq. (4.13) is intractable. Furthermore, in many cases of interest, characterizing the optimization space (e.g., the marginal polytope) is not possible. Thus usually one cannot solve the variational representation defined by Eq. (4.13) analytically. Variational inference amounts to seeking an optimal q^* under a *relaxed* variational representation, which is entailed by approximating the entropy H_q; or redefining (e.g., relaxing or tightening) the optimization space $\mathcal{Q}$, so that within the redefined space, referred to as a *feasible space*, the entropy of q is tractable; or doing both. We refer to the resulting $q*$ as a *variational approximation* to the true distribution p:

Definition 1 Variational approximation

$$(\mathbf{VP}) \qquad q \;=\; \arg\max_{q\in\mathcal{Q}_v} \Big\{ -\big\langle E \big\rangle_q - F_v(q) \Big\} \tag{4.14}$$

where $\mathcal{Q}_v$ is the feasible space of realizable distributions, and $F_v(q)$ is an approximate entropy term defined on q.

4.3.4.2 Mean field methods

One class of variational inference methods attempts to approximate a distribution p using a family of tractable distributions, $q(\mathbf{x}|\gamma)$, which are defined on subgraphs of the original graph $G(p)$, for which exact computation of the entropy H_q is feasible. The γ are a set of free "variational parameters." This class of methods is referred to as "mean field methods" [Jordan *et al.*, 1999], a terminology that reflects the classical setting in which $q(\mathbf{x}|\gamma)$ is taken to be a completely factorized distribution. From an optimization theoretic point of view, a mean field method solves a reduced version of problem (4.14), in which $\mathcal{Q}_v = \mathcal{T}$, where $\mathcal{T}$ denotes the space of all distributions that factor according to *tractable subgraphs* of $G(p)$. This is an *inner approximation* of the space of all possible distributions (i.e., $\mathcal{T} \subset \mathcal{Q}$) [Wainwright and Jordan, 2003]. In these methods, $F_v(q) = H_q$, is the exact entropy for q. It is easy to see that such a reduction defines a lower bound on the likelihood $p(\mathbf{x}_E)$ (because we are optimizing over a subspace of the original optimization space), and hence mean field methods are essentially maximizing a lower bound of the true likelihood, a

nice property useful in justifying their application, especially in likelihood-based model learning (i.e., parameter estimation), although in practice the tightness of the bound heavily depends on the choice of feasible space.

Recall that $\mathcal{Q}$ consists of two components: the space of potential functions $\boldsymbol{\phi}$ and the space of parameters $\boldsymbol{\theta}$. For a general multivariate probability distribution, the potential space spans the choices of both the *coupling topology* (i.e., which subsets of variables $\mathbf{x}_D$ come under a single potential) and the *coupling kernel* (i.e., the functional form of $\phi(\cdot)$). The coupling topology is encoded in the graphical representation of a multivariate distribution, and the coupling kernels reflect choices of mappings from the joint state configurations of variable subsets to values related to their joint probabilities. In principle, optimization could take place in the space of, 1) all tractable subgraphs, 2) all valid potential functions (kernels) on such subgraphs, and 3) all valid parameters associated with the given set of potentials. In practice, nearly all extant mean field algorithms focus on parameter optimization (i.e., the 3rd aspect) but rarely explore the other two aspects, or only do so in an *ad hoc* way. For example, the classical mean field method makes use of the simplest subgraph of $G(p)$—the fully disjoint graph (i.e., with all edges removed), and chooses the potential function of each singleton to be the variable itself (i.e., $\phi(x) = x$). More recent *structured variational inference* methods [Jordan *et al.*, 1999] use more complex subgraphs of $G(p)$, in particular, some specific disjoint partitions of $G(p)$ motivated by both domain knowledge and computational tractability, and a set of model-specific choices of potential functions associated with the subgraph. To explore the third aspect of the optimization space, these methods seek an optimal value of the variational parameters via an iterative procedure using fixed-point equations derived in a problem-specific manner (e.g., depending on the choice of the coupling topology and the potential functions for the approximate distribution). Since substantial mathematical skills are usually involved, sophisticated mean field methods have not gained much popularity among practitioners of approximate inference.

4.3.4.3 Belief propagation

Recently, Yedidia *et al.* [2001b] realized that Pearl's belief propagation (BP) algorithm—when applied to general loopy graphs—is also a variational algorithm. The inference problem is transformed to an optimization functional—the "Bethe free energy"—that imposes local consistency on the approximate marginals. Specifically, BP, and related algorithms (e.g., GBP, CVM, etc.), seek to directly estimate a set of marginals of interest associated with the study distribution p, for example, all marginals of variable pairs that are adjacent in the graph $G(\mathcal{V}, \mathcal{E})$, i.e., $\{\mu_{ij} \equiv \left\langle X_i X_j \right\rangle_p | \forall i, j, \text{ s.t., } (ij) \in \mathcal{E}\}$, and all the singleton marginals, i.e., $\{\mu_i \equiv \left\langle X_i \right\rangle_p | \forall i, \text{ s.t., } i \in \mathcal{V}\}$, by optimizing a so-called *Bethe free energy*. As pointed out by Wainwright and Jordan [2003], this problem can be understood as seeking a particular mean parameterization for an approximate distribution.

Under the general framework of variational approximation described by Eq. (4.14), the Bethe free energy is equal to the sum of the expected energy $\left\langle E \right\rangle_q$ as in Eq. (4.14), and another term called the *Bethe entropy*, H_{Bethe}, which is an approximation to the true entropy H_q. Recall that H_q is intractable for general distributions; H_{Bethe} makes use of all single node entropies $H_i(\mu_i)$ and edgewise mutual information terms $I_{ij}(\mu_{ij})$ to form an approximation to H_q:

$$H_{Bethe}(\mu) \triangleq -\sum_{i \in \mathcal{V}} H_i(\mu_i) + \sum_{(i,j) \in \mathcal{E}} I_{ij}(\mu_{ij}). \tag{4.15}$$

An exact characterization of the marginal polytope given all the potential functions of the distribution p is intractable. To overcome this, BP optimizes over the space of *locally* consistent pairwise marginals (i.e., tree-consistent marginals): $\mathcal{M}_B \equiv \{\tau \geq 0 | \sum_{x_i} \tau_i(x_i) = 1, \sum_{x_i} \tau_{ij}(x_i, x_j) = \tau_j(x_j)\}$, which is an *outer approximation* to the original marginal polytope. The recently developed GBP algorithm optimizes over marginals of larger clusters of nodes to capture more complex couplings (than the pairwise couplings in baseline loopy BP) in the distribution p, which leads to a more complex optimization problem over the space of locally-consistent cluster marginals (a tighter outer approximation of the marginal polytope of p than that from the pairwise marginals), and on an objective function known as the *Kikuchi free energy* [Kikuchi, 1951] (a better approximation to

the true free energy than the Bethe free energy). Similar to the mean field methods, essentially BP algorithms also begin with an *ad hoc* choice of coupling topology (that determines variables to be included in cluster marginals), followed by an iterative procedure to search for fixed-points in the relaxed feasible space of marginals associated with each cluster.

An advantage of the Bethe (or Kikuchi) variational approach is the simplicity of the BP algorithms. Generic fixed-point equations can be derived based on the variational principle [Yedidia *et al.*, 2001b], which alleviates the need for model specific derivations in applications to a variety of specific problems. The flexibility provided by the ability to choose clusters of varying sizes in the GBP and CVM algorithms is a significant important step forward. However, due to the *ad hoc* relaxation of the original optimization functional and the feasible space for tractability, the marginals resulting from GBP are not necessarily globally consistent (i.e., not necessarily in the marginal polytope), so the inequality in Eq. (4.11) may no longer apply. Thus, the GBP approximation does not necessarily yield a lower bound on the likelihood and a GBP algorithm may not converge. Also note that since, in general, finding the mapping function from mean parameterization to the usual exponential parameterization is as difficult as performing inference, obtaining an explicit form of the approximate distribution via BP is non-trivial, which makes certain probabilistic queries, e.g., arbitrary marginals of p, difficult to handle. By contrast, in the mean field method, the solution is an explicit approximate distribution in exponential parameterization, on which general inference is tractable.

4.4 Generalized Mean Field Inference

Mean field methods can provide flexibility similar to that by the GBP methods via the choice of approximating distribution $q(\mathbf{x}|\gamma)$, and so-called "structured mean field methods" have been based on choosing $q(\mathbf{x}|\gamma)$ to be a tree or some other sparse subgraph of the original graph to which an exact inference algorithm such as the junction tree algorithm can be feasibly applied [Saul and Jordan, 1996]. Recently, Wiegerinck presented a general framework for structured mean field methods involving arbitrary clusterings [Wiegerinck, 2000]. In particular, his approach allows the use of

overlapping clusters, which leads to a set of mean field equations reminiscent of a junction tree algorithm. Although there continue to be developments in this area (e.g., [El-Hay and Friedman, 2001; Bishop *et al.*, 2003; Bishop and Winn, 2003]), it is fair to say that in practice the use of mean-field-based variational methods requires substantial mathematical skill and that a systematic approach with the generality, flexibility and ease of implementation of GBP has yet to emerge. In this section we describe a generalized mean field method that aims to fill this gap. The approach yields a simple general methodology that applies to a wide range of models. To obtain the desired simplicity our approach makes use of *nonoverlapping* clusters, specializing Wiegerinck's general approach, and yielding a method that is somewhat reminiscent of block methods in MCMC such as Swendsen-Wang [Swendsen and Wang, 1987].

Note that the choice of clusters is generally done manually both within the GBP tradition and the mean field tradition. Another reason for our interest in nonoverlapping clusters is that it suggests algorithms for automatically choosing clusters based on graph partitioning ideas. We will discuss a preliminary exploration of these ideas in § 4.5.

4.4.1 GMF Theory and Algorithm

As stated, the mean field approximation refers to a class of variational approximation methods that approximate the true distribution $p(\mathbf{x}|\theta)$ on a graph G with a simpler distribution, $q(\mathbf{x}|\gamma)$, for which it is feasible to do exact inference. Such distributions are referred to as *tractable families*. A tractable family usually corresponds to a subgraph of a graphical model.

4.4.1.1 Naive mean field approximation

The naive mean field approximation makes use of a subgraph that is completely disconnected. Thus, the approximating distribution is fully factorized:

$$q(\mathbf{x}) = \prod_{i \in \mathcal{V}} q_i(x_i). \tag{4.16}$$

For example, to use this family of distributions to approximate the joint probability of the Boltzmann machine: $p(\mathbf{x}) = \frac{1}{Z} \exp\{\sum_{i<j} \theta_{ij} x_i x_j + \sum_i \theta_{i0} x_i\}$ where $x_i \in \{0, 1\}$, one defines $q_i(x_i) =$

$\mu_i^{x_i}(1-\mu_i)^{1-x_i}$, where the μ_i are the variational parameters. Minimizing the Kullback-Leibler (KL) divergence between q and p, which is equivalent to solving Eq. (4.14) over the space of μ_i, one obtains the classical "mean field equations":

$$\mu_i = \sigma\big(\sum_{j\in\mathcal{N}_i} \theta_{ij}\mu_j + \theta_{i0}\big), \tag{4.17}$$

where $\sigma(z) = 1/(1+e^{-z})$ is the logistic function, and $\mathcal{N}_i$ is the set of nodes neighboring i. A little algebra shows that indeed each singleton marginal can be expressed as a conditional distribution of the relevant node given the expectation of all its neighbors, and this distribution reuses the set of coupling weights of the original distribution p:

$$\begin{aligned} q_i(x_i) &= \exp\Big\{\theta_{i0}x_i + \sum_{j\in\mathcal{N}_i}\theta_{ij}x_i\big\langle X_j\big\rangle_{q_j} + A_i\Big\} \\ &= p(x_i|\{\big\langle X_j\big\rangle_{q_j} \mid j\in\mathcal{N}_i\}). \end{aligned} \tag{4.18}$$

As the second line of Eq. (4.18) suggests, the mean field approximation to the singleton marginal is isomorphic to the corresponding singleton conditional under the original distribution p, with all the neighboring nodes of the singleton being conditioned on replaced by their expectations under their own singleton marginals. Conceptually, $\big\langle X_j\big\rangle_{q_j}$ resembles a 'message" sent from node j to i, and $\{\big\langle X_j\big\rangle_{q_j} \mid j\in\mathcal{N}_i\}$ forms a "mean field" applied to X_i from its neighborhood (Fig. 4.8).

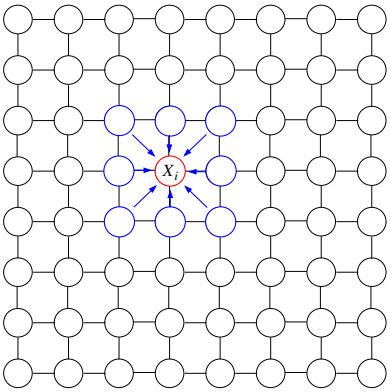


Figure 4.8: Mean field messages. The red node (X_i) denotes the variable whose marginal is being approximated; the blue nodes are neighbors that send the messages (assuming that these are the nodes whose couplings to node i, i.e., θ_{ij}, are non-zero).

Naive mean field approximation can be efficiently solved by fixed-point iteration. Procedurally, this is similar to a Gibbs sampling scheme (see Chapter 5) in which one iteratively samples each variable using a predictive distribution that conditions on the previously sampled values of the neighboring variables. However, due to the deterministic replacement of the true value with an expectation taken under an approximate marginal, the quality of the naive mean field approximation for arbitrary graphical models could break down in cases where the original graphs are sparse (so that the distribution of influences from the neighborhood may not be highly concentrated over an expectation) and the pairwise couplings are not uniform over all edges (i.e., the magnitudes of $\theta_{i,j}$ vary significantly over different node pairs, so that presence of strongly coupled pairs can bias the approximation).

4.4.1.2 Generalized mean field theory

The completely disconnected subgraph underlying the naive mean field approximation differs significantly from the original graph, implying that many of the dependencies present in the original model are left uncaptured. Intuitively, a subgraph with fewer edges removed would capture more such dependencies and would define a family of distributions better at approximating the original distribution. The basic idea of *generalized mean field* approximation is to employ a richer set of tractable approximate distributions which correspond to a subgraphs made up of tractable connected components of clusters of nodes.

Given a (disjoint) variable clustering $\mathcal{C}$, we define a *cluster-factorized distribution* as a distribution of the form $q(\mathbf{x}) = \prod_{C_i \in \mathcal{C}} q_i(\mathbf{x}_{C_i})$, where $q_i(\mathbf{x}_{C_i}) = \exp\{-E_i'(\mathbf{x}_{C_i})\}, \forall C_i \in \mathcal{C}$, are free distributions to be optimized. As discussed in §4.3.4, this optimization problem can be cast as that of maximizing a lower bound on the likelihood over the space of all valid cluster marginals respecting a given clustering $\mathcal{C}$. The solution to this problem leads a *generalized mean field approximation* to the original distribution $p(\mathbf{x})$. In the following, we present the generalized mean field theorem that states this result.

To make the exposition of the theorem and the resulting algorithm simple, we introduce some

definitions.

Definition 2 (Mean field factor): For a factorizable potential $\phi_\beta(\mathbf{x}_{D_\beta})$, let I_β denote the set of indices of those clusters that have nonempty intersection with D_β. Thus, $\phi_\beta(\mathbf{x}_{D_\beta})$ has as factors the potentials $\phi_{\beta_i}(\mathbf{x}_{C_i \cap D_\beta}), \forall i \in I_\beta$. Then, the *mean field factor* $f_{i\beta}$ is defined as:

$$f_{i\beta} \triangleq \langle \phi_{\beta_i}(\mathbf{X}_{C_i \cap D_\beta}) \rangle_{q_i}, \quad \text{for } i \in I_\beta \tag{4.19}$$

where $\langle \cdot \rangle_{q_i}$ denotes the expectation with respect to q_i.

Definition 3 (Generalized mean fields): For any cluster C_j in a given variable partition, the set of mean field factors associated with the nodes in its *Markov blanket* is referred as the set of *generalized mean fields* of cluster C_j:

$$\mathcal{F}_j \triangleq \{f_{i\beta} : D_\beta \in \mathcal{B}_j, i \in I_\beta, i \neq j\}. \tag{4.20}$$

From Eq. (4.9), replacing $E'(\mathbf{x}_H)$ with $\sum_{C_i \in \mathcal{C}} E'_i(\mathbf{x}_{C_i})$ and omitting $A(\mathbf{x}_E)$ (which is a constant determined by p, the distribution to be approximated) the optimal generalized mean field approximation to p is specified as the solution of the following constrained optimization problem:

$$\{E_i'^{GMF}(\mathbf{x}_{C_i})\}_{C_i \in \mathcal{C}} = \arg \max_{E'_i \in \mathbb{E}(\mathbf{x}_{C_i})} \int d\mathbf{x} \exp\{-\sum_{C_i \in \mathcal{C}} E'_i(\mathbf{x}_{C_i})\} \Big(1 - \big(E(\mathbf{x}) - \sum_{C_i \in \mathcal{C}} E'_i(\mathbf{x}_{C_i})\big)\Big), \tag{4.21}$$

where $\mathbb{E}(\mathbf{x}_{C_i})$ denotes the set of all valid energy functions of variable set $\mathbf{x}_{C_i}$. (Because evidence variables are fixed constants in inference, for simplicity, we omit explicit mention of the evidence $\mathbf{x}_E$, and the subscript $_H$ in the energy term $E(\cdot)$ above and in other relevant terms in the following derivation. In should be clear that, in situations where such subscripts are omitted, $\mathbf{x}$ and related symbols denote only the hidden variables.) The solution to this problem leads to the follow Generalized Mean Field Theorem (the proof is provided in Appendix B.1),

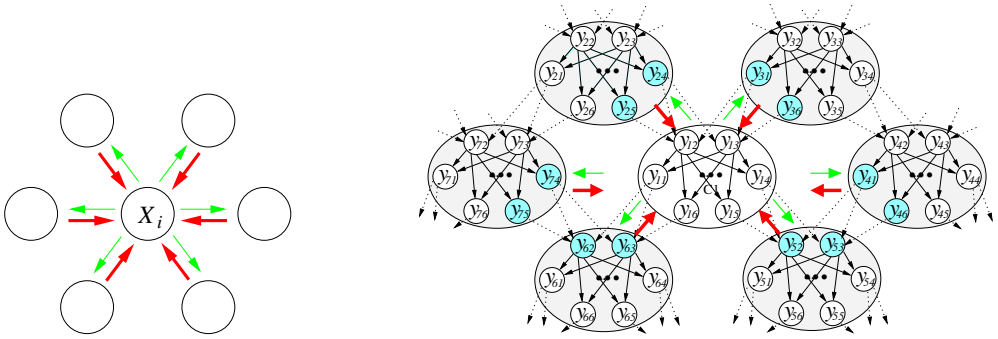
Theorem 2 (GMF approximation): *For a general undirected probability model* $p(\mathbf{x}_H, \mathbf{x}_E)$ *where* $\mathbf{x}_H$ *denotes hidden nodes and* $\mathbf{x}_E$ *denotes evidence nodes, and for a clustering* $\mathcal{C} : \{\mathbf{x}_{H,C_i}, \mathbf{x}_{E,C_i}\}_{i=1}^{I}$ *of both hidden and evidence nodes, if all the potential functions that cross cluster borders are cluster-factorizable, then the generalized mean field approximation to the joint posterior* $p(\mathbf{x}_H|\mathbf{x}_E)$ *with respect to clustering* $\mathcal{C}$ *is a product of cluster marginals* $q^{GMF}(\mathbf{x}_H) = \prod_{C_i \in \mathcal{C}} q_i^{GMF}(\mathbf{x}_{H,C_i})$ *satisfying the following* generalized mean field equations*:*

$$q_i^{GMF}(\mathbf{x}_{H,C_i}) \quad = \quad p(\mathbf{x}_{H,C_i}|\mathbf{x}_{E,C_i}, \mathcal{F}_i), \quad \forall i. \tag{4.22}$$

Remark 1 Note that each variational cluster marginal is isomorphic to the isolated model fragment corresponding to original cluster posterior given the intra-cluster evidence and the *generalized mean fields* from outside the cluster. Thus, each variational cluster marginal inherits all local dependency structures inside the cluster from the original model.

The mean field equations in Theorem 2 are analogous to naive mean field approximation by Eq. (4.18). The *generalized mean fields* appearing in Eq. (4.22) play a role that is similar to the conventional mean field, now applying to the entire cluster rather than a single node, and conducting probabilistic influence from the remaining part of the model to the cluster. It is easy to verify that when the clusters reduce to singletons, Eq. (4.22) is equivalent to the classical mean field equation Eq. (4.17) (Fig. 4.9). From a conditional independence point of view, the generalized mean fields can be also understood as an *expected Markov blanket* of the corresponding cluster, rendering its interior nodes conditionally independent of the remainder of the model and hence localizing the inference within each cluster given its generalized mean fields.

Mean field approximation for directed models is also covered by Theorem 2. This is true because any directed network can be converted into an undirected network via moralization, and designation of the potentials as local conditional probabilities. The following corollary makes this generalization explicit:



(a) (b)

Figure 4.9: The generalized mean fields in: (a) a naive mean field approximation and, (b) a GMF approximation. Red arrows denote GMFs received by the center cluster (or node) from its neighborhood, green arrows denote GMFs contributed by the center cluster (or node) to its neighborhood.

Corollary 3 *For a directed probability model* $p(\mathbf{x}_H, \mathbf{x}_E) = \prod_i p(x_i|\mathbf{x}_{\pi_i})$ *and a given disjoint variable partition, if all the local conditional models* $p(x_i|\mathbf{x}_{\pi_i})$ *across the cluster borders admit cluster-factorizable potentials, then the generalized mean field approximation to the original distribution has the following form:* $q^{GMF}(\mathbf{x}_H) = \prod_{C_i \in \mathcal{C}} q_i^{GMF}(\mathbf{x}_{H,C_i})$*, and*

$$q_i^{GMF}(\mathbf{x}_{H,C_i}) \quad = \quad p(\mathbf{x}_{H,C_i}|\mathbf{x}_{E,C_i}, \mathcal{F}_i), \quad \forall i, \tag{4.23}$$

where $\mathcal{F}_i$ *refers to the generalized mean fields of the exterior parents, children and co-parents of the variables in cluster* i*.*

These theorems make it straightforward to obtain generalized mean field equations. All that is needed is to decide on a subgraph and a variable clustering, to identify the Markov blanket of each cluster, and to plug in the mean fields of the Markov blanket variables according to Eqs. (4.22) or (4.23). We illustrate the application of the generalized mean field theorem to several typical cases—undirected models, directed models, and models that combine continuous and discrete random variables.

Example 1 *(2-d nearest-neighbor Ising model):* For a 2-d nearest neighbor Ising model, we can pick a subgraph whose connected components are square blocks of nodes in the original graph

(Fig. 4.10). The cluster marginal of a square block G_k is simply $q(\mathbf{x}_{G_k}) = \exp\{\sum_{(ij)\in\mathcal{E}(G_k)} \theta_{ij}x_i x_j + \sum_{i\in\mathcal{V}(G_k)} \theta_{i0}x_i + \sum_{(ij)\in\mathcal{E}(G),j\in\mathcal{MB}(G_k)} \theta_{ij}\langle x_j\rangle x_i\}$, an Ising model of smaller size, with singleton potentials for the peripheral nodes adjusted by the mean fields of the adjacent nodes outside the block (which are the $\mathcal{MB}$ of $\mathbf{x}_{G_k}$). ◇

Example 2 *(factorial hidden Markov models):* For the fHMM, whose underlying graph consists of multiple chains of discrete hidden Markov variables coupled by a sequence of output nodes, taken to be linear-Gaussian for concreteness, a possible subgraph that defines a tractable family is shown in Figure 4.12, in which we retain only the edges within each chain of the original graph. Given a clustering $\mathcal{C}$, in which each cluster k contains a subset of HMM chains c_k (the dashed boxes in Fig. 4.12), the MB of each cluster consists of all nodes outside the cluster. Hence the cluster marginal of c_k is: $q(\{\mathbf{x}^{(m_i)}\}_{i\in c_k}) \propto \prod_{i\in c_k} p(\mathbf{x}^{(m_i)})p(\mathbf{y}|\{\mathbf{x}^{(m_i)}\}_{i\in c_k}, \{f(\mathbf{x}^{(m_j)})\}_{j\in c_l, l\neq k})$, where $\mathbf{x}^{(m_i)}$ denotes variables of chain m_i, $p(\mathbf{x}^{(m_i)})$ is the usual HMM of a single chain, and $p(\mathbf{y}|\cdot)$ is linear-Gaussian. When each c_k contains only a single chain, we recover the structured variational inference equations in Ghahramani and Jordan [1997]. ◇

Example 3 *(Variational Bayesian learning):* Following the standard setup in Ghahramani and Beal [2001], we have a *complete data likelihood* $P(\mathbf{x}, \mathbf{y}|\theta)$, where $\mathbf{x}$ is hidden, and a *prior* $p(\theta|\eta, \nu)$, where η, ν are *hyperparameters*. Partitioning all domain variables into two clusters, $\{\mathbf{x}, \mathbf{y}\}$ and $\{\theta\}$, if the potential function at the cluster border, $\phi(\mathbf{x}, \theta)$, is factorizable (which is equivalent to the condition of *conjugate exponentiality* in Ghahramani and Beal [2001]), we obtain the following cluster marginals using Corollary 3:

$$\begin{aligned} q(\theta) &= p(\theta|\eta, \nu, f(\mathbf{x}), \mathbf{y}) \propto p(f(\mathbf{x}), \mathbf{y}|\theta)p(\theta|\eta, \nu) \\ q(\mathbf{x}) &= p(\mathbf{x}|\mathbf{y}, f(\theta)). \end{aligned}$$

These coupled updates are identical to the variational Bayesian learning updates of Ghahramani and Beal [2001] and Attias [2000]. ◇

4.4.2 A more general version of GMF theory

Recall that the GMF theory developed in the last section assumes the potential functions of the cliques in the graphical models are *cluster-factorizable*, which is not always true for general distributions, for example, in case of a distribution defined by tabular potential functions. Now we briefly sketch a more *general* version the GMF theory, which subsumes the previous version.

Given a disjoint variable partitioning, $\mathcal{C}$, the true *cluster conditional* of each variable cluster C_i given its Markov blanket $\mathcal{MB}_i$ is:

$$p(\mathbf{x}_{C_i}|\mathbf{X}_{\mathcal{MB}_i} = \mathbf{x}_{\mathcal{MB}_i}) \quad \propto \quad \exp\Big\{ \sum_{D_\alpha \subseteq C_i} \theta_\alpha \phi_\alpha(\mathbf{x}_{D_\alpha}) + \sum_{D_\beta \subseteq \mathcal{B}_i} \theta_\beta \phi_\beta(\mathbf{x}_{D_\beta \cap C_i}, \mathbf{x}_{D_\beta \cap \mathcal{MB}_i}) \Big\}, \tag{4.24}$$

where $\mathcal{B}_i$ denotes the set of cliques that intersect with but are not contained in cluster C_i. Note that in Eq. (4.24), we distinguish two types of variables in each clique: $\mathbf{x}_{D_\beta \cap C_i}$ represents the variables in the intersection of clique D_β and cluster C_i, and $\mathbf{x}_{D_\beta \cap \mathcal{MB}_i}$ represents the variables in clique D_β but outside cluster C_i. Without loss of generality, we assume that all the potentials are positively weighted (i.e., $\theta > 0$) and the signs are subsumed in the potential functions.

Given a clique D_β, recall that we use I_β to denote the set of indices of clusters that have non-empty intersection with D_β. Let $I_{\beta i}$ denotes $I_\beta \setminus i$, which indexes the set of clusters other than C_i that intersect with clique D_β; let $q_{I_{\beta i}}(\cdot) = \prod_{j \in I_{\beta i}} q_j(\mathbf{x}_{C_j})$ denote the marginal distribution (defined by a product of mean field cluster marginals) over the variables in these clusters (note that $\mathbf{x}_{D_\beta \cap \mathcal{MB}_i}$ is a subset of the set of all variables in these clusters: $\{\mathbf{x}_{C_j} \,|j \in I_{\beta i}\}$). Finally, let us refer to the (marginal) expectation of the potential $\phi_\beta(\mathbf{X}_{D_\beta})$ under the mean field cluster marginals indexed by $I_{\beta i}$ as a *peripheral marginal potential* of cluster C_i:

$$\begin{aligned} \phi'_\beta(\mathbf{x}_{D_\beta \cap C_i}, q_{I_{\beta i}}) \quad &\triangleq \quad \big\langle \phi_\beta(\mathbf{x}_{D_\beta}) \big\rangle_{q_{I_{\beta i}}} \\ &= \quad \int \phi_\beta(\mathbf{x}_{D_\beta \cap C_i}, \mathbf{x}_{D_\beta \cap \mathcal{MB}_i}) q_{I_{\beta i}}(\mathbf{x}_{D_\beta \cap MB_i}) d\mathbf{x}_{D_\beta \cap \mathcal{MB}_i}, \end{aligned} \tag{4.25}$$

which is only a functions of the variables in the intersection of clique D_β cluster C_i, and $q_{I_{\beta i}}(\cdot)$.

Given the peripheral marginal potentials of all the cliques intersecting with cluster C_i, we can easily show (similar to the proof of Theorem 2 in Appendix B.1, and hence omitted) that the GMF approximation to the cluster marginal of this cluster is:

$$q_i(\mathbf{x}_{C_i}) \quad \propto \quad \exp\Big\{ \sum_{D_\alpha \subseteq C_i} \theta_\alpha \phi_\alpha(\mathbf{x}_{D_\alpha}) + \sum_{D_\beta \subseteq \mathcal{B}_i} \theta_\beta \phi'_\beta(\mathbf{x}_{D_\beta \cap C_i}, q_{I_{\beta i}}) \Big\}, \tag{4.26}$$

from which the isomorphism of the GMF approximation of the cluster marginal to the true cluster conditional (i.e., Eq. (4.24)) is apparent.

The definition of peripheral marginal potentials is more general than the *mean field messages* defined in the last section, which can be viewed as a special case that applies to *cluster-factorizable potentials*. For other non-factorizable potentials, such as tabular potentials, peripheral marginal potentials are still well defined.

4.4.3 A Generalized Mean Field Algorithm

Eqs. (4.22) and (4.23) are a coupled set of nonlinear equations, which are solved numerically via asynchronous iteration until a fixed point is reached. This iteration constitutes a simple, message-passing style, generalized mean field algorithm.

GMF (model: $p(\mathbf{x}_H, \mathbf{x}_E)$, partition: $\{\mathbf{x}_{H,C_i}, \mathbf{x}_{E,C_i}\}_{i=1}^I$)
Initialization
– Randomly initialize the hidden nodes at the border of cluster i, $\forall i$.
– Initialize $f_{i\beta}^{\ 0}$ by evaluating the potentials using the current values of the associated nodes.
– Initialize $\mathcal{F}_i^{\ 0}$ with the current $f_{i\beta}^{\ 0}$.
While not converged
For $i = 1 : I$
– Update $q_i^{t+1}(\mathbf{x}_{H,C_i}) = p(\mathbf{x}_{H,C_i} | \mathbf{x}_{E,C_i}, \mathcal{F}_i^{\ t})$.
– Compute the mean field factors $f_{i\beta}^{\ t+1}$ of all potential factors at the border of C_i via local inference using q_i^{t+1} as in Eq. (4.19).
– Send the $f_{i\beta}^{\ t+1}$ messages to all Markov blanket clusters of i by updating the appropriate elements in their GMFs: $\mathcal{F}_j^t \rightarrow \mathcal{F}_j^{t+1}, \forall j \in \mathcal{MBC}_i$.
End
Return $q(\mathbf{x}_H) = \prod_i q_i(\mathbf{x}_{H,C_i})$, the GMF approximation

Remark 2 Note that the right-hand side of the mean field equation of cluster marginal q_i (Eqs. (4.22) and (4.23)) depends only on a set of cluster marginals that are functions on the Markov blanket variables of cluster C_i; this set of marginals does not include q_i. Thus, the iterative update is a form of coordinate ascent in the factored model space (i.e., we fix all $q_j(\mathbf{x}_{H,C_j}), j \neq i$ and maximize with respect to $q_i(\mathbf{x}_{H,C_i})$ at each step), which will lead to a fixed point. Therefore we have the following convergence theorem.

Theorem 4 *The GMF algorithm is guaranteed to converge to a local optimum, which is a lower bound for the likelihood of the model (see Remark 2 for a proof sketch).*

Theorem 4 is an important consequence of the use of a *disjoint* variable partition underlying the variational approximate distribution. It distinguishes GMF from other variational methods such as GBP [Yedidia *et al.*, 2001b], or the general case in Wiegerinck's framework [Wiegerinck, 2000], in which overlapping variable partitions are used, and which optimize an approximate free energy function with respect to marginals which must satisfy local constraints.

The complexity of each iteration of GMF is exponential in the tree width of the *local* networks of each cluster of variables, since inference is reduced to local operations within each cluster. However, this also means that a computational advantage can only be obtained if the maximum clique size of q_i is much smaller than that of p, suggesting that an appropriate variable partition which breaks large cliques is important for the success of GMF, an issue we explore in the next section.

Since GMF is guaranteed to converge to a local optimum, in practice it can be performed in a stochastic multiple-initialization setting similar to the usual practice in EM, to increase the chance of finding a better local optimum.

4.4.4 Experimental Results

Although GMF supports several types of applications, such as finding bounds on the likelihood or log-partition function, computation of approximate marginal probabilities, and parameter estimation, in this section we focus solely on the quality of approximate marginals. We have performed experiments on three canonical models: a nearest neighbor Ising model (IM), a sigmoid network

(SN), and a factorial HMM (fHMM); and we have compared the performance of GMF using different tractable families (specifically, using variable clusterings of different granularity) with regard to the accuracy on single-node marginals. To assess the error, we use an L_1-based measure

$$\frac{1}{\sum_{i=1}^{N} M_i} \sum_{i=1}^{N} \sum_{k=1}^{M_i} |p(X_i = k) - q(X_i = k)|,$$

where N is the total number of variables, and M_i is the number of (discrete) states of the variable x_i. The exact marginals were obtained via the junction tree algorithm. We also compared the performance with that of the belief propagation (BP) algorithm, especially in cases where BP is expensive, and examined whether GMF provides a reasonably efficient alternative.

We used randomly generated problems for the IM and SN and real data for the fHMM. For the first two cases, in any given trial we specified the distribution $p(\mathbf{x}|\theta)$ by a random choice of the model parameter θ from a uniform distribution. For models with observable output (i.e., evidence), observations were sampled from the random model. Details of the sampling are specified in the tables presenting the results. For each problem, 50 trials were performed. The fHMM experiment was performed on models learned from a training data set.

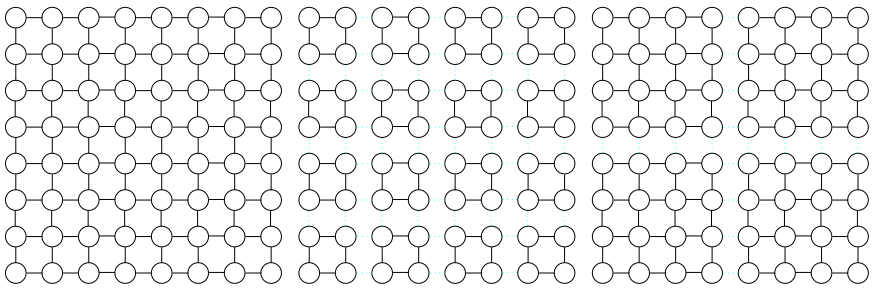


Figure 4.10: Ising model and GMF approximations.

Ising models: We used an 8×8 grid with binary nodes. Two different tractable models were used for the GMF approximation, one based on a clustering of 2×2 blocks, the other on 4×4 blocks (Fig. 4.10). Results on strongly attractive and repulsive Ising models (which are known to be difficult for naive MF) are reported in Table 4.1. The rightmost column also shows the mean CPU time (in seconds).

Table 4.1: L_1 errors on nearest neighbor Ising models. Upper panel: attractive IM ($\theta_{i0} \in (-0.25, 0.25), \theta_{ij} \in (0, 2)$); Lower panel: repulsive IM ($\theta_{i0} \in (-0.25, 0.25), \theta_{ij} \in (-2, 0)$).

Algorithm	Mean $\pm$ std	Median	Range	time
2×2 GMF	0.366±0.054	0.382	[0.276,0.463]	2.0
4×4 GMF	0.193±0.103	0.226	[0.004,0.400]	29.4
BP	0.618±0.304	0.663	[0.054,0.995]	17.9
GBP	0.003±0.002	0.002	[0.000,0.005]	166.3
2×2 GMF	0.367±0.052	0.383	[0.279,0.449]	1.2
4×4 GMF	0.185±0.102	0.161	[0.009,0.418]	22.1
BP	0.351±0.286	0.258	[0.009,0.954]	14.3
GBP	0.003±0.003	0.003	[0.000,0.014]	117.5

As expected, GMF using a clustering with fewer nodes decoupled yields more accurate estimates than a clustering in which more nodes are decoupled, albeit with increased computational complexity. Overall, the performance of GMF is better than that of BP, especially for the attractive Ising model. For this particular problem, we also compared to the GBP algorithm, which also defines beliefs on larger subsets of nodes, with a more elaborate message-passing scheme. We found that for Ising models, GBP performs significantly better than the other methods, but at a cost of significantly longer time to convergence.

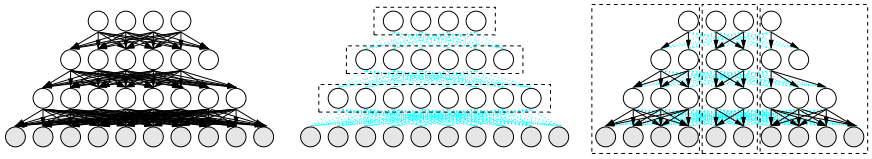


Figure 4.11: Sigmoid network and GMF approximations.

Sigmoid belief networks: The two sigmoid networks we studied are composed of three hidden layers (18 nodes), with and without a fourth observed layer (10 nodes), respectively. We used a row clustering and a block clustering of nodes as depicted in Figure 4.11 for GMF. Table 4.2 summarizes the results.

Table 4.2: L_1 errors on sigmoid networks ($\theta_{ij} \in (0, 1)$). Upper: hidden layers only; Lower: with observation layer.

Algorithm	Mean $\pm$ std	Median	Range	time
block GMF	0.013±0.004	0.013	[0.006,0.032]	6.8
row GMF	0.172±0.036	0.175	[0.100,0.244]	0.5
BP	0.273±0.025	0.271	[0.227,0.346]	9.2
block GMF	0.018±0.009	0.014	[0.009,0.038]	8.4
row GMF	0.061±0.021	0.059	[0.023,0.145]	0.7
BP	0.187±0.044	0.189	[0.096,0.312]	139.2

For the network without observations, the block GMF, which retains a significant number of

edges from the original graph, is more accurate by an order of magnitude than the row GMF, which decouples the original network completely. Interestingly, when a bottom layer of observed nodes is included in the network, a significant improvement in approximation accuracy is seen for the row GMF, but it still does not surpass the block GMF. The performance of BP is poor on both problems, and the time complexity scales up significantly for the network with the observation layer, because of the large fan-in associated with the nodes in the bottom layer.

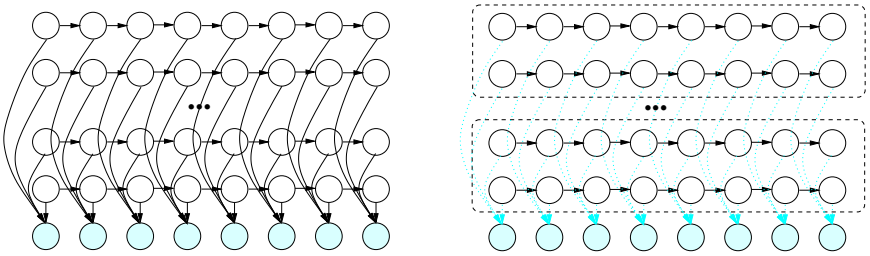


Figure 4.12: An fHMM and a GMF approximation (illustrative graph; the actual model contains 6 chains and 40 steps).

Factorial HMM: We studied a 6-chain fHMM, with (6-dimensional) linear-Gaussian emissions, ternary hidden state and 40 time steps. The model was trained using the EM algorithm (with exact inference) on 40 Bach chorales from the UCI Repository [Blake and Merz, 1998]. Inference was performed with the trained model on another 18 test chorales. GMF approximations were based on clusterings in which each cluster contains either singletons (i.e., naive mean field), one hidden Markov chain, two chains, or three chains, respectively. The statistics of the L_1 errors are presented in Table 4.3.

Table 4.3: L_1 errors on factorial HMM

Algorithm	Mean $\pm$ std	Median	Range	time
naive MF	0.254$\pm$0.095	0.269	[0.083,0.397]	9.8
1-chain GMF	0.237$\pm$0.107	0.233	[0.029,0.392]	14.3
2-chain GMF	0.092$\pm$0.081	0.064	[0.019,0.314]	5.6
3-chain GMF	0.118$\pm$0.092	0.089	[0.035,0.357]	15.6
BP	0	0	-	106.2

Since the moral graph of an fHMM is a clique tree, BP is exact in this case, but the computational complexity grows exponentially with the number of chains and the cardinality of the variables; hence BP cannot scale to large models. Using GMF, we obtain reasonable accuracy, which in general increases with the granularity of the variable clustering. The 2-chain GMF appears to be

a particularly good granularity of clustering in this case, leading to both better estimation and faster convergence.

In summary, GMF shows reasonable performance in all three of the canonical models we tested, and provides a flexible way to trade off accuracy for computation time. It is guaranteed to converge, and the computational complexity is determined by the treewidth of the subgraph. BP, on the other hand, may fail to converge. Furthermore, the complexity of computing BP messages is exponential in the size of the maximal clique in the moralized graph, which makes it very expensive in directed models with dense local dependencies. However, note that there are multiple ways of decomposing a graphical model (Fig. 4.13); in all three examples just studied, the clusterings of variables are chosen manually by examining the graph topology and studying the model semantics, and the choice affects the approximation quality significantly. Can we do this in a more principled way, especially for less structured graphs? In the following section, we address this problem.

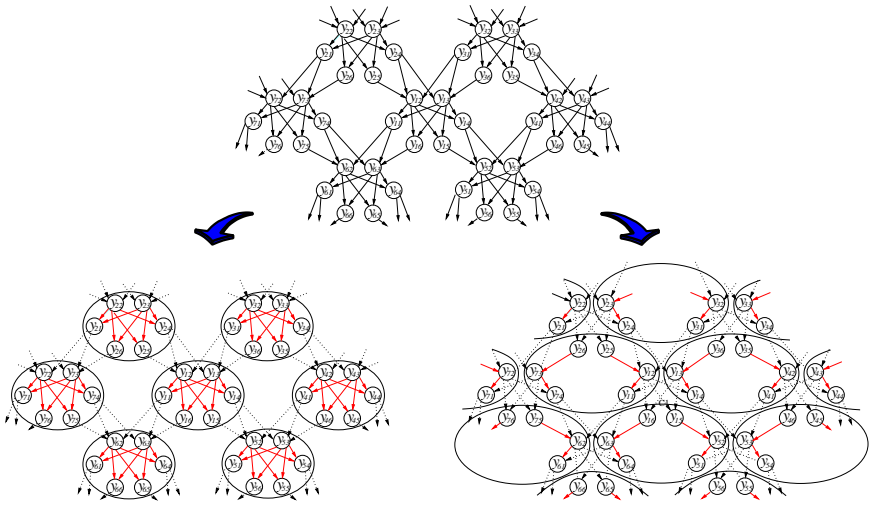


Figure 4.13: Two possible schemes for partitioning a graph to construct the GMF approximation. Which one is better?

4.5 Graph Partition Strategies for GMF Inference

What are the prospects for fully autonomous algorithms for variational inference in graphical models? Recent years have seen an increasingly systematic treatment of an increasingly flexible range

of algorithms for variational inference. In particular, the cluster variational framework has provided a range of algorithms that extend the basic "belief propagation" framework [Yedidia *et al.*, 2001a]. Similarly, general clusters of variables are also allowed in recent treatments of structured mean field algorithms [Wiegerinck, 2000]. Empirical results have shown that both kinds of generalization can yield more effective algorithms.

While these developments provide much-needed flexibility for the design of effective algorithms, they also raise a new question—how are the clusters to be chosen? Until now, this issue has generally been left in the hands of the algorithm designer; moreover, the designer has been provided with little beyond intuition for making these choices. For some graphical model architectures, there are only a few natural choices, and these can be explored manually. In general, however, we wish to envisage a general piece of software for variational inference which can be asked to perform inference for an arbitrary graph. In this setting, it is essential to begin to explore automatic methods for choosing clusters.

In the previous section, we presented a generalized mean field algorithm for inference based on a *disjoint* clustering of random variables in a graphical model, noting that the assumption of disjoint clusters leads to a simple and generic set of inference equations that can be easily implemented. Disjoint clusters have another virtue as well, which is the subject of this section—they open the door to a role for graph partitioning algorithms in choosing clusters for inference.

There are several intuitions that support a possible role for graph partitioning algorithms in the autonomous choice of clusters for graphical model inference. The first is that minimum cuts are to be preferred, so that as much as possible of the probabilistic dependence is captured within clusters. It also seems likely that the values of parameters should matter because they often reflect the "coupling strength" of the probabilistic dependences among random variables. Another intuition is that maximum cuts should be preferred, because in this case the mean field acting across a large cut may have an expectation that is highly concentrated, a situation which corresponds to the basic assumption underlying mean field methods [Jordan *et al.*, 1999]. Again, specific values of parameters should matter.

In this section we provide a preliminary formal analysis and a thoroughgoing empirical exploration of these issues. We present a theorem that relates the weight of the graph cut to the quality of the bound of GMF approximation, and study random graphs and a variety of settings of parameter values. We compare several different kinds of partitioning algorithms empirically. As we will show, our results turn out to provide rather clear support for a clustering algorithm based on minimal cut, which is consistent with implications drawn from the formal analysis. These promising results open up the possibility for a fully autonomous variational inference algorithm for complex models based on automatic node partitioning of a graphical model and GMF fixed point iterations as illustrated in the following flowchart in Figure 4.14.

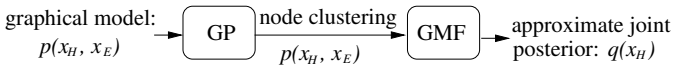


Figure 4.14: Flowchart of a autonomous variational inference algorithm.

4.5.1 Bounds on GMF Approximation

The quality of the GMF approximation depends critically on the choice of variable clustering of the graphical model. The following is a theorem that formally characterizes this relationship.

Theorem 5 (GMF bound on KL divergence): *The Kullback-Leibler divergence from the GMF approximate joint posterior to the true joint posterior is bounded by the sum of the weights of potential functions associated with the cross-border cliques, up to some constants intrinsic to the graphical model:*

$$aW \leq KL(q\|p) \leq bW, \tag{4.27}$$

where $W = \sum_{D_\beta \subseteq \cup \mathcal{B}_i} \theta_\beta$ *and,* a *and* b *are constants determined by the potential functions of the cross-border cliques (but independent of the potentials internal to the clusters.)*

A proof of this theorem is provided in Appendix B.2. Theorem 5 provides a clear guideline for choosing a desirable partitioning of a general graphical model: heuristically, it is desirable

to break cliques associated with small weights while clustering the variables in the graph; more systematically, we can use a graph partitioning algorithm to seek an optimal decomposition of the graph underlying the model. In the following, we explore several graph partitioning strategies on random graphs with pairwise potentials (each clique contains only two variables) to confirm and exploit Theorem 5 experimentally.

4.5.2 Variable Clustering via Graph Partitioning

A wide variety of graph partitioning algorithms have been explored in recent years in a number of fields (e.g., [Goemans and Williamson, 1995; Rendl and Wolkowicz, 1995]). Given our focus on disjoint clusters in the GMF approach, these algorithms have immediate relevance to the problem of choosing clusters for inference. In this section, we describe the methods that we have explored.

4.5.2.1 Graph partitioning

Let $G(\mathcal{V}, \mathcal{E}, A)$ be a weighted undirected graph with node set $\mathcal{V} = \{1, \ldots, n\}$, edge set $\mathcal{E}$ and nonnegative weights a_{ij}, for $(i, j) \in \mathcal{E}$ ($a_{ij} = 0$ if there is no edge between node i and j; also $a_{ii} = 0, \forall i$). We refer to the symmetric matrix $A = \{a_{ij}\}$ as the *affinity matrix*. We equip the space of $n \times n$ matrices with the trace inner product $A \bullet B = \text{tr}\, AB$; let $A \succeq 0$ denote positive semidefiniteness ($A \succeq B$ denotes $A - B \succeq 0$); and let $A \geq 0$ denote elementwise non-negativity of A. The linear operator $\text{Diag}(a)$ forms a diagonal matrix from the vector a, and its adjoint operator $\text{diag}(A)$ yields a vector containing the diagonal elements of A. We denote by e_k the vector containing k ones.

Equi-MinCut. We first consider graph partitioning (GP) problems based on minimum cuts. Given a graph $G(\mathcal{V}, \mathcal{E}, A)$ as described above, a classical formulation [Rendl and Wolkowicz, 1995] asks to partition the node set into k disjoint subsets, $(C_1, \ldots, C_k)$, of specified sizes $m_1 \geq m_2 \geq \ldots \geq m_k$, $\sum_{j=1}^{k} m_j = n$, so as to minimize the total weight of the edges connecting nodes in distinct subsets of the partition. This is known as the minimum k-*cut* of G. In this section, we concern ourselves with the special case of this problem in which all subsets have equal cardinality m, a

problem that we refer to as *k equi-MinCut* (k-$MinC$). [4] Equi-MinCut avoids potentially skewed cuts on highly imbalanced graphs, and leads to a balanced distribution of computational complexity among clusters.

A k-way node partition can be represented by an *indicator matrix* $X \in \mathbb{R}^{n\times k}$ with the j-th column, $x_j = (x_{1j}\ \ x_{2j} \ldots x_{nj})^t$, being the *indicator vector* for the set $C_j,\ \forall j$:

$$x_{ij} = \begin{cases} 1 & : \quad \text{if } i \in C_j \\ 0 & : \quad \text{if } i \notin C_j \end{cases}.$$

Thus, *k-equipartitions* of a graph are in one-to-one correspondence with the set

$$\mathcal{F}_k = \{X : Xe_k = e_n,\ X^t e_n = me_k,\ x_{ij} \in \{0,1\}\}.$$

For each partition X, the total weight of the edges connecting nodes within cluster C_i to nodes in its complement $\bar{C}_i$ is equal to $\frac{1}{2}x_i^t(D-A)x_i$, where $D = \text{Diag}(Ae_n)$. As a result, the total weight of the k-cut is

$$C_k = \sum_i \frac{1}{2}x_i^t(D-A)x_i = \frac{1}{2}\text{tr}X^tLX, \tag{4.28}$$

where $L \triangleq D - A$ is the *Laplacian matrix* associated with G.

Thus, k equi-MinCut can be modeled as the following integer programming problem

$$(k\text{-}MinC) \quad MinC_k^* := \min\{\text{tr } X^tLX \ :\ X \in \mathcal{F}_k\}.$$

Equi-MaxCut. We may also wish to find a k-partition that *maximizes* the total weight of the cut. This problem is known as the Max k-Cut in combinatorial optimization. Even without any size constraint this problem is NP-hard. In this paper, we again concern ourselves with a constrained version of the problem, in which all subsets have equal cardinality m. Thus we have the following *k equi-MaxCut* (k-$MaxC$) problem

$$\begin{aligned}(k\text{-}MaxC) \quad MaxC_k^* &:= \max\{\text{tr } X^tLX \ :\ X \in \mathcal{F}_k\} \\ &= \sum_i d_{ii} - \min\{\text{tr } X^tAX \ :\ X \in \mathcal{F}_k\}.\end{aligned}$$

[4] In combinatorial optimization, this problem is traditionally referred to as the *k-partition problem*. It is NP-hard, and to be distinguished from *unconstrained* minimum cut, which is *not* NP-hard.

We see that both k-$MaxC$ and k-$MinC$ are quadratic programs, and the relaxations that we consider will treat them identically. Note that due to the equality in the second line of the above equation, k-$MaxC$ can be solved in a similar manner to k-$MinC$, which amounts to using a different "cost" matrix in the objection function.

Weight matrices. The design of the affinity matrix has a fundamental impact on the results that are produced by graph partition algorithms. The naive choice in our case is to simply let $a_{ij} = 1$ when node i and j are connected in a graphical model, and let $a_{ij} = 0$ otherwise. Intuitively, an equi-MinCut using such an affinity matrix will capture more of the local dependency structure in the model, while an equi-MaxCut will lead to lower computational cost for inference in each cluster.

One can also partition the graphical model based on *coupling strength*, i.e., letting $a_{ij} = \theta_{ij}$, the weight of the pairwise potential, so that an equi-MinCut results in clusters with strong intra-cluster coupling, whereas an equi-MaxCut produces a clustering with only weak couplings left in each cluster.

It also seems sensible to consider weighting schemes that favor large cuts with small coupling strength, or small cuts and large coupling strengths. We explore such a scheme by choosing weights that are inversely related to coupling strength.

The following table summarizes the various partition strategies explored in this paper, and the corresponding design of the affinity matrix.

Table 4.4: Graph partition schemes

GP scheme	k-$MinC_a$	k-$MinC_b$	k-$MinC_c$	k-$MaxC_a$	k-$MaxC_b$	k-$MaxC_c$
a_{ij} value	$\{1, 0\}$	$\{\theta_{ij}, 0\}$	$\{\frac{1}{\theta_{ij}}, 0\}$	$\{1, 0\}$	$\{\theta_{ij}, 0\}$	$\{\frac{1}{\theta_{ij}}, 0\}$

4.5.2.2 Semi-definite relaxation of GP

Both *k equi-MinCut* and *k equi-MaxCut* are NP-hard. But there exist a variety of heuristics for finding approximate solutions to these problems [Frieze and Jerrum, 1995; Karisch and Rendl, 1998]. some applicable to quite large graphs [Falkner *et al.*, 1994]. In the sequel, we describe

an algorithm that finds an approximate solution to k-$MinC$ and k-$MaxC$ using a semidefinite programming (SDP) relaxation [Karisch and Rendl, 1998].

Semidefinite programming. Semidefinite programming (SDP) refers to the problem of optimizing a convex function over the convex cone of symmetric and positive semidefinite matrices, subject to linear equality constraints [Vandenberghe and Boyd, 1996]. A canonical (primal) SDP takes the form:

$$(\textbf{SDP}) \quad \begin{cases} \underline{\min} & C \bullet X \\ \underline{\text{s.t.}} & A_i \bullet X = b_i \quad \text{for } i = 1, \ldots, m \\ & X \succeq 0 \end{cases}$$

Because of the convexity of the objective function and the feasible space, every SDP problem has a single global optimum. With the development of efficient, general-purpose SDP solvers based on interior-point methods (e.g., SeDuMi [Sturm, 1999]), SDP has become a powerful tool in solving difficult combinatorial optimization problems. Here, we describe a simple SDP relaxation for solving graph partitioning problems.

SDP relaxation of GP. We now derive a semidefinite relaxation for GP. For simplicity, we illustrate it only for k-$MinC$; k-$MaxC$ follows similarly with the appropriate change to the objective.

The first step in SDP relaxation involves replacing $X^t L X$ with tr LY, where Y is equal to XX^t; this *linearizes* the objective. Let us define the set $\mathcal{T}_k$:

$$\mathcal{T}_k := \{Y : \exists X \in \mathcal{F}_k \text{ such that } Y = XX^t\}.$$

Thus k-$MinC$ reads: $MinC_k^* := \min\{\text{tr } LY \ : \ Y \in conv(\mathcal{T}_k)\}$.

Note that due to linearization of the objective, our feasible set can be rewritten as the convex hull of the original set $\mathcal{T}_k$. The next step is to approximate the convex hull of $\mathcal{T}_k$ by outer approximations that can be handled efficiently. Karisch and Rendl [1998] describe a nested sequence of outer approximations for GP that leads from the well-known eigenvalue bound of Donath and Hoffman to increasingly accurate bounds. Omitting details, one of their relaxation schemes results in the following SDP relaxation for k-$MinC$:

$$(P) \quad \begin{cases} \underline{\max} & \frac{1}{2}\text{tr } LY \\ \underline{\text{s.t.}} & \text{diag}(Y) = e_n \\ & Ye_n = me_n \\ & Y \geq 0 \quad \text{elementwise} \\ & Y \succeq 0,\ Y = Y^t \end{cases}$$

(P) is an SDP and can be solved by an interior-point solver such as SeDuMi.

4.5.2.3 Finding a closest feasible solution

While in some cases a bound is the major goal of a relaxation, in our case we require that the relaxation give us a (feasible) solution. In particular, the optimal solution of problem (P) is in general not feasible for the original GP problem, and we need to recover from the approximate solution a closest feasible solution, X, to the original GP problem. We use the following scheme in this section.

- From the relaxed solution Y, find a decomposition $Y = X'X'^t$ via SVD (note that X' is usually full rank rather than of rank k as in the feasible case).
- Treat each row in X' as a point in $\mathbb{R}^n$; cluster these points using a variant of the standard K-means algorithm that finds equi-size clusters. (We use multiple restarts and pick the result with the best cut value).
- Complete the feasible index matrix X: $x_{ij} = 1$ iff row i of X' gets assigned to cluster j.

This rounding scheme is related to the randomized projection heuristic studied by Goemans and Williamson [1995] in their work on Max-Cut. In this approach, the label (-1 or +1) of each vector is chosen according to whether the vector is above or below a randomly chosen hyperplane passing through the origin. Frieze and Jerrum [1995] generalized this scheme to max k-cut. Rendl and Wolkowicz [1995] proposed another alternative involving a first-order Taylor expansion of the cost function around the relaxed X'. However, these schemes make it difficult to enforce size constraints on the clusters, and occasionally produce artifacts such as having an empty cluster. Empirically, we have found that a K-means heuristic usually leads to superior and often near-optimal results.

4.5.3 Experimental Results

In this section, we combine graph partitioning with the GMF algorithm to perform inference on randomly generated undirected graphical models with singleton and pairwise potentials. We analyze three aspects of the overall procedure—the quality of the graph partition, the accuracy of the approximate marginal probabilities, and the tightness of the lower bounds on the log partition function.

For each trial, we use a random graph of 24 nodes [5] and specify the distribution $p(\mathbf{x}|\theta)$ by making a random choice of the model parameter θ from a uniform distribution $\mathcal{U}(a, b)$. For single node weights θ_i, we set $a = -w_{\text{obs}}$ and $b = -w_{\text{obs}}$. For pairwise weights θ_{ij}, we set $a = -w_{\text{coup}}$, $b = 0$ for *repulsive* coupling; $a = -w_{\text{coup}}$, $b = w_{\text{coup}}$ for *mixed* coupling; and $a = 0$, $b = w_{\text{coup}}$ for *attractive* coupling, respectively.

Table 4.5: GP performance. (default: K-means rounding; rp: random projection rounding)

	Equi-MinCut			Equi-MaxCut				
k	lower-b	feas. X	f/b	upper-b	feas. X	f/b	feas. X (rp)	f/b (rp)
3	34	38	1.10±0.03	78	75	0.96±0.02	71	0.91±0.04
4	41	45	1.09±0.02	82	80	0.97±0.02	74	0.90±0.04
6	52	55	1.06±0.03	83	81	0.97±0.01	77	0.93±0.02
8	59	61	1.03±0.02	83	82	0.99±0.01	79	0.95±0.02
3	73	77	1.05±0.02	122	119	0.97±0.01	113	0.92±0.03
4	86	90	1.05±0.02	135	130	0.97±0.01	122	0.91±0.03
6	104	207	1.03±0.01	140	137	0.98±0.01	128	0.91±0.02
8	116	118	1.02±0.01	140	138	0.99±0.01	131	0.93±0.01

4.5.3.1 Partitioning random graphs

Our graphs are generated by sampling an edge with probability p for each pair of nodes. Table 4.5 summarizes the performance (over 100 trials) of various graph partition schemes on random graphs. To assess performance, we compute the ratio f/b between the feasible cut that was found and the bound on the optimal cut provided by the SDP relaxation (the optimal solution must fall between f and b). In the top panel, we show results for partitioning unweighted graphs with $p = 0.3$ into

[5] In fact, a standard SDP solver can readily handle larger graphs (e.g., with more than 100 nodes). But the exact solutions of the singleton marginals for larger graphs are very expensive to compute, which makes it difficult to obtain good estimates of the inference error.

$k = 3, 4, 6$, and 8 clusters. The bottom panel shows results for partitioning denser unweighted graphs with $p = 0.5$. Partitioning on weighted graphs show similar performance.

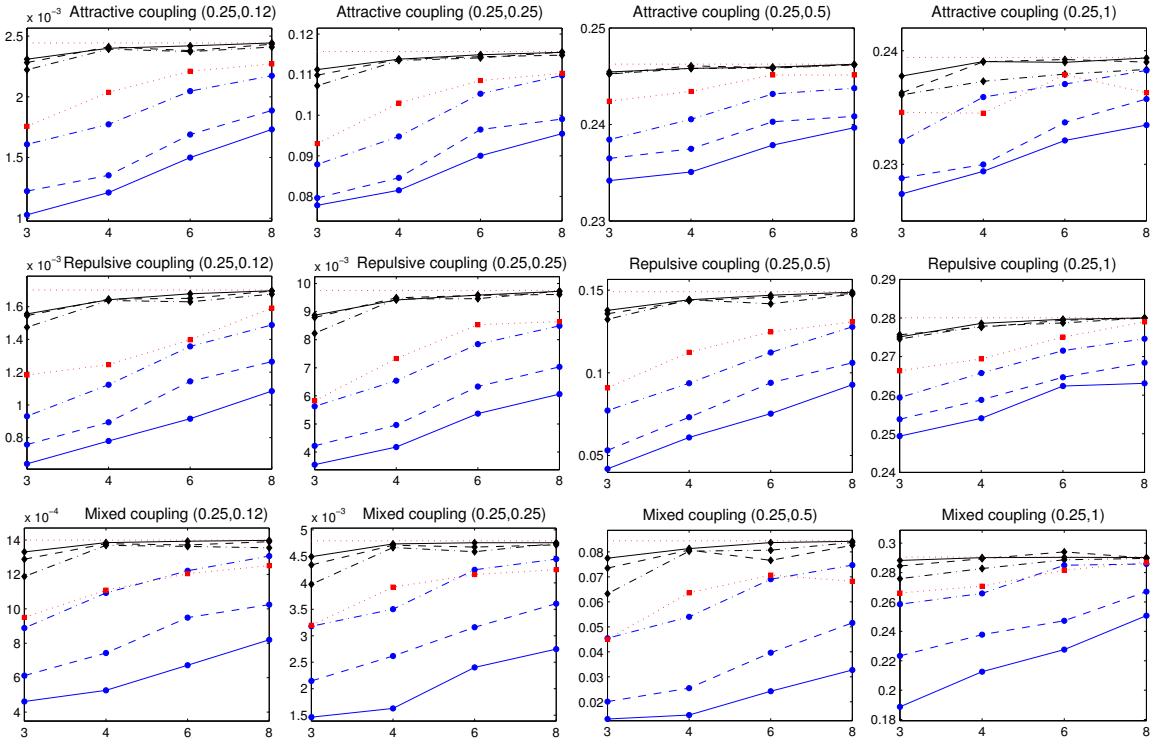


Figure 4.15: L_1 errors of singleton marginals on random graphs, with different coupling types and strengths. Each experiment is based on 20 trials. The sampling ranges of the model parameters for each set of trials are specified on top of each graph as $(w_{\text{obs}}, w_{\text{coup}})$. ($x$-axis: the number of clusters; y-axis: the ℓ_1 error; solid lines: cut based on θ_{ij}-weighted A; dashed lines: cut based on unweighted A; dashed-dot lines: cut based on $1/\theta_{ij}$-weighted A; lines with diamond symbols: equi-MaxCut (black); lines with round-dot symbols: equi-MinCut (blue); dotted line with square symbols: random cut (red). For reference, the dotted (red) line with no symbol marks the baseline error of naive mean field.)

We see that the SDP-based GP provides very good and stable partitioning results, usually no worse than 10% off the optimal cut values, and often within 5%. Note also that the K-means rounding scheme outperforms the random projection rounding (rp).

4.5.3.2 Single-node marginals

We compared the performance of GMF using different graph partition schemes with regard to the accuracy on single-node marginals. We used all six GP strategies summarized in Table 4.4, as well as a random clustering scheme. To assess the error, we use an L_1-based measure as described in the

last section. The exact marginals are obtained via exhaustive enumeration. We used graphs of two different densities in our experiments: *moderately connected* graphs, with treewidth 12 ± 1, more than an order of magnitude greater than the largest cluster to be formed; and *densely connected* graphs, with treewidth 16 ± 1. For simplicity, we show only results for the moderately connected graphs.

Figure 4.15 shows that for all variable clusterings, GMF almost always improves over the naive mean field. As expected, equi-MinCut always provides better results than other partition strategies. In particular, equi-MinCut based on coupling strength yields the best results (consistent with Theorem 5), followed by equi-MinCut based on node degree, then equi-MinCut that cuts the least number of heavy edges. This suggests that, to better approximate the true marginals, it is important to capture strong couplings within clusters. Equi-MaxCut fares less well; indeed, it is worse than a random cut in most cases. It is worth noting, however, that cutting lightweight edges (i.e., maximizing the sum of $\frac{1}{\theta_{ij}}$ across clusters) leads to better performance than degree- or coupling-based cuts.

Not surprisingly, the performance of GMF improves as the size of the clusters increase, which allows more dependencies to be captured within each cluster.

For denser graphs (results not shown), the performance gap between different clustering schemes becomes smaller, but the trend and the relative order remain the same.

4.5.3.3 Bounds on the log partition function

Figure 4.16 shows the lower bounds on the log partition functions given by the GMF approximations. Comparing to Figure 4.15, we see that there is a good correspondence between the performance on approximating marginals and the tightness of the lower bound, a reassuring result in the context of mean-field algorithms.

In summary, our empirical results provide rather clear support for a weighted version of MinCut as a useful clustering algorithm for GMF inference, which is consistent with the implications from

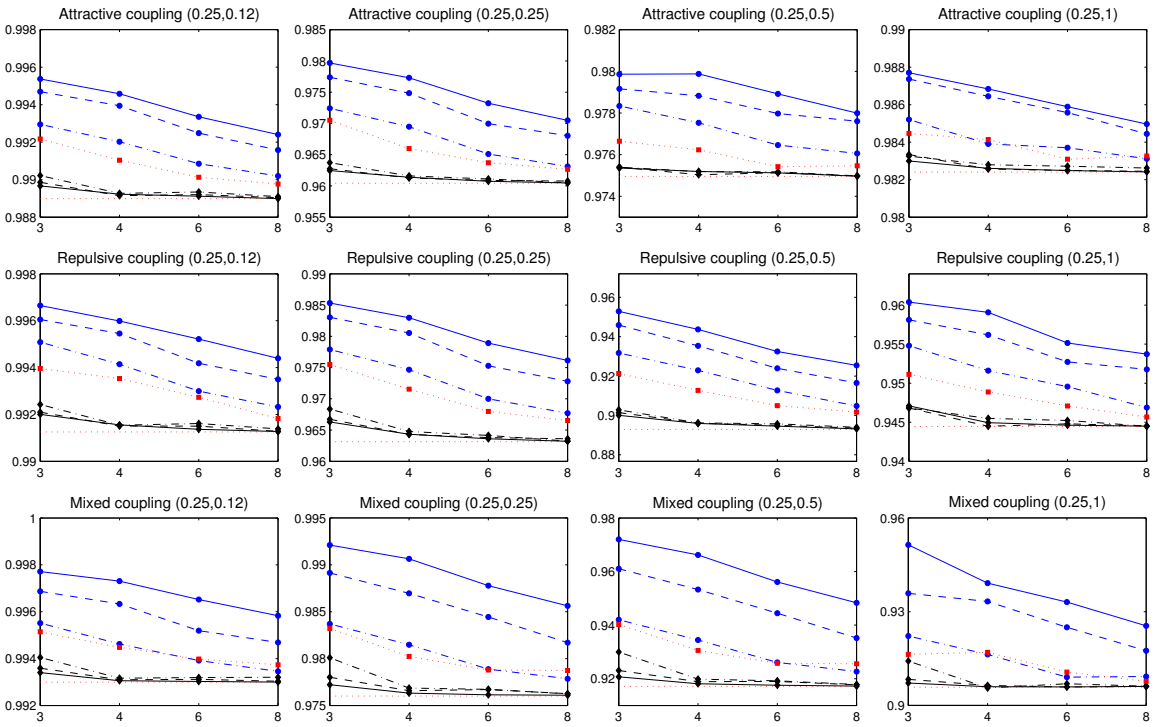


Figure 4.16: Accuracy of the lower bound on the log partition function. The ordering of the panels and the legends are the same as in Fig 4.15, except that the y-axis now corresponds to the ratio of the lower bound of the log partition function due to GMF versus the true log partition function.

the formal analysis. This combination of graph partitioning algorithms with the generalized mean field inference algorithm manifests a promising prototype for an autonomous variational inference algorithm for arbitrary graphical models, optimizing variational approximations over the space of model parameters as well as over the choice of tractable families used for the variational approximation, and making it possible to perform distributed approximate inference on large-scale network models arising from challenging problems in fields such as systems biology and sensor networks.

4.6 Extensions of GMF

In light of the foregoing exposition, there are a number of extensions of the research reported here that potentially lead to further improved GMF approximation.

4.6.1 Higher Order Mean Field Approximation

One possible extension involves the use of higher-order expansions in the basic variational bounds. Leisink and Kappen [2001] have shown how to upgrade first-order variational bounds such as that shown in Eq. (4.7) to yield higher-order bounds. In particular, the following third-order lower bound can be obtained for the likelihood:

$$p(\mathbf{x}_E) \geq \int d\mathbf{x} \exp\big\{ -E'(\mathbf{x}_H)\big\}\Big[1 - \Delta + \frac{1}{2}\exp(\xi)\Delta^2\Big],$$

where $\xi = \frac{1}{3}\langle\Delta^3\rangle/\langle\Delta^2\rangle$, $\Delta = E(\mathbf{x}_H, \mathbf{x}_E) - E'(\mathbf{x}_H)$, and $\langle\cdot\rangle$ denotes expectation over the approximate distribution $q(\mathbf{x}_H) = \exp\{-E'(\mathbf{x}_H)\}$. The optimizer of this lower bound cannot be found analytically. However, we can compute the gradient of the lower bound with respect to E'_i (assuming a cluster-factorized approximate distribution), which requires computation of up to third-order cumulants of the nodes in the bordering cliques in the subgraph. Leisink and Kappen [2001] reported an application of such a strategy to the 2-D lattice model and sigmoid belief network, approximated by a completely disconnected subgraph, and reported significantly improved bounds. In the GMF setting, which uses an approximating subgraph with more structure, the computation of the gradient is even simpler because fewer nodes are involved in the cumulant calculation.

4.6.2 Alternative Tractable Subgraphs

Another possible extension is to replace the disjoint clustering with a *tree-connected clustering*. The term W in the GMF bound can be also viewed as the total weight of the disrupted cliques (with respect to the original graph) in the subgraph underlying a GMF approximation. Thus, we may further reduce W by departing from the completely disjoint clustering to tree-connected clusters, in which we connect all the disjoint clusters resulting from a graph partition using a tree whose nodes are clusters. The link between every pair of connected clusters is chosen to be the maximally weighted clique shared by the clusters. Such a tree can be easily obtained by constructing a maximal spanning tree of variable clusters. The motivation of using tree-connected clusters rather than arbitrary subgraphs to approximate the true joint distribution is that under such a subgraph, the message-passing-based GMF algorithm described earlier is still guaranteed to converge and yield a set of globally consistent approximate cluster marginals.

4.6.3 Alternative Graph Partitioning Schemes

Eq. (B.5) in the appendix suggests that it may be advantageous to use other weighting schemes, such as the entropy-like clique weights (expected potentials) $\langle \phi_\beta \rangle_q$, and seek a partition that minimizes the sum of expected cross-border potentials. Obviously, exact computation of the entropy-like weights requires the true joint distribution, and is thus infeasible. We may approximate the expected potential of each clique by replacing the true marginal distribution of the variables in the corresponding clique with a naive mean-field-like approximation to the marginal: $q(\mathbf{x}_{D_\beta}) \propto \exp\{\theta_\beta \phi_\beta(\mathbf{x}_{D_\beta}) | \mathcal{F}_\beta\}$ where $\mathcal{F}$ denotes mean field messages from neighboring cliques; this turns the computation of the expectation into a local computation. It is possible to use an algorithm that iterates between GMF (to update the marginal $q(\cdot)$) and GP (to update the partition). It would be also interesting to look at unequal partitions in MinCut, which allows modularities of the graph structure to be explored in a more flexible way (e.g., as we do in the following for the **LOGOS** model).

4.7 Application to the LOGOS Model

The generalized mean field theorem makes it straightforward to obtain the fixed-point equations of the variational approximation to a variety of probability distributions of practical interest. All that is needed is to decide on a subgraph and a variable clustering, to identify the Markov blanket of each cluster, and to plug in the mean fields of the Markov blanket variables according to Eqs. (4.23) (or more generally, the marginal potentials of the peripheral cliques of each cluster). As pointed out in Remark 1, since all the original intra-cluster dependencies are preserved in the mean field cluster marginal, probabilistic inference in the GMF approximate distribution is reduced to local and modular operations within each cluster. Hence, the overall inference problem is fully decomposed based on the variable clustering.

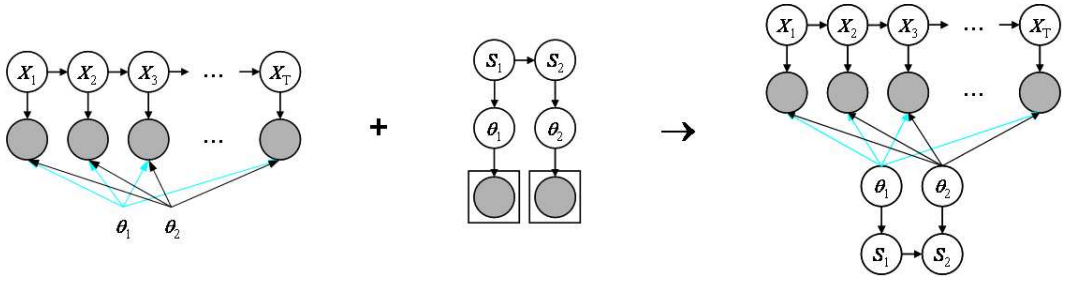


Figure 4.17: The modular structure of the **LOGOS** motif model.

For the **LOGOS** model developed in Chapter 2 for N sequences containing K types of motifs, the modularity of the model structure naturally suggests a bipartite variable clustering: a *motif cluster* $\{S_l^{(k)}, \theta_l^{(k)} \mid k = 1, \ldots, K, l = 1, \ldots, L_k\}$, and a *sequence cluster* $\{Y_t^{(n)}, X_t^{(n)} \mid n = 1, \ldots, N, t = 1, \ldots, T_n\}$ (Fig. 4.17). Applying Corollary 3, we obtain the following GMF cluster marginals:

$$q_s(\mathbf{x}) \propto \prod_{n=1}^{N} p(x^{(n)}|\upsilon, \Upsilon) p(y^{(n)}|x^{(n)}, \{\langle \ln \theta^{(k)} \rangle_{q_m}\}_{k=1}^{K}, \theta_{bg}), \tag{4.29}$$

$$q_m(\boldsymbol{\theta}, \mathbf{s}) = \prod_{k=1}^{K} p(s^{(k)}|\nu, \Omega) p(\theta^{(k)}|s^{(k)}, \alpha, \{\langle h^{(k)}(\mathbf{x}, \mathbf{y}) \rangle_{q_s}\}_{k=1}^{K}) \tag{4.30}$$

where $\langle h^{(k)}(\mathbf{x}, \mathbf{y})\rangle_{q_s}$ is the expectation of the sufficient statistics for motif k determined from DNA sequence set $\mathbf{y}$ by state sequences $\mathbf{x}$ (i.e., the count matrix of nucleotides of all sequence sites that are of motif k as specified by $\mathbf{x}$); and $\langle \ln \theta^{(k)}\rangle_{q_m}$ is the posterior means of the logarithms of the position-specific multinomial parameters of the motif k (often referred to as the natural parameters of the multinomial distribution). Note that $q_s(\mathbf{x})$ is now just a re-parameterized HMM, and $q_m(\boldsymbol{\theta}, \mathbf{s})$ is a re-parameterized HMDM model. Inference in both submodels is straightforward and inexpensive. For simplicity, again we omit the super(sub)scripts k and n in the following expositions, and give equations for a generic motif type or a generic sequence.

4.7.1 A GMF Algorithm for Bayesian Inference in LOGOS

Due to the isomorphism of GMF approximations of the cluster marginals to the original local and global submodels of **LOGOS** (Eqs. (4.29∼4.30)), variational Bayesian inference on **LOGOS** can be "divided and conquered" into coupled local inferences on: 1) the isolated local alignment model, i.e., an HMDM, as if we had "observations", $\bar{h} = \langle h(\mathbf{x}, \mathbf{y})\rangle_{q_s}$, to obtain the posterior distribution of the PWM of each motif; and 2) the isolated global distribution model, i.e., an HMM, as if the position-specific multinomial parameters of the motifs, in the natural parameter form $\bar{\phi}(\theta) = \langle \ln \theta\rangle_{q_m}$ were given, to compute the posterior probabilities of motif locations. This gives rise to the following EM-like fixed-point iteration procedure (referred to as a "variational EM" algorithm in [Ghahramani and Beal, 2001], although strictly speaking the analogy is only procedural but not mathematical, because GMF is not doing maximal likelihood parameter estimation as in an EM algorithm but Bayesian estimation.), which is a special case of the GMF algorithm in §4.4.3:

Variational "E" step: Compute the expected sufficient statistics, the count matrix $\bar{h}$, via inference in the global motif distribution model given $\bar{\phi}(\theta)$ and sequence y:

$$\bar{h} = \sum_{t=1}^{T-L+1} h(y_{t:t+L-1}) p(X_t = 1 | y, \bar{\phi}), \tag{4.31}$$

where $p(X_t = 1|y)$ is the posterior probability of the indicator at position t being the motif-start state, which can be computed using the forward-backward algorithm. (See Appendix A.3

for details.)

Variational "M" step: Compute the posterior mean of the natural parameter, $\bar{\phi}(\theta)$, via inference in the local motif alignment model given $\bar{h}$:

$$\begin{aligned}\bar{\phi}(\theta_{l,j}) &= \int_{\theta} \sum_{s_l} \ln \theta_{l,j} p(\theta_l | s_l, \alpha, \bar{h}) p(s_l | \bar{h}) d\theta_l \\ &= \sum_{i=1}^{I} p(S_l = i | \bar{h}) \big(\Psi(\alpha_{i,j} + \bar{h}_{l,j}) - \Psi(|\alpha_i| + |\bar{h}_l|) \big), \end{aligned} \tag{4.32}$$

where $\Psi(x)$ is the digamma function, and $p(S_l = i|\bar{h})$ is the posterior probability of hidden state q given 'observation' $\bar{h}$, which can be computed using the standard forward-backward algorithm of HMM. (See Appendix A.4 for details.)

According to Theorem 4, this message-passing procedure will converge. Once it converges, we can compute the MAP estimate of motif locations in the global HMM submodel and the Bayesian estimate of the motif PWMs from the local HMDM submodels.

The generalized mean field theory provides a *divide-and-conquer* computational tool to work with complex models, especially for those coming from a modular design using the graphical model formalism. It provides computational support for an upgrade path toward more sophisticated models, which may be needed for improving motif detection. For example, the global distribution model is completely open to user design and can be made highly sophisticated to model complex properties of multiple motifs without complicating the inference in the local alignment model. Similarly, the local motif alignment model can also be more expressive without interfering with the motif distribution model. In the literature, Bayesian inference in large scale models are usually approached via a Monte Carlo sampling algorithm. In Chapter 5 we describe a *collapsed* Gibbs sampling procedure for Bayesian inference on **LOGOS**. Following is an illustration of the convergence behavior of the GMF algorithm on **LOGOS** and an empirical comparison of the GMF algorithm and the Gibbs sampling algorithm on **LOGOS** for motif detection tasks of modest difficulty.

4.7.2 Experimental Results

We use semi-realistic test datasets described before, each containing 20 artificially generated DNA sequences (500-600 bp long) harboring one real motif or three different real motifs (of length 18, 22, and 26 bases, respectively). The performance of inference is evaluated based on the error rate ((false positive + false negative)/2) of predicted motif occurrences.

4.7.2.1 Convergence behavior of GMF

Since the GMF algorithm is only guaranteed to convergence to a local minimum, we run GMF with 50 random restarts, each followed by fixed-point (FP) iterations until convergence. To obtain a "convergence curve" of a full run of multiply restarted GMF, we sequentialize the output of all rounds of FP iterations. After each single cross-update step in each single round of FP-iteration, we record the lowest value of the free energy, $\langle E \rangle_q + H_q$, achieved so far (since the first round of FP iteration), and compute the empirical error rates of motif prediction made from the GMF posterior q corresponding to the current lowest free energy, which gives a performance trace.

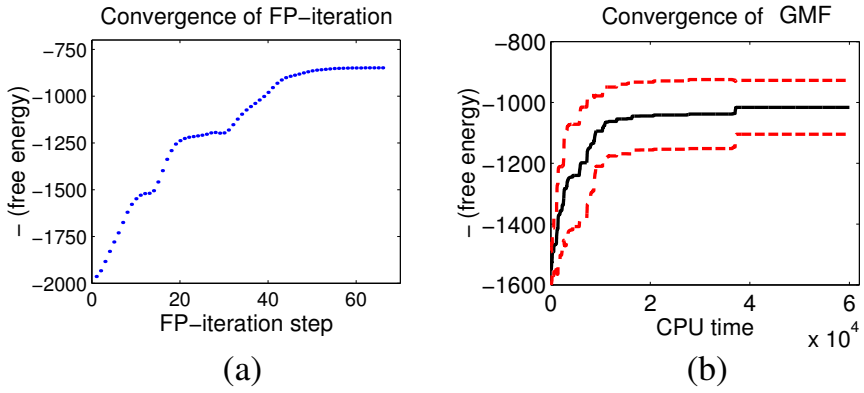


Figure 4.18: (a) Convergence of a single round of FP-iteration of GMF (Each point represents one step of iteration.) (b) The "convergence curve" of GMF with 50 random re-starts. (The solid line is the mean value over 10 independent runs, and the dashed line represents the std.)

Figure 4.18 illustrates the convergence behavior of GMF on a motif detection task. Typically, a single round of GMF takes about $30 \sim 60$ iterations to converge (Fig 4.18a). GMF with multiple random restarts in general plateaus within less then 50 restarts (Fig 4.18b), suggesting a possibility

of reaching global (or near global) optimality. Thus, in the following experiments, we perform GMF with 50 random restarts and pick the one resulting in the lowest free energy for the given sequences as the final result.

4.7.2.2 A comparison of GMF and the Gibbs sampler for motif inference

We compared the performance of motif inference on the **LOGOS** model using GMF and a Gibbs sampler (see §5.2). Convergence of the Gibbs sampler is diagnosed based on the standard Gelman-Rubin (GR) statistics [Gelman, 1998]. We infer motif locations using the sample means of X during Gibbs sampling, which yield an on-line measure of the performance.

Table 4.6: Median hit-rate of motif detection in test set containing one genuine and one decoy motif.

	abf1	gal4	gcn4	gcr1	mat	mcb	mig1	crp
GMF	0.81	0.82	0.71	0.65	0	0.	0.73	0.58
Gibbs	0.71	0.79	0.65	0.53	0.75	0	0.91	0.63

Table 4.6 summarizes the results obtained via GMF and Gibbs sampling for motif detection in a simple one-per-sequence setting using HMDM as the local model (see §2.4.4). The performances of the two algorithms are largely comparable, with the Gibbs sampler slightly better. However, the convergence time for the Gibbs sampler is significantly longer, typically 5 to 10 times that of GMF.

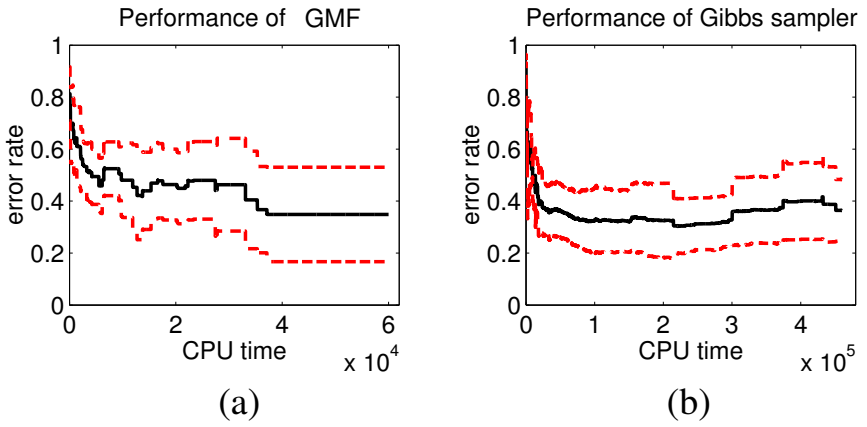


Figure 4.19: Comparison of GMF and Gibbs sampler on performance (in terms of predictive error rate vs. time). Note the difference in the scale of the x-axis in the two plots.

For more difficult problems, such as simultaneous detection of multiple motifs in a large dataset, the mixing time of the Gibbs sampler becomes prohibitively long, and the results obtained within

a tolerable time span from a Gibbs sampler are not comparable to those of the GMF, which uses far less time. Figure 4.19 illustrates the convergence curve, in terms of predictive error rate versus time, for GMF and the Gibbs sampler (obtained from the same experiment from which we plotted the convergence curve in Fig. 4.18). As evident in Fig 4.19a, the error rate of motif detection using GMF generally follows an improving trend consistent with that of the free energy in Fig. 4.18b, although not exactly monotonically decreasing, which is not surprising since the generative model described by **LOGOS** does not necessarily model the motif sequences exactly (thus some local optima may yield slightly better predictions than others). The error curve of the Gibbs sampler, on the other hand, is less stable (Fig 4.19b), showing that the sampling process explores the state space in a non-deterministic fashion, therefore providing less reliable performance in finite time. The choice of random seeds seems to affect convergence quality for both GMF and Gibbs.

Table 4.7: Performance (mean error rate) of GMF and Gibbs over 5 test datasets.

dataset	1	2	3	4	5
GMF	0.27±0.17	0.26±0.13	0.38±0.18	0.35±0.18	0.39±0.17
Gibbs	0.49±0.19	0.41±0.23	0.56±0.20	0.41±0.23	0.49±0.21

Table 4.7 summarizes the the performances of GMF and the Gibbs sampler over 5 different test datasets for simultaneous detection of three motifs (as described in §2.6.1). GMF outperforms the Gibbs sampler (run with finite allowable time, i.e. $10\times$ the time for GMF) in all cases. We reason that for a complex motif model such as **LOGOS**, the state space is likely to be highly multi-modal and poorly connected, and thus tends to trap the Gibbs sampler at sub-optimality; whereas GMF can explore such a space much more efficiently by doing more random restarts than a Gibbs sampler can afford, and is guaranteed to reach a local minimum from each restart.

4.8 Conclusions and Discussions

We have presented a generalized mean field approach to probabilistic inference in graphical models, in which a complex probability distribution is approximated via a distribution that factorizes over a disjoint partition of the graph. Locally optimal variational approximations are obtained via an algorithm that performs coordinate ascent on a lower bound of the log-likelihood, with guaranteed

convergence. For a broad family of models in practical use, we showed that the GMF approximations of the cluster marginals are isomorphic to the original model in the sense that they inherit all of its intra-cluster independence structure. Moreover, these marginals are independent of the rest of the model given the expected potential factors (mean fields) of the Markov blanket of the cluster. The explicit and generic formulation of the "mean fields" in terms of the Markov blanket of variable clusters also leads to a simple, generic message-passing algorithm for complex models. This result is somewhat surprising as it shows that we can do approximate inference for arbitrary subsets of hidden variables locally and tractably by capturing all the dependences external to the variable subset with an expected Markov blanket, and applying existing inference algorithms locally (i.e., on the cluster marginal) as a subroutine.

Disjoint clusterings have also been used in sampling algorithms to improve mixing rates for large problems. For example, the Swendsen-Wang algorithm [1987] samples the Ising (or Potts) model at critical temperatures by grouping neighboring nodes with the same spin value, thereby forming random clusters (of coupled spins) that are effectively independent of each other, allowing an MCMC process to collectively sample the spin of each cluster independently and at random. This method often dramatically speeds up the mixing of the MCMC chain. [Gilks *et al.*, 1996] also noted that when variables are highly correlated in the stationary distribution, blocking highly correlated components into higher-dimensional components may improve mixing. However, in the sampling framework, clusterings are usually obtained dynamically, based on the coupling strength rather than the topology of the network.

We have also investigated combinations of graph partitioning algorithms with the generalized mean field algorithm, which allows mean field approximations to be optimized over both parameter space and variable partition space in an autonomous fashion. We proved that the quality of the GMF approximation is bounded by the total absolute weight of the potentials of the disrupted cliques due to the disjoint variable clustering. Empirically, we confirmed that although all graphs partitions lead to improvement over a naive mean field approximation, a minimal cut equipartition clearly yields the best GMF approximation, measured both by singleton marginals and lower bounds of

the true log partition function. Moreover, there is a good association between the qualities of the approximate marginals and lower bounds.

Our work represents an initial foray into the problem of choosing clusters for cluster-based variational methods. There is clearly much more to explore. First, we should note that we are far from the ideal approach, where we base the clustering criterion on the ultimate goal—that of obtaining accurate estimates of marginal probabilities. This is of course an ambitious goal to aim for, and in the near term it seems advisable to attempt to find effective surrogates. In particular, we do not want the problem of choosing clusters to be as computationally complex as the inference problem that we wish to solve! (Fortunately, many efficient solvers are available to solve the GP problem nearly optimally via SDP or spectral relaxation.) We should consider surrogates that involve more general combinations of parameter values along cuts. In particular, we found little support for the use of maximum cuts in our experiments, but perhaps if we search for large cuts along which the parameter values are uniformly small we will have more success in this regard. In general, we might ask for a surrogate that aims to capture both the setting under which mean field approximations are effective, and the setting under which important local dependencies can be treated tractably.

Note also that we have focused on partitioning methods that decompose a large graphical model into clusters of equal size. With no prior knowledge of the local connectivity within the clusters, this equal-size heuristic seems reasonable; we wish to distribute resources roughly equally to each cluster (e.g., to balance the load in a parallel computing setting). Again, however, it would be useful to explore surrogates that attempt to capture local connectivity in the clustering criterion.

In an exemplary biological motif detection problem involving Bayesian inference in a hybrid, large-scale graphical model, GMF outperforms conventional Gibbs sampling methods in both convergence speed and error rate. We believe that due to its flexibility and efficiency, GMF simplifies the application of variational methods to general probabilistic inference, and can significantly increase the expressive power of languages that can be considered “practical” for knowledge representation and reasoning under uncertainty.

Chapter 5

Probabilistic Inference II: Monte Carlo Algorithms

Monte Carlo algorithms are based on the fact that while it may not be feasible to perform statistical computations on a complex joint or posterior distribution, say, $p(x)$, it may be possible to obtain samples from $p(x)$, or from a closely related distribution, such that marginals and other expectations can be approximated using sample-based averages. In contrast to the variational inference approaches discussion in the previous chapter, which seek deterministic approximations to $p(x)$, Monte Carlo algorithms yield a stochastic representation of $p(x)$ that is asymptotically exact and easy to apply. General-purpose Monte Carlo inference software such as the BUGS system has been developed for use with a general-purpose statistical modeling language (see [Gilks *et al.*, 1996]). For some statistical models, such as the Dirichlet process mixture model for haplotypes and non-parametric Bayesian models in general, although the variational approach has been vigorously pursued [Blei and Jordan, 2004], so far Monte Carlo algorithms are still the only practical approach to yield reliable performance.

5.1 A Brief Overview of Monte Carlo Methods

We discuss two examples of Monte Carlo algorithms—Gibbs sampling and the Metropolis-Hastings algorithm—that are commonly used in the graphical model setting and in particular within the Bayesian paradigm.

Gibbs sampling is an example of a Markov chain Monte Carlo (MCMC) algorithm. In an MCMC algorithm, samples are obtained via a Markov chain whose stationary distribution is the desired $p(\mathbf{x})$ (typically a complex multivariate distribution). The state of the Markov chain is an assignment of a value to each of the variables. After a suitable "burn-in" period so that the chain approaches its stationary distribution, these states are used as samples.

The Markov chain for the Gibbs sampler is constructed in the following way: 1) at each step one of the variables X_i is selected (at random or according to some fixed sequence); 2) the conditional distribution $p(x_i|\mathbf{x}_{\mathcal{V}\setminus i})$ is computed (recall that $\mathcal{V}$ is the set of indices of all the variables in a graphical model $G(\mathcal{V},\mathcal{E})$); 3), a value x_i is sampled from this distribution; and 4) the sampled x_i replaces the previous value of the ith variable.

The Markov properties of graphical models are particularly useful for a Gibbs sampler: conditioning on the so-called Markov blanket of a given node renders the node independent of all other variables. Therefore, $p(x_i|\mathbf{x}_{\mathcal{V}\setminus i}) = p(x_i|\mathbf{x}_{\mathcal{MB}_i})$. Thus, the implementation of Gibbs sampling reduces to the computation of the conditional distributions of individual variables given their Markov blankets. For graphical models, these conditionals take the following form:

$$\begin{aligned} p(x_i|\mathbf{x}_{\mathcal{V}\setminus i}) &= p(x_i|\mathbf{x}_{\mathcal{MB}_i}) \\ &= \frac{\prod_{\alpha\in\mathcal{MBK}_i}\phi_\alpha(\mathbf{x}_{D_\alpha})}{\sum_{x_i}\prod_{\alpha\in\mathcal{MBK}_i}\phi_\alpha(\mathbf{x}_{D_\alpha})} \end{aligned} \tag{5.1}$$

where $\mathcal{MBK}_i$ denotes the set of cliques containing X_i **and** its Markov blanket nodes (note the difference of $\mathcal{MBK}_i$ and $\mathcal{MBC}_i$ defined in Chapter 4). The set $\mathcal{MBK}_i$ is often much smaller than the set $\mathcal{D}$ of all cliques of G, and in such cases each step of the Gibbs sampler can be implemented efficiently. Indeed the computation of the conditionals often takes the form of a simple message-passing algorithm that is reminiscent of the junction tree algorithm or the GMF algorithm.

When the computation in Eq. (5.1) is overly complex, the Metropolis-Hastings algorithm can provide an effective alternative. Metropolis-Hastings is an MCMC algorithm that is not based on conditional probabilities, and thus does not require normalization as in Eq. (5.1). Given the current state x of the algorithm, Metropolis-Hastings chooses a new state x^* from a "proposal distribution"

$q(x^*|x)$, which often simply involves picking a variable X_i at random and choosing a new value for that variable, again at random. The algorithm then computes the "acceptance probability":

$$\xi = \min\left(1, \frac{q(x|x^*)}{q(x^*|x)} \frac{\prod_{\alpha\in\mathcal{A}} \phi_\alpha(\mathbf{x}^*_{D_\alpha})}{\prod_{\alpha\in\mathcal{A}} \phi_\alpha(\mathbf{x}_{D_\alpha})}\right). \tag{5.2}$$

With probability ξ the algorithm accepts the proposal and moves to x^*, and with probability $1-\xi$ the algorithm remains in state x. For graphical models, if only one of the variables (say X_i) is resampled, this computation also turns out to often take the form of a simple message-passing algorithm, of which samples of the Markov blanket of X_i can be regarded as the *message*.

The principal advantages of Monte Carlo algorithms are their simplicity of implementation and their generality. Under weak conditions, the algorithms are guaranteed to converge. A problem with the Monte Carlo approach, however, is that convergence times can be long (e.g., see §4.7.2), and it can be difficult to diagnose convergence.

5.2 A Gibbs Sampling Algorithm for LOGOS

In the last chapter, we described a GMF algorithm for variational Bayesian inference for *de novo* motif detection under the **LOGOS** model, which deterministically approximates the posterior distribution of motif locations and the Bayesian estimates (resulted from an integration over the posterior distribution) of PWMs. Here we present a Gibbs sampling algorithm for the same tasks. A comparison of its performance to that of the GMF algorithm was given in §4.7.2.

5.2.1 The *Collapsed* Gibbs Sampler

Given a set of DNA sequences denoted by $\mathbf{y} = \{y^{(n)}\}_{n=1}^N$, where $y^{(n)} = (y_1^{(n)}, \ldots, y_{T_n}^{(n)})$, a Gibbs sampler periodically samples the state configurations of latent variables from variable sets $\mathbf{X} = \{X^{(n)}\}_{n=1}^N$, $\mathbf{\Theta} = \{\theta^{(k)}\}_{k=1}^K$ and $\mathbf{S} = \{S^{(k)}\}_{k=1}^K$, one at a time, conditioning on the state configurations of the rest of the variables sampled during the previous iterations. Again, for simplicity we drop in the sequel the superscript n associated with variables X, Y, and the superscript k associated with variables θ, S. The predictive distributions to be derived for sampling apply to every sequence and motif.

In principal, a standard data augmentation (DA) [Tanner and Wong, 1987] approach can be used to solve the Bayesian missing data problem for motif detection under **LOGOS**. But the fact that θ is a high-dimensional continuous variable implies that it is very expensive to approximate its posterior distribution by samples and also, the Markov chain that generates these samples can mix very slowly. As pointed out in Liu [1994], the conjugacy between $p(\theta, \mathbf{s})$ and $p(\mathbf{x}, \mathbf{y}|\theta)$ suggests that we can integrate out $\boldsymbol{\theta}$ and derive a *collapsed* Gibbs sampling scheme. Essentially, we sample iteratively from only two sets of discrete variables in the Markov chain: the Dirichlet component indicator sequence $S = (S_1, \ldots, S_L)$ in the local HMDM model, and the motif location indicator sequence $X = (X_1, \ldots, X_T)$ in the global HMM model.

Let l denote an arbitrary state taken by X_t (i.e, column l of a motif whose index is omitted) and, and h denote the sufficient statistics of the PWM θ (i.e., the nucleotide count matrix of the aligned instances of each motif). For convenience, we use the subscript $[-t]$ to denote an index set excluding the tth element for variables, or an indication of the source (i.e., all but the tth element) from which a sufficient statistic is collected. To keep the exposition simple, in the sequel we focus on a Bayesian treatment of the PWMs only, and let the transition probability matrices $\{\Omega_{i,j}\}$ and $\{\Upsilon_{i,j}\}$ in the global and local model, respectively, be constant. A Bayesian treatment of these parameters (e.g., Ω) was discussed in §3.2, and can be similarly implemented in the collapsed Gibbs sampler. Given the current states of all (discrete) variables in the **LOGOS** model except X_t, the Bayesian conditional predictive distribution for X_t is:

$$
\begin{aligned}
& p(X_t = l|\mathbf{x}_{[-t]}, \mathbf{s}, \mathbf{y}) \\
= \;& p(X_t = l|x_{t-1}, x_{t+1}, y_t, h_{[-t]}, \mathbf{s}) \\
= \;& \frac{1}{Z} p(X_t = l|x_{t-1}, x_{t+1}) p(y_t|X_t = l, h_{[-t]}, s_l) \\
= \;& \frac{1}{Z} \Upsilon_{x_{t-1} \to l} \Upsilon_{l \to x_{t+1}} \frac{\Gamma(|\alpha_{s_l}|)}{\prod_{j \in \mathbb{N}} \Gamma(\alpha_{s_l,j})} \int \prod_{j \in \mathbb{N}} \theta_j^{\alpha_{s_l,j} + h_{[-t],l,j} + \delta(y_t,j) - 1} d\theta_j \\
= \;& \frac{1}{Z} \Upsilon_{x_{t-1} \to l} \Upsilon_{l \to x_{t+1}} \frac{\Gamma(|\alpha_{s_l}|)}{\prod_{j \in \mathbb{N}} \Gamma(\alpha_{s_l,j})} \cdot \frac{\prod_{j \in \mathbb{N}} \Gamma(\alpha_{s_l,j} + h_{[-t],l,j} + \delta(y_t, j))}{\Gamma(|\alpha_{s_l} + h_{[-t],l} + 1|)} \\
= \;& \frac{1}{Z} \Upsilon_{x_{t-1} \to l} \Upsilon_{l \to x_{t+1}} \frac{\Gamma(|\alpha_{s_l}|)}{\prod_{j \in \mathbb{N}} \Gamma(\alpha_{s_l,j})} \cdot \frac{\prod_{j \in \mathbb{N}} \Gamma(\alpha_{s_l,j} + h_{l,j})}{\Gamma(|\alpha_{s_l} + h_l|)},
\end{aligned}
\tag{5.3}
$$

where h_l represents the count vector of column l of a motif (whose index is omitted) resulting from the current assignment of $\mathbf{x}_{[-t]}$ plus the count induced by x_t, and $\Upsilon_{\alpha\to\beta}$ denotes the transition probability from state α to β.

Given the current states of all variables except S_l, the Bayesian conditional predictive distribution of variable S_l is:

$$\begin{aligned} p(S_l = i|\mathbf{s}_{[-l]}, \mathbf{x}, \mathbf{y}) &= p(S_l = i|s_{l-1}, s_{l+1}, h_l) \\ &= \frac{1}{Z} p(S_l = i|s_{l-1}, s_{l+1}) p(h_l|s_l = i) \\ &= \frac{1}{Z} \Omega_{s_{l-1}\to i}\Omega_{i\to s_{l+1}} \frac{\Gamma(|\alpha_i|)}{\prod_{j\in\mathbb{N}}\Gamma(\alpha_{i,j})} \int \prod_{j\in\mathbb{N}} \theta_j^{\alpha_{i,j}+h_{l,j}-1} d\theta_j \\ &= \frac{1}{Z} \Omega_{s_{l-1}\to i}\Omega_{i\to s_{l+1}} \frac{\Gamma(|\alpha_i|)}{\prod_{j\in\mathbb{N}}\Gamma(\alpha_{i,j})} \cdot \frac{\prod_{j\in\mathbb{N}}\Gamma(\alpha_{i,j}+h_{l,j})}{\Gamma(|\alpha_i+h_l|)} \end{aligned} \tag{5.4}$$

A full sweep of variables X_n results in a new set of labellings of motif/background in a DNA sequence and requires $O(TKL)$ operations. The maximal *a posteriori* estimates of the motif locations are obtained by summarizing the empirical sample statistics.

5.2.2 Convergence Diagnosis

Since motifs are short stochastic substring patterns in a large "sea" of background sequences, the posterior distribution defined by **LOGOS** is not only very high-dimensional, but also likely to be multi-modal due to the possible presence of many genuine or pseudo motif patterns in the sequences. Such a distribution can cause very slow mixing for the Markov chain, as well difficulties in detecting when the stationary distribution is reached.

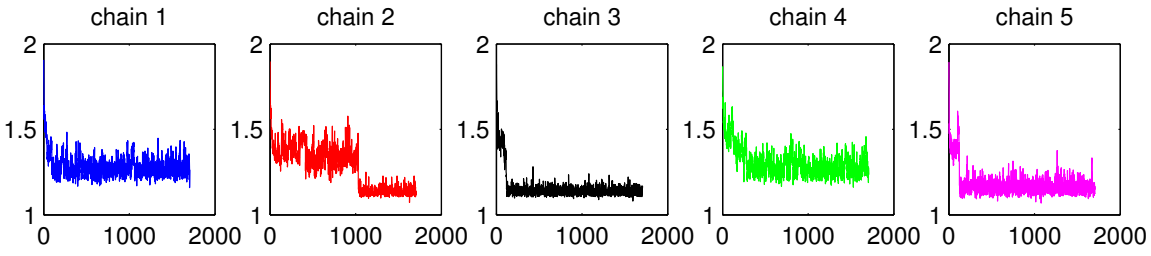


Figure 5.1: Multiple runs of Gibbs sampling, as traced by the column-average entropy of $\bar{h}$.

To increase the chance of proper mixing, M independent runs of sampling, with different random seeds, are simultaneously performed (Fig. 5.1). Convergence can be monitored at run-time using an on-line minimal pairwise Gelman-Rubin (GR) statistics [Gelman, 1998] of some scalar summaries of the model parameters obtained in each Markov chain. For **LOGOS**, two scalar summaries of the model parameters are used: 1) the posterior means of all the count matrices $\bar{h}$ [1]; 2) the column-average entropy of the column-normalized h (denoted by $\bar{\hbar}$): $Ent(\bar{\hbar}) = (\sum_{k=1}^{K}\sum_{l=1}^{L_k} H(\bar{\hbar}_l^{(k)}))/\sum_{k=1}^{K} L_k$, as suggested in [Lawrence *et al.*, 1993]. In the first case, $4 \times \sum_{k=1}^{K} L_k$ values (i.e., elements of $\bar{h}$) need to be monitored, and the minimal GR statistic (which is a matrix $\{GR(\bar{h}_{ij})\}$, containing the GR statistics of all the elements of $\bar{\hbar}$'s of a pair of chains) is computed as the GR statistic that has the minimal Frobenius norm (among all pairs of MCMC chains). To diagnosis convergence, we act conservatively by monitoring the maximum element in this minimal GR statistic matrix. For the second strategy, we just compute the GR statistics of the scalar summary $Ent(\bar{\hbar})$ for all possible pairs of MCMC chains. For both cases, we stop when the minimum among all pairwise GR statistics reaches $1 + \epsilon$, where ϵ is set to be a small scaler (e.g. 0.05). The rationale underlying this approach is that it is unlikely for identical suboptimal convergence to be reached by several independent MCMC chains before the optimum solution is found once.

A comparison of this Gibbs sampler with the GMF algorithm on motif detection was presented in §4.7.

5.3 Markov Chain Monte Carlo for Haplotype Inference

In this section, we describe a Gibbs sampling algorithm for exploring the posterior distribution under the Dirichlet process mixture model for haplotypes, including the latent ancestral pool. We also present a Metropolis-Hastings variant of this algorithm that appears to mix better in practice. We follow the notations used in Chapter 3 and hereby omit an reiteration of notational details.

[1] A simple matching heuristic is used to match the count matrices from different chains when different chains number the motifs differently (e.g., the same set of motifs $(1, 2, 3)$ found in chain 1 may be numbered $(3, 1, 2)$ in another chain). We use minimum discrepancy amount all permutations to find the best matching.

5.3.1 A Gibbs Sampling Algorithm

Recall that the Gibbs sampler draws samples of each random variable from a conditional distribution of that variable given (previously sampled) values of all the remaining variables. The variables needed in our algorithm are: C_{i_t}, the index of the ancestral template of a haplotype instance t of individual i; $A_j^{(k)}$, the allele pattern at the jth locus of the kth ancestral template; $H_{i_t,j}$, the t-th allele of the SNP at the jth locus of individual i; and $G_{i,j}$, the genotype at locus j of individual i (the only observed variables in the model). All other variables in the model—θ and γ—are integrated out. The Gibbs sampler thus samples the values of C_{i_t}, $A_j^{(k)}$ and $H_{i_t,j}$.

Conceptually, the Gibbs sampler alternates between two coupled stages. First, given the current values of the hidden haplotypes, it samples the c_{i_t} and subsequently $a_j^{(k)}$, which are associated with the Dirichlet process prior. Second, given the current state of the ancestral pool and the ancestral template assignment for each individual, it samples the $h_{i_t,j}$ variables in the basic haplotype model.

In the first stage, the conditional distribution of c_{i_t} is:

$$\begin{aligned}
& p(c_{i_t} = k \,|\mathbf{c}_{[-i_t]}, \mathbf{h}, \mathbf{a}) \\
\propto\ & p(c_{i_t} = k \,|\mathbf{c}_{[-i_t]}) \int p(h_{i_t} \,|c_{i_t} = k, \theta_k, a^{(k)}) p(\theta^{(k)} | \{h_{i'_{t'}} \ :\ i'_{t'} \neq i_t, c_{i'_{t'}} = k\}, a^{(k)}) d\theta^{(k)} \\
=\ & p(c_{i_t} = k \,|\mathbf{c}_{[-i_t]}) p(h_{i_t} \,|a^{(k)}, \mathbf{c}, \mathbf{h}_{[-i_t]}) \\
=\ & \begin{cases} \frac{n_{[-i_t],k}}{n-1+\tau} p(h_{i_t} | a^{(k)}, \mathbf{m}_{[-i_t],k}) & \text{if } k = c_{i'_{t'}} \text{ for some } i'_{t'} \neq i_t \\ \frac{\tau}{n-1+\tau} \sum_{a'} p(h_{i_t} | a') p(a') & \text{if } k \neq c_{i'_{t'}} \text{ for all } i'_{t'} \neq i_t \end{cases}
\end{aligned} \tag{5.5}$$

where $[-i_t]$ denotes the set of indices excluding i_t; $n_{[-i_t],k}$ represents the number of $c_{i'_{t'}}$ for $i'_{t'} \neq i_t$ that are equal to k; n represents the total number of instances sampled so far; and $\mathbf{m}_{[-i_t],k}$ denotes the sufficient statistics m associated with all haplotype instances originating from ancestor k, except h_{i_t}. This expression is simply Bayes theorem with $p(h_{i_t}|a^{(k)}, \mathbf{c}, \mathbf{h}_{[-i_t],})$ playing the role of the likelihood and $p(c_{i_t} = k \,|\mathbf{c}_{[-i_t]})$ playing the role of the prior.

The likelihood $p(h_{i_t}|a^{(k)}, \mathbf{m}_{[-i_t],k})$ is obtained by integrating over the parameter $\theta^{(k)}$, as in Eq. (3.9), up to a normalization constant:

$$p(h_{i_t}|a^{(k)}, \mathbf{m}_{[-i_t],k}) \quad \propto \quad R(\alpha_h, \beta_h) \frac{\Gamma(\alpha_h + m_{i_t,k}) \Gamma(\beta_h + m'_{i_t,k})}{\Gamma(\alpha_h + \beta_h + m_{i_t,k} + m'_{i_t,k})} \Big(\frac{1}{|B| - 1}\Big)^{m'_{i_t,k}}, \tag{5.6}$$

where $m_{i_t,k} = m_{[-i_t],k} + \sum_j \mathbb{I}(h_{i_t,j} = a_j^{(k)})$ and $m'_{i_t,k} = m'_{[-i_t],k} + \sum_j \mathbb{I}(h_{i_t,j} \neq a_j^{(k)})$, both functions of h_{i_t} (note that $m_{i_t,k} + m'_{i_t,k} = nJ$) [2]. It is easy to see that the normalization constant is the marginal likelihood $p(\mathbf{m}_{[-i_t],k} \mid a^{(k)})$, which leads to:

$$p(h_{i_t}|a^{(k)}, \mathbf{m}_{[-i_t],k}) = \frac{\Gamma(\alpha_h + m_{i_t,k})\Gamma(\beta_h + m'_{i_t,k})}{\Gamma(\alpha_h + m_{[-i_t],k})\Gamma(\beta_h + m'_{[-i_t],k})} \cdot \frac{\Gamma(\alpha_h + \beta_h + (n_k - 1)J)}{\Gamma(\alpha_h + \beta_h + n_k J)} \Big(\frac{1}{|B| - 1}\Big)^J. \tag{5.7}$$

For $p(h_{i_t}|a)$, the computation is similar, except that the sufficient statistics $\mathbf{m}_{[-i_t],k}$ are now null (i.e., no previous matches with a newly instantiated ancestor):

$$p(h_{i_t}|a) \quad = \quad R(\alpha_h, \beta_h) \frac{\Gamma(\alpha_h + m_{i_t})\Gamma(\beta_h + m'_{i_t})}{\Gamma(\alpha_h + \beta_h + J)} \Big(\frac{1}{|B| - 1}\Big)^{m'_{i_t}}, \tag{5.8}$$

where $m_{i_t} = \sum_j \mathbb{I}(h_{j,i_t} = a_j)$ and $m'_{i_t} = J - m_{i_t,k}$ are the relevant sufficient statistics associated only with haplotype instance h_{i_t}.

The conditional probability for a newly proposed equivalence class k that is not populated by any previous samples requires a summation over all possible ancestors: $p(h_{i_t}) = \sum_{a'} p(h_{i_t}|a')p(a')$. Since the gamma function does not factorize over loci, computing this summation takes time that is exponential in the number of loci. To skirt this problem we endow each locus with its own mutation parameter $\theta_j^{(k)}$, with all parameters admitting the same prior $\mathrm{Beta}(\alpha_h, \beta_h)$ [3]. This gives rise to a closed-form formula for the summation and also for the normalization constant in Eq. (5.5). It is also, arguably, a more accurate reflection of reality. Specifically,

$$\begin{aligned} p(h_{i_t}|a) \quad &= \quad \prod_j R(\alpha_h, \beta_h) \frac{\Gamma(\alpha_h + m_{i_t,j})\Gamma(\beta_h + m'_{i_t,j})}{\Gamma(\alpha_h + \beta_h + 1)} \Big(\frac{1}{|B| - 1}\Big)^{m'_{i_t,j}} \\ &= \quad \prod_j \Big(\frac{\alpha_h}{\alpha_h + \beta_h}\Big)^{\mathbb{I}(h_{i_t,j} = a_j)} \Big(\frac{\beta_h}{(|B| - 1)(\alpha_h + \beta_h)}\Big)^{\mathbb{I}(h_{i_t,j} \neq a_j)}. \end{aligned} \tag{5.9}$$

[2] Recall that in §3.3.2 we use the symbol m_k to denote the count of matching SNP alleles in those individual haplotypes associated with ancestor $a^{(k)}$ (and m'_k for those inconsistent with the ancestor $a^{(k)}$). Here, we use a variant of these symbols to denote the pair of random counts (as indicated by the additional subscript i_t) resulting from the original m_k (or m'_k) for individual haplotypes known to associate with $a^{(k)}$ plus a randomly assigned haplotype h_{i_t} (whose actual associated ancestor is unknown).

[3] Note that now we also need to split counts $m_{[-i_t],k}$, $m_{i_t,k}$ and m_{i_t} into site-specific counts, $m_{[-i_t],k,j}$, $m_{i_t,k,j}$ and $m_{i_t,j}$, respectively, where j denotes a single SNPs site.

Assuming that loci are also independent in the base measure $p(a)$ of the ancestors and that the base measure is uniform, we have:

$$\begin{aligned}\sum_a p(h_{i_t}|a)p(a) &= \prod_j \Big(\sum_{l\in B} p(a_j = l)p(h_{i_t,j}|a_j = l)\Big) \\ &= \prod_j \left(\sum_{l\in B} \frac{1}{|B|}\left(\frac{\alpha_h}{\alpha_h+\beta_h}\right)^{\mathbb{I}(h_{i_t,j}=l)}\left(\frac{\beta_h}{(|B|-1)(\alpha_h+\beta_h)}\right)^{\mathbb{I}(h_{i_t,j}\neq l)}\right) \\ &= \left(\frac{1}{|B|}\right)^J \end{aligned} \tag{5.10}$$

In this case (that each locus has its own mutation parameter), the conditional likelihood computed in Eq. (5.7) is:

$$\begin{aligned}& p(h_{i_t,j}|a_j^{(k)}, \mathbf{m}_{[-i_t],k,j}) \\ =\ & \prod_j \left(\frac{\alpha_h + m_{[-i_t],k,j}}{\alpha_h+\beta_h+n_k-1}\right)^{\mathbb{I}(h_{i_t,j}=a_j^{(k)})}\left(\frac{\beta_h + m'_{[-i_t],k,j}}{(|B|-1)(\alpha_h+\beta_h+n_k-1)}\right)^{\mathbb{I}(h_{i_t,j}\neq a_j^{(k)})}\end{aligned} \tag{5.11}$$

Note that during the sampling of c_{i_t}, the numerical values of c_{i_t} are arbitrary, as long as they index distinct equivalence classes.

Now we need to sample the ancestor template $a^{(k)}$, where k is the newly sampled ancestor index for c_{i_t}. When k is not equal to any other existing index $c_{i'_{t'}}$, a value for a_k needs to be chosen from $p(a|h_{i_t})$, the posterior distribution of A based on the prior $p(a)$ and the single dependent haplotype h_{i_t}. On the other hand, if k is an equivalence class populated by previous samples of $c_{i'_{t'}}$, we draw a new value of $a^{(k)}$ from $p(a|\{h_{i_t},\ :\ c_{i_t}=k\})$. If after a new sample of c_{i_t}, a template is no longer associated with any haplotype instance, we remove this template from the pool. The conditional distribution for this Gibbs step is therefore:

$$\begin{aligned} p(a^{(k)}|\mathbf{a}^{(-k)}, \mathbf{h}, \mathbf{c}) &= p(a^{(k)}|\{h_{i_t},\ :\ c_{i_t}=k\}) \\ &= \frac{p(\{h_{i_t},\ :\ c_{i_t}=k\}|a^{(k)})}{\sum_a p(\{h_{i_t},\ :\ c_{i_t}=k\}|a^{(k)}=a)} \\ &= \prod_j \frac{p(m_{k,j}|a_j^{(k)})}{\sum_{l\in B} p(m_{k,j}|a_j^{(k)}=l)}. \end{aligned} \tag{5.12}$$

We can sample $a_1^{(k)}, a_2^{(k)}, \ldots,$ sequentially:

$$p(a_j^{(k)} \,|\{h_{i_t,j} \,:\, c_{i_t} = k\}) =$$

$$\begin{cases} \frac{1}{Z} p(h_{i_t,j}|a_j^{(k)}) & \\ = \left(\frac{\alpha_h}{\alpha_h+\beta_h}\right)^{\mathbb{I}(h_{i_t,j}=a_j^{(k)})} \left(\frac{\beta_h}{(|B|-1)(\alpha_h+\beta_h)}\right)^{\mathbb{I}(h_{i_t,j}\neq a_j^{(k)})} & \text{if } k \text{ is not previously instantiated} \\ \\ \frac{1}{Z} p(\{h_{i_t,j} \,:\, c_{i_t} = k\}|a_j^{(k)}) & \\ = \frac{1}{Z} \frac{\Gamma(\alpha_h+m_{k,j})\Gamma(\beta_h+m'_{k,j})}{\Gamma(\alpha_h+\beta_h+n_k)\cdot(|B|-1)^{m'_{k,j}}} & \\ = \frac{\Gamma(\alpha_h+m_{k,j})\Gamma(\beta_h+m'_{k,j})/(|B|-1)^{m'_{k,j}}}{\sum_{l\in B}\Gamma(\alpha_h+m_{k,j}(l))\Gamma(\beta_h+m'_{k,j}(l))/(|B|-1)^{m'_{k,j}(l)}} & \text{if } k \text{ is previously instantiated,} \end{cases} \tag{5.13}$$

where $m_{k,j}$ (respectively, $m'_{k,j}$) is the number of allelic instances originating from ancestor k at locus j that are identical to (respectively, different from) the ancestor, when the ancestor has the pattern $a_j^{(k)}$; and $m_{k,j}(l)$ (respectively, $m'_{k,j}(l)$) is the value of $m_{k,j}$ (respectively, $m'_{k,j}$) when $a_j^{(k)} = l$. [4]

We now proceed to the second sampling stage, in which we sample the haplotypes h_{i_t}. We sample each $h_{i_t,j}$, for all j, i, t, sequentially according to the following conditional distribution:

$$\begin{aligned} & p(h_{i_t,j}|\mathbf{h}_{[-(i,j)]}, h_{i_{\bar{t}},j}, \mathbf{c}, \mathbf{a}, \mathbf{g}) \\ & \quad \propto \; p(g_i|h_{i_t,j}, h_{i_{\bar{t}},j}, \mathbf{u}_{[-(i,j)]}) p(h_{i_t,j}|a_j^{(k)}, \mathbf{m}_{[-(i_t,j)],k}) \\ & \quad = \; R_g \frac{\Gamma(\alpha_g+u)\Gamma(\beta_g+(u'+u''))}{\Gamma(\alpha_g+\beta_g+IJ)} [\mu_1]^{u'} [\mu_2]^{u''} \times R_h \frac{\Gamma(\alpha_h+m_{i_t,k,j})\Gamma(\beta_h+m'_{i_t,k,j})}{\Gamma(\alpha_h+\beta_h+n_k)\cdot(|B|-1)^{m'_{i_t,k,j}}}, \end{aligned} \tag{5.14}$$

where $[-(i_t,j)]$ denotes the set of indices excluding (i_t,j) and $m_{i_t,k,j} = m_{[-(i_t,j)],k,j} + \mathbb{I}(h_{i_t,j} = a_j^{(k)})$ (and similarly for the other sufficient statistics). Note that during each sampling step, we do not have to recompute the $\Gamma(\cdot)$, because the sufficient statistics are either not going to change (e.g.,

[4] Note that here the counts m_k (and m'_k) vary with different possible configurations of the ancestor $a^{(k)}$ under given $\mathbf{h}$, unlike previously in Eqs. (5.6)-(5.11), in which they vary with different possible configurations of h_{i_t} under given $a^{(k)}$.

when the newly sampled $h_{i_t,j}$ is the same as the old sample), or only going to change by one (e.g., when the newly sampled $h_{i_t,j}$ results in a change of the allele). In such cases the new gamma function can be easily updated from the old one.

5.3.2 A Metropolis-Hasting Sampling Algorithm

Note that for a long list of loci, a $p(a)$ that is uniform over all possible ancestral template patterns will render the probability of sampling a new ancestor infinitesimal, due to the small value of the smoothed marginal likelihood of any haplotype pattern h_{i_t}, as computed from Eq. (5.5). This could result in slow mixing.

An alternative sampling strategy is to use a partial Gibbs sampling strategy with the following Metropolis-Hasting updates, which could allow more complex $p(a)$ (e.g., non-factorizable and non-uniform) to be readily handled. To sample the equivalence class of h_{i_t} from the target distribution $\pi(c_{i_t}) = p(c_{i_t}|\mathbf{c}_{[-i_t]}, \mathbf{h}, \mathbf{a})$ described in Eq. 5.5, consider the following proposal distribution:

$$q(c^*_{i_t} = k|c_{[-i_t]}) = \begin{cases} \frac{n_{[-i_t],k}}{n-1+\tau} : \text{ if } k = c_{i'_{t'}} \text{ for some } i'_{t'} \neq i_t \\ \frac{\tau}{n-1+\tau} : \text{ if } k \neq c_{i'_{t'}} \text{ for all } i'_{t'} \neq i_t \end{cases} \tag{5.15}$$

Then we sample $a^{(c^*_{i_t})}$ from the prior $p(a)$. For the target distribution $p(c_{i_t} = k|\mathbf{c}_{[-i_t]}, \mathbf{h}, \mathbf{a})$, the proposal factor cancels when computing the acceptance probability ξ [5], leaving:

$$\xi(c^*_{i_t}, c_{i_t}) = \min\left[1, \frac{p(h_{i_t}|a^{c^*_{i_t}}, \mathbf{c}, \mathbf{h}_{[-i_t]})}{p(h_{i_t}|a^{c_{i_t}}, \mathbf{c}, \mathbf{h}_{[-i_t]})}\right]. \tag{5.17}$$

The choice of a more informative $p(a)$ is an open issue. Besides using a uniform prior, one can, for example, begin with a (small and hence inexpensive) finite mixture model using EM to roughly

[5] The cancellation of the proposal in ξ can be seen from the following steps:

$$\begin{aligned} \frac{q(c_{i_t}|\mathbf{c}_{[-i_t]})}{q(c^*_{i_t}|\mathbf{c}_{[-i_t]})}\frac{\pi(c^*_{i_t})}{\pi(c_{i_t})} &= \frac{q(c_{i_t}|\mathbf{c}_{[-i_t]})}{q(c^*_{i_t}|\mathbf{c}_{[-i_t]})}\frac{p(c^*_{i_t}|\mathbf{c}_{[-i_t]}, \mathbf{h}, \mathbf{a})}{p(c_{i_t}|\mathbf{c}_{[-i_t]}, \mathbf{h}, \mathbf{a})} \\ &= \frac{q(c_{i_t}|\mathbf{c}_{[-i_t]})}{q(c^*_{i_t}|\mathbf{c}_{[-i_t]})}\frac{p(c^*_{i_t}|\mathbf{c}_{[-i_t]})p(h_{i_t}|a^{(c^*_{i_t})}, \mathbf{c}, \mathbf{h}_{[-i_t]})}{p(c_{i_t}|\mathbf{c}_{[-i_t]})p(h_{i_t}|a^{(c_{i_t})}, \mathbf{c}, \mathbf{h}_{[-i_t]})} \\ &= \frac{p(h_{i_t}|a^{(c^*_{i_t})}, \mathbf{c}, \mathbf{h}_{[-i_t]})}{p(h_{i_t}|a^{(c_{i_t})}, \mathbf{c}, \mathbf{h}_{[-i_t]})}, \end{aligned} \tag{5.16}$$

ascertain major population haplotypes, and then construct a $p(a)$ by letting the EM-derived populational haplotypes take a large portion of the probability mass, and leaving some mass uniformly to all other possible ancestors. Using a non-rigorous heuristic, we sample according to Eq. (5.13). It can be shown that with this proposal, the acceptance rate defined by Eq. 5 still roughly holds [6]. In practice, we found that the above modification to the Gibbs sampling algorithm leads to substantial improvement in efficiency for long haplotype lists (even with a uniform base measure for A), whereas for short lists, the Gibbs sampler remains better due to the high (100%) acceptance rate.

5.3.3 A Sketch of MCMC Strategies for the Pedi-haplotyper model

Recall that the Pedi-haplotyper model is an extension of the basic Dirichlet process mixture haplotype model (i.e., the DP haplotyper model) that incorporates pedigree information for some individuals in a study population. The MCMC sampling strategy for the Pedi-haplotype model is similar to the one for the basic DP-haplotyper described above, except that we need to sample a few more variables newly introduced on top of the DP-haplotyper model, which requires collecting a few more sufficient statistics for updating the predictive distributions of these variables.

In addition to the sufficient statistics $\mathbf{m}$ (for the consistency between the ancestral and individual haplotypes (i.e., the number of cases of which the ancestral and individual haplotypes agree in a single sweep during sampling), and $\mathbf{u}$ (for the consistency between the individual haplotypes and genotype (i.e., the number of cases of which the genotype and its corresponding haplotype pair agree in a single sweep during sampling), needed in the DP-haplotyper model, we need to update the following sufficient statistics during each sampling step that sweeps all the random variables:

[6] To see this, note that now the proposal distribution is: $q(c_{i_t}|c_{[-i_t]})p(a^{(c_{i_t})}|\mathbf{a}^{[-c_{i_t}]}, \mathbf{h}, \mathbf{c})$, and the desired equilibrium distribution is $\pi(c_{i_t}, a^{(c_{i_t})}) = p(c_{i_t}, a^{(c_{i_t})} \mid \mathbf{c}_{[-i_t]}, \mathbf{h}, \mathbf{a}^{[-c_{i_t}]})$. The Markov transition probability is therefore:

$$
\begin{aligned}
& \frac{q(c_{i_t}|c_{[-i_t]})}{q(c^*_{i_t}|c_{[-i_t]})} \frac{p(a_{c_{i_t}}|\mathbf{a}^{[-c_{i_t}]}, \mathbf{h}, \mathbf{c})}{p(a_{c^*_{i_t}}|\mathbf{a}^{[-c^*_{i_t}]}, \mathbf{h}, \mathbf{c})} \frac{\pi(c^*_{i_t}, a^{[c^*_{i_t}]})}{\pi(c_{i_t}, a^{[c_{i_t}]})} \\
= & \frac{q(c_{i_t}|c_{[-i_t]})}{q(c^*_{i_t}|c_{[-i_t]})} \frac{p(a^{c_{i_t}}|\mathbf{a}^{[-c_{i_t}]}, \mathbf{h}, \mathbf{c})}{p(a^{c^*_{i_t}}|\mathbf{a}^{[-c^*_{i_t}]}, \mathbf{h}, \mathbf{c})} \frac{p(c^*_{i_t}|\mathbf{c}_{[-i_t]})p(h_{i_t}|a^{[c^*_{i_t}]}, \mathbf{c}, \mathbf{h}_{[-i_t]})}{p(c_{i_t}|\mathbf{c}_{[-i_t]})p(h_{i_t}|a^{[c_{i_t}]}, \mathbf{c}, \mathbf{h}_{[-i_t]})} \frac{p(a^{c^*_{i_t}}|\mathbf{a}^{[-c^*_{i_t}]}, \mathbf{h}, \mathbf{c}_{[-i_t]})}{p(a^{c_{i_t}}|\mathbf{a}^{[-c_{i_t}]}, \mathbf{h}, \mathbf{c}_{[-i_t]})} \\
\approx & \frac{p(h_{i_t}|a^{c^*_{i_t}}, \mathbf{c}, \mathbf{h}_{[-i_t]})}{p(h_{i_t}|a^{c_{i_t}}, \mathbf{c}, \mathbf{h}_{[-i_t]})}.
\end{aligned}
\tag{5.18}
$$

- $\mathbf{w}$: the sufficient statistics of the transition probability $\boldsymbol{\zeta}$,

$$w_{rr'} = \sum_t \sum_i \sum_j 1(s_{i_t,j} = r)1(s_{i_t,j+1} = r').$$

  If we prefer to model the recombination rates in males and females differently, then we compute $\mathbf{w}_t$ separately for $t = 0$ and $t = 1$.

- $\mathbf{v}$: the sufficient statistics of the single generation inheritance (i.e., non-mutation) rate ϵ,

$$v = \sum_t \sum_r \sum_i \sum_j 1(h_{i_t,j} = h_{\pi_r(i_t),j})1(s_{i_t,j} = r).$$

The ancestral template indicators associated with the founding subjects and the ancestor pool can be sampled as usual using the Gibbs and/or MH procedures used for the basic Dirichlet process mixture model for haplotypes. Now we derive the additional predictive distributions needed for collapsed Gibbs sampling for the Pedi-haplotyper model. For each predictive distribution of the hidden variables, we integrate out the model parameters given their (conjugate) priors (see §3.3 and §5.3.1 for definitions of most of the notations used here).

- To sample a founding haplotype:

$$\begin{aligned}
& p(h_{i_t,j}|\mathbf{h}_{[-(i,j)]}, h_{i_{\bar{t}},j}, \mathbf{s}, \mathbf{c}, \mathbf{a}, \mathbf{g}) \\
= \; & p(h_{i_t,j}|h_{i_{\bar{t}},j}, h_{\lambda(i),j}, s_{\lambda(i),j}, a_{c_{i_t},j}, g_i, \mathbf{v}_{[-(i,j)]}, \mathbf{u}_{[-(i,j)]}, \mathbf{m}_{[-(i,j)]}) \\
\propto \; & p(h_{i_t,j}, h_{\lambda(i),j}, g_i|h_{i_{\bar{t}},j}, s_{\lambda(i),j}, a_{c_{i_t},j}, \mathbf{v}_{[-(i,j)]}, \mathbf{u}_{[-(i,j)]}, \mathbf{m}_{[-(i,j)]}) \\
= \; & p(h_{\lambda(i),j}|h_{i_t,j}, h_{i_{\bar{t}},j}, s_{\lambda(i),j}, \mathbf{v}_{[-(i,j)]}) p(g_i|h_{i_t,j}, h_{i_{\bar{t}},j}, \mathbf{u}_{[-(i,j)]}) p(h_{i_t,j}|a_{c_{i_t},j}, \mathbf{m}_{[-(i,j)]}) \\
= \; & R_m \frac{\Gamma(\alpha_m + v(h_{i_t,j}))\Gamma(\beta_m + v'(h_{i_t,j}))}{\Gamma(\alpha_m + \beta_m + v(h_{i_t,j}) + v'(h_{i_t,j}))} \times \\
& R_g \frac{\Gamma(\alpha_g + u(h_{i_t,j}))\Gamma(\beta_g + u'(h_{i_t,j}) + u''(h_{i_t,j}))}{\Gamma(\alpha_m + \beta_m + IJ)} \mu_1^{u'} \mu_2^{u''} \times \\
& R_h \frac{\Gamma(\alpha_h + m(h_{i_t,j}))\Gamma(\beta_h + m'(h_{i_t,j}))}{\Gamma(\alpha_h + \beta_h + m(h_{i_t,j}) + m'(h_{i_t,j})) \cdot (|B| - 1)^{m'(h_{i_t,j})}},
\end{aligned} \tag{5.19}$$

  where $h_{\lambda(i),j}$ refers to the allele in the child of i that is inherited from i. For simplicity, we suppose only one child. For the case of multiple children, the first term of Eq. (5.19) becomes a product of such terms each corresponding to one child.

- To sample a non-founding haplotype:

$$
\begin{aligned}
& p(h_{i_t,j}|\mathbf{h}_{[-(i,j)]}, h_{i_{\bar{t}},j}, \mathbf{s}, \mathbf{c}, \mathbf{a}, \mathbf{g}) \\
= \;& p(h_{i_t,j}|\mathbf{h}_{[-(i,j)]}, h_{i_{\bar{t}},j}, h_{\lambda(i),j}, h_{\pi(i_t)_0,j}, h_{\pi_t(i_t),j}, s_{i_t,j}, s_{\lambda(i),j}, g_i, \mathbf{v}_{[-(i,j)]}, \mathbf{u}_{[-(i,j)]}) \\
\propto \;& p(h_{i_t,j}, h_{\lambda(i),j}, g_i|\mathbf{h}_{[-(i,j)]}, h_{i_{\bar{t}},j}, h_{\pi(i_t)_0,j}, h_{\pi_t(i_t),j}, s_{i_t,j}, s_{\lambda(i),j}, \mathbf{v}_{[-(i,j)]}, \mathbf{u}_{[-(i,j)]}) \\
= \;& p(h_{i_t,j}|h_{\pi(i_t)_0,j}, h_{\pi(i_t)_1,j}, s_{i_t,j}, \mathbf{v}_{[-(i,j)]}) p(h_{\lambda(i),j}|h_{i_t,j}, h_{i_{\bar{t}},j}, s_{\lambda(i),j}, \mathbf{v}_{[-(i,j)]}) \\
& p(g_i|h_{i_t,j}, h_{i_{\bar{t}},j}, \mathbf{u}_{[-(i,j)]}) \\
= \;& R_m \frac{\Gamma(\alpha_m + v(h_{i_t,j}))\Gamma(\beta_m + v'(h_{i_t,j}))}{\Gamma(\alpha_m + \beta_m + v(h_{i_t,j}) + v'(h_{i_t,j}))} \times \\
& R_g \frac{\Gamma(\alpha_g + u(h_{i_t,j}))\Gamma(\beta_g + u'(h_{i_t,j}) + u''(h_{i_t,j}))}{\Gamma(\alpha_m + \beta_m + IJ)} \mu_1^{u'} \mu_2^{u''}.
\end{aligned}
\tag{5.20}
$$

- To sample the segregation variable:

$$
\begin{aligned}
& p(s_{i_t,j}|\mathbf{h}, \mathbf{s}_{[-(i,j)]}, s_{i_{\bar{t}},j}, \mathbf{c}, \mathbf{a}, \mathbf{g}) \\
= \;& p(s_{i_t,j}|h_{i_t,j}, h_{\pi_0(i_t),j}, h_{\pi_1(i_t),j}, s_{i_t,j-1}, s_{i_t,j+1}, \mathbf{v}_{[-(i,j)]}, \mathbf{w}_{[-(i_t,j)]}) \\
\propto \;& p(h_{i_t,j}|h_{\pi_0(i_t),j}, h_{\pi_1(i_t),j}, s_{i_t,j}, \mathbf{v}_{[-(i,j)]}) p(s_{i_t,j-1}|s_{i_t,j}, \mathbf{w}_{[-(i_t,j)]}) \\
= \;& p(s_{i_t,j}|s_{i_t,j+1}, \mathbf{w}_{[-(i_t,j)]}) \\
= \;& R_m \frac{\Gamma(\alpha_m + v(s_{i_t,j}))\Gamma(\beta_m + v'(s_{i_t,j}))}{\Gamma(\alpha_m + \beta_m + v(s_{i_t,j}) + v'(h_{i_t,j}))} \times \\
& R_s \frac{\Gamma(\alpha_s + w_{00}(s_{i_t,j}) + w_{11}(s_{i_t,j}))\Gamma(\beta_s + + w_{01}(s_{i_t,j}) + w_{10}(s_{i_t,j}))}{\Gamma(\alpha_s + \beta_s + |\mathbf{w}|)},
\end{aligned}
\tag{5.21}
$$

  where $|\mathbf{w}| = \sum_{r,r'} w_{r,r'}$.

5.3.4 Summary

In this section we presented stochastic inference algorithms based on a pure Gibbs sampling scheme and a variant based on a Metropolis Hasting scheme for haplotype inference under a Dirichlet process mixture model—DP haplotyper. We also sketched Pedi-haplotyper, a Gibbs sampler for haplotype inference with pedigree information. We implemented the DP-haplotyper and validated it on

both simulated and real genotype data (see §3.4), and demonstrated superior performance compared to the state-of-the-art algorithm for haplotype inference. An implementation of the Pedi-haplotyper Gibbs sampler is deferred to future work.

If desired, we can also use these algorithms as subroutines to compute Bayesian estimates of model parameters of interest, such as the recombination rate ζ under the Pedi-haplotyper model. This can be done via a Monte Carlo EM algorithm, where in the E step we sample the hidden variables using the algorithms just presented, and in the M step we use sufficient statistics from the samples to estimate the parameter based on sample average. We omit further discussion on this subject.

5.4 Conclusion

In comparison to the GMF algorithms presented in the previous chapter, modulo time complexity (for reaching equilibrium) and space complexity (for storing the samples), Monte Carlo methods are arguably more general and easier to apply for statistical computations (especially Bayesian inference) in a wide range of probabilistic models. Under the graphical model formalism where the conditional independencies among variables are made explicit, implementing a Markov chain Monte Carlo algorithm such as a Gibbs sampler is particularly straightforward—the proposal distribution of each variable reduces to a conditional distribution under the Markov blanket of the variable, which is easily identifiable from the graph topology and can be automated. In this chapter, we presented MCMC algorithms for the large-scale Bayesian models we developed for motif detection and haplotype inference, taking advantage of the simplicity offered by our graphical model formalism.

In particular, for certain graphical models, such as the non-parametric Bayesian models defined via a Dirichlet process prior (as for haplotype inference), MCMC algorithms appear to be the only practical methodology for probabilistic inference. They naturally handle the issue of representing the densities of a potentially infinite dimensional mixture model via sequentially generated samples from such a distribution, whereas it seems that a variational approximation (still being developed

by several authors, including the author of this thesis) has to apply an *ad hoc* predetermined truncation scheme to represent the approximate density. This reduces the original distribution to a finite mixture model [Blei and Jordan, 2004], greatly diminishing the flexibility offered by the original non-parametric model. Our Gibbs sampling algorithm for the Dirichlet process mixture model for haplotypes is quite competent in performance, although a comparison to a variational inference algorithm under the same model would be interesting to reveal any performance/cost trade-off.

Such a comparison was done for the **LOGOS** model, which belongs to the family of parametric Bayesian models (and hence is of fixed dimensionality). Despite the simplicity of implementing the Gibbs sampler for **LOGOS**, and acceptable performance in small-scale test problems, we found that the GMF algorithm significantly outperforms the Gibbs sampler in more challenging large-scale problems given finite time (see §4.7). This suggests that GMF is a competent and efficient alternative to Monte Carlo methods for what we believe to be a wide range of large fixed-dimensional parametric models, especially when the performance/cost trade-off needs to be tilted toward lowering the computational cost without sacrificing significantly in performance.

Chapter 6

Conclusions

6.1 Conclusions from This Work

In this work, we focused on probabilistic graphical models and algorithms for analyzing two particular types of genomic data known as gene regulatory sequences and single nucleotide polymorphisms. We presented new algorithms to solve the related computational biology tasks of motif detection and haplotype inference.

In Chapter 2, we re-formulated the conventional unsupervised *de novo* motif detection problem in genomic analysis as a semi-unsupervised learning problem, and developed a modular Bayesian Markovian model called **LOGOS**, which can be trained on biologically identified motifs and generalized to novel motif patterns. This model captures various properties of motifs, including canonical structures of motif families, syntax of motif occurrences, and the distribution of nucleotides in background sequences. The graphical model formalism enables us to model these aspects with individual submodels in a divide-and-conquer fashion, and results in a joint model that can be efficiently solved using an approximate inference algorithm based on generalized mean field approximation.

Chapter 3 introduces a novel application of the non-parametric Bayesian approach to the haplotype inference problem. Our model extends a conventional finite mixture model to a potentially infinite mixture model via a Dirichlet process that induces a prior distribution over the centroids (i.e., the identities of populational ancestral haplotypes) and the cardinality (i.e., the number of distinct ancestral haplotypes) of the mixture model. Such an extension is particularly suitable for data

with a complex unknown distribution. It provides an alternative approach to the conventional model selection methods based on a finite model space, imposes an implicit parsimonious bias on the degree of diversity of haplotypes and allows a model to expand in a statistically consistent fashion to accommodate increasing data that may have new patterns. Our model also incorporates a likelihood factor that naturally handles missing values and statistical errors in the haplotype/genotype relationship.

Another major technical focus of this thesis is the development of efficient approximate algorithms for probabilistic inference in complex models that are intractable for exact algorithms. In Chapter 4, we developed a generalized mean field theory for approximate probabilistic inference in complex graphical models using a generic optimization procedure based on graph partitioning and message passing that provably converges to globally consistent marginals and a lower bound on the likelihood. This framework generalizes previous works on model-specific structured variational approximation yet specializes a previous study suggesting non-disjoint model decompositions, and appears to strike the right balance between approximation quality and complexity. This work aims to develop a turnkey algorithm for distributed approximate inference with bounded performance. The GMF algorithm has been successfully used as the main algorithm for inference and learning in the **LOGOS** model and exhibits superior performance compared to its MCMC counterpart. However, under a non-parametric Bayesian setting, as used for haplotyping, MCMC algorithms developed in Chapter 5 still appear to be the only practical approach.

6.2 Future Work

6.2.1 Modeling Gene Regulation Networks of Higher Eukaryotes in Light of Systems Biology and Comparative Genomics

It is widely believed that using diverse sources of related data and modeling them jointly is essential to gain deep insight into complex biology phenomena. As discussed briefly in §2.8, joint models comprising aspects of regulatory sequences, gene expression (e.g., microarray data), protein binding (CHiP data), and phylogenetic information, have begun to emerge and have shown promising

potential. We intend to explore extensions of our motif models along these directions under the **LOGOS** framework.

In particular, we are interested in studying the gene regulatory networks of higher eukaryotic organisms under a developmental context that involves temporal-spatial regulation of gene activities. Note that during the formation of a multicellular system such as an early embryo from a single cell such as a fertilized egg, each cell in the embryo has the same DNA content, but almost every single cell has a different function. This is somewhat analogous to a massive heterogeneous parallel system bootstrapped from the same program and subsequently differentiated by executing (temporally and spatially) context-specific subroutines of the common program. Deciphering the control mechanisms underlying such a system is crucial for understanding many biological processes typical of higher eukaryotes but nonexistent in bacteria or yeast, such as embryogenesis and differentiation, which are closely related to important biomedical problems such as birth defects and cancer development. Due to the high complexity of higher eukaryotic genomes and the technical difficulties of directly profiling gene expression patterns in such species, (e.g., conventional approaches such as cDNA microarrays used in uni-cellular organisms, which reflect the average effect of a homogeneous cell population, are not sufficiently informative), a mere extrapolation of extant techniques developed on the bacteria/yeast platform is not sufficient. Departing from the **LOGOS** model, we plan to develop more accurate and expressive statistical models that facilitate investigations of the gene regulation networks of higher eukaryotes in light of richer information from systems biology and comparative genomics. Specifically, the following extensions are of particular interest:

Richer motif models. To improve the sensitivity and specificity of motif prediction in higher eukaryotic genomic sequences, it is necessary to upgrade both the local submodel and global submodel of the current **LOGOS** model to encode richer regulatory grammar and capture higher-order dependencies among and within the regulatory signals. An immediate extension of the models presented in this thesis is to replace the 1st-order Markov models over sequence positions with more elaborated Bayesian networks to model richer dependencies. Another promising future direction is to

combine the generative framework we adopted in this thesis with discriminative models such as conditional random fields [Lafferty *et al.*, 2001], so that long range interactions of sequence elements and the influence of neighborhood statistical properties on motif locations can be comprehensively integrated in a semi-supervised fashion.

Joint models for temporal-spatial profiles of gene expression. Image profiles of *in situ* hybridization (e.g., see Fig. 2.2) and immuno-staining are standard tools for cell and developmental biologists to study the whole-body temporal-spatial patterns of gene expression in higher eukaryotes, and prove to be much more informative than microarray profiles of homogenized tissue samples. Correlating this representation of gene expression with *cis*-regulatory sequences is an intriguing open problem, which demands much effort in both computational image analysis and the development of appropriate probabilistic models that can interface the image models and the sequence models.

Joint models for comparative genomics. It can be highly informative to investigate an organism in the light of its evolutionary relationship to other organisms. Therefore, comparative studies of non-protein-coding genomic sequences in several related species can potentially help to improve motif detection. Along this direction, plausible evolutionary models of motif sequences, and general methods to address the problem of low-quality alignment of regulatory sequences (compared to that of gene sequences) during comparative genomic analysis (which critically depends on alignment quality) are still to come. We intend to develop a joint model that correctly models within-species and cross-species variations of motif sequences resulting from genomic stochasticity and from speciation, respectively, in order to infer the compositions and locations of these recurring elements from either aligned or unaligned genomic sequences of multiple species.

In summary, an integration of heterogeneous biological data via unified and consistent joint mathematical models is essential for analyzing biological systems at a much more comprehensive scale, and will greatly help the pursuit of a predictive understanding of how developmental gene

regulatory networks are encoded and evolved.

6.2.2 Genetic Inference and Application Based on Polymorphic Markers

SNPs comprise the largest class of individual differences in genomes, and have become a focus of research interest because of their value for investigating the genetic and evolutionary basis of multi-factorial diseases and complex traits. Such investigations require an integration of polymorphic molecular markers, such as the SNP markers we studied, with genetic linkage maps, complex phenotypic traits, pedigrees, etc., under a unified model. Continuing on our current work on phasing SNP haplotypes of an *iid* population, future directions include both theoretical explorations of the evolutionary mechanisms and dynamics of the populational diversity reflected in the haplotypes and their implications for trait diversification and inheritance; and practical upgrades of our current models into ones that can be used to infer SNP blocks concurrently with phase resolution, to infer haplotypes under the constraints of partial pedigrees (briefly sketched in §3.6), to infer map-locations of genetic traits associated with phenotypic patterns, etc. The graphical model framework used in this thesis makes it straightforward to pursue these future directions by constructing advanced models using the Dirichlet process mixture model developed in this thesis as a basic building block. For example, the following extensions are immediately on the horizon:

Bayesian treatment of the scaling parameter in DP. The scaling parameter τ in the Dirichlet process controls the prior tendency to instantiate new ancestral haplotypes in a population. Since DP can be described by a metaphor of non-Darwinian evolution process, τ may indeed reveal certain aspects of the dynamics of genetic drift and fixation during evolution and hence plays an interesting role in modeling populational diversity. In Bayesian non-parametrics, it is standard to introduce an easy-to-handle prior for τ [West *et al.*, 1994; Rasmussen, 2000], which makes it adaptable to populational diversity, and allows it to be estimated *a posteriori*.

Hierarchical DP for ethnic-group-specific populational diversity. The early split of an ancestral population following a populational bottle-neck (e.g., due to sudden migration or environmental

changes) may lead to ethnic-group-specific populational diversity, which features both ancient haplotypes (that have high variability) shared among different ethnic groups, and modern haplotypes (that are more strictly conserved) uniquely present in different ethnic groups. This structure is analogous to a hierarchical clustering setting in which different groups comprising multiple clusters may share clusters with common centroids (e.g., different new topics may share some common key words). The hierarchical Dirichlet process mixture model developed by Teh *et al.* [2004] provides a promising Bayesian approach to model such structure. We are pursuing an extension of our (flat) DP haplotyper model using this approach.

Linkage analysis. The degree of correlation between haplotypes of genetic markers (SNPs) and phenotypic traits (e.g., disease susceptibility, drug response, body features, etc.), formally known as linkage disequilibrium, reflects the frequency of genetic recombinations (hence the physical distance) between the marker(s) and the potential causal gene(s) of the phenotypic traits on the chromosome, a measure of great medical and clinical value. In principle, a joint model for linkage analysis and haplotype inference can be obtained by replacing (or extending) the simple genotype model discussed in this thesis with a more sophisticated phenotype model that comprises 1) a *recombination submodel*, describing the dependencies between the marker and the target gene, e.g., via a stochastic process capturing distance-dependent decay of the recombination rate, 2) a *penetration submodel*, describing the correspondence between the target gene and the phenotypic traits, and 3) a likelihood submodel, capturing the stochasticity in phenotypic measurements. In practice, for multi-factorial traits, the problem is complicated by the necessity of modeling complex dependencies between multiple causal genes and their net effects at the phenotype level, which is still an open-ended problem that calls for advances in modeling and probabilistic inference methodology.

In summary, a long-term goal we intend to pursue along this direction is to build clinical-grade phasing and mapping software that performs routine genetic diagnosis based on individual or familial SNP records. Generalizing SNPs to general markers, the model to be developed can also

be extended to general pedigree inference, which is applicable to forensic analysis based on genetic material, a problem also of great interest and practical value. We believe that with the models and inference algorithms developed in this thesis, technical foundations are in place for developing a full-scale joint model for statistical genetic inference.

6.2.3 Automated Inference in General Graphical Models

Large-scale probability models, such as the ones we developed in this thesis, have outgrown the ability of current (and probably future) exact inference algorithms to compute posteriors and learn parameters. For this reason, development of efficient and broadly applicable approximation algorithms is critical to further progress. The generalized mean field theory we developed potentially opens paths to the implementation of efficient and general-purpose variational inference engines that are easily scalable and adaptable to a wide range of complex probabilistic models using canonical computational procedures, which should require little or no work on model-specific derivations, and should be capable of answering arbitrary probabilistic queries. To further improve the approximation quality, we also expect that better tractable families associated with higher-order approximations or novel model decomposition schemes will need to be explored. Analysis of the relationships between the structure of the optimization space and the quality of the resulting bounds on approximation error also deserves further investigation.

To conclude, in order to pursue a predictive understanding of how developmental gene regulatory networks are encoded and evolved, and the genetic basis of multi-factorial diseases and complex traits, thorough understanding of the biological entities under investigation and high-throughput generation of experimental data must join forces with rigorous quantitative models based on solid mathematical foundations and algorithms for efficient computation. In particular, we expect that the exploration of formalisms for data fusion and for modularizing large-scale probabilistic models, and the development of more powerful inference and learning algorithms scalable to complex models,

will be essential to keep up with the rapid pace of biological research, and furthermore will contribute to applications in other science and engineering domains involving predictive understanding and reasoning under uncertainty.

Appendix A

More details on inference and learning for motif models

A.1 Multinomial Distributions and Dirichlet Priors

To model a categorical random variable Z, which can take J possible discrete values (e.g., all 4 possible nucleotides, A, C, G and T, in a DNA sequence), a standard distribution is the **multinomial distribution**: $p(Z = j|\theta) = \theta_j$, $|\theta| = \sum_{j=1}^{J} \theta_j = 1$, $\theta_j > 0, \forall j$, where j represents one of the J possible values. The (column) vector $\theta = [\theta_1, \ldots, \theta_J]^t$ is called the multinomial parameter vector [1]. For a set of M *i.i.d.* samples of Z, $\mathbf{z} = (z_1, \ldots, z_M)$ (e.g. a whole column of nucleotides in a multi-alignment $\mathbf{A}$), the sufficient statistics are the counts of each possible value: $h_j = \sum_{m=1}^{M} \delta(z_m, j)$, where $\delta(a, b) = 1$ if $a = b$ and 0 otherwise. Under a multinomial distribution, the likelihood of a single sample z_m is:

$$p(z_m|\theta) = \prod_{j=1}^{J} \left[\theta_j\right]^{\delta(z_m,j)}, \tag{A.1}$$

and the joint likelihood of the *i.i.d.* sample set $\mathbf{z}$ is:

$$p(\mathbf{z}|\theta) = \prod_{m=1}^{M} \prod_{j=1}^{J} \left[\theta_j\right]^{\delta(z_m,j)} = \prod_{j=1}^{J} \left[\theta_j\right]^{h_j}. \tag{A.2}$$

[1] Note that for simplicity, in this thesis we reuse the symbol θ (and also h and α in the sequel) to denote a single column vector, whose elements are singly subscripted (e.g. θ_j); whereas in the main text and the next section, these symbols each denote a two-dimensional array consisting of a sequence of column vectors, whose elements are consequently doubly subscripted (e.g., $\theta_{l,j}$).

To model uncertainty about the multinomial parameters, we can treat θ as a multivariate continuous random variable, and use a **Dirichlet density** to define a prior distribution $\text{Dir}(\alpha)$ for θ:

$$p(\theta|\alpha) = C(\alpha) \prod_{j=1}^{J} [\theta_j]^{\alpha_j - 1}, \tag{A.3}$$

where the hyperparameters $\alpha = [\alpha_1, \ldots, \alpha_J]^t$, $\alpha_j > 0, \forall j$ are called the Dirichlet parameters, and $C(\alpha)$ is the normalizing constant which can be computed analytically:

$$C(\alpha) = \frac{\Gamma(|\alpha|)}{\prod_{j=1}^{J} \Gamma(\alpha_j)}, \tag{A.4}$$

where $\Gamma(\cdot)$ is the *gamma* function.

Now we can calculate the joint probability $p(\theta, \mathbf{z}|\alpha)$:

$$\begin{aligned} p(\theta, \mathbf{z}|\alpha) &= p(\mathbf{z}|\theta)p(\theta|\alpha) \\ &= C(\alpha) \prod_{j=1}^{J} [\theta_j]^{\alpha_j + h_j - 1} \\ &= \frac{C(\alpha)}{C(\alpha + h)} \text{Dir}(\alpha + h). \end{aligned} \tag{A.5}$$

Integrating Eq. (A.5) over θ, we obtain the marginal likelihood:

$$\begin{aligned} p(\mathbf{z}|\alpha) &= \int p(\theta, \mathbf{z}|\alpha) d\theta \\ &= \frac{\Gamma(|\alpha|)}{\Gamma(|\alpha| + |h|)} \prod_{j=1}^{J} \frac{\Gamma(\alpha_j + h_j)}{\Gamma(\alpha_j)} \\ &= \frac{C(\alpha)}{C(\alpha + h)}. \end{aligned} \tag{A.6}$$

From Eq. (A.6) we can see that the quantity $\alpha_j - 1$ can be thought of as an imaginary count of the number of times that event $(Z = j)$ has already occurred. Furthermore, we have the posterior distribution $p(\theta|\mathbf{z}, \alpha) = p(\theta, \mathbf{z}|\alpha)/p(\mathbf{z}|\alpha) = \text{Dir}(\alpha + h)$, which is isomorphic to the prior distribution, and thus analytically integrable. This isomorphism between the prior and posterior is called *conjugacy* and priors of such nature are called *conjugate priors*.

A.2 Estimating Hyper-Parameters in the HMDM Model

We can compute the maximum likelihood estimate of the hyper-parameters $\Theta = \{\alpha, \upsilon, \Upsilon\}$ of the HMDM model from a training dataset of known motifs using an EM algorithm. This approach is often referred to as empirical Bayes parameter estimation.

Following Sjölander *et al.* [1996], for a given set of multi-alignment matrices $\{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(K)}\}$, where each $\mathbf{A}^{(k)}$ represents a multiple alignment of M_k biologically identified instances of motif k of length L_k, the likelihood of the count vector $h_l^{(k)}$ summarizing the column of aligned nucleotides at site l of motif k, under the Dirichlet prior α_i, is

$$p(h_l^{(k)}|\alpha_i) = \frac{\Gamma(|h_l^{(k)}|+1)\Gamma(|\alpha_i|)}{\Gamma(|h_l^{(k)}|+|\alpha_i|)} \prod_{j=1}^{4} \frac{\Gamma(h_{l,j}^{(k)}+\alpha_{i,j})}{\Gamma(h_{l,j}^{(k)}+1)\Gamma(\alpha_{i,j})}. \tag{A.7}$$

Note that this formula is slightly different from Eq. (A.6) because $h_l^{(k)}$ can result from $\frac{\Gamma(|h_l^{(k)}|+1)}{\prod_{j=1}^{4}\Gamma(h_{l,j}^{(k)}+1)}$ distinct permutations of the M_k nucleotides. Since no particular ordering of the motif instances in multi-alignment matrices is assumed for the training data, it is more appropriate to model the probability of the count matrices h resulting from $\mathbf{A}$ than that of $\mathbf{A}$ itself [Sjölander *et al.*, 1996].

Thus, the complete log likelihood of the count matrices $h^{(k)} = \{h_1^{(k)}, \ldots, h_{L_k}^{(k)}\}, \forall k$, and the latent HMDM state sequences $s^{(k)} = \{s_1^{(k)}, \ldots, s_{L_k}^{(k)}\}, \forall k$, can be obtained by replacing the $\mathbf{A}^{(k)}$'s in Eq. (2.17) with $h^{(k)}$'s, integrating over each $\theta^{(k)}$ (which results in a term like Eq. (A.7) for each count vector), and taking the logarithm of the resulting marginal:

$$\begin{aligned}
l_c(\{\alpha, \upsilon, \Upsilon\}) &= \log p(h^{(1)}, \ldots, h^{(K)}, s^{(k)}, \ldots, s^{(K)}|\{\alpha, \upsilon, \Upsilon\}) \\
&= \log\left\{\prod_{k=1}^{K}\left[p(s_1^{(1)}|\upsilon)\cdot\left[\prod_{l=1}^{L_k-1} p(s_{l+1}^{(k)}|s_l^{(k)}, \Upsilon)\right]\cdot\left[\prod_{l=1}^{L_k} p(h_l^{(k)}|s_l^{(k)}, \alpha)\right]\right]\right\} \\
&= \sum_{k=1}^{K}\sum_{i=1}^{I}\delta(s_1^{(k)}, i)\log\upsilon_i + \sum_{k=1}^{K}\sum_{l=1}^{L_k-1}\sum_{i,i'=1}^{I}\delta(s_l^{(k)}, i)\delta(s_{l+1}^{(k)}, i')\log\Upsilon_{i,i'} \\
&+ \sum_{k=1}^{K}\sum_{l=1}^{L_k}\sum_{i=1}^{I}\delta(s_l^{(k)}, i)\left(\log\frac{\Gamma(|h_l^{(k)}|+1)\Gamma(|\alpha_i|)}{\Gamma(|h_l^{(k)}|+|\alpha_i|)} + \sum_{j=1}^{4}\log\frac{\Gamma(h_{l,j}^{(k)}+\alpha_{i,j})}{\Gamma(h_{l,j}^{(k)}+1)\Gamma(\alpha_{i,j})}\right).
\end{aligned} \tag{A.8}$$

The EM algorithm is essentially a coordinate ascent procedure that maximizes the expected complete log likelihood $E_{Q(s)}[l_c(\{\alpha, \upsilon, \Upsilon\})]$ (also written as $\langle l_c(\Theta)\rangle_Q$ for simplicity) over the distribution $Q(s)$ and the parameters $\Theta = \{\alpha, \upsilon, \Upsilon\}$ [Neal and Hinton, 1998]. In the E step, we seek $Q(s) = \arg\max_Q \langle l_c(\Theta)\rangle_Q$, which turns out to be $Q(s) = p(s|h, \Theta) = \prod_k p(s^{(k)}|h^{(k)}, \Theta)$. Thus the E step is equivalent to computing $\langle l_c(\Theta)\rangle_{p(s|h,\Theta)}$, which reduces to replacing the sufficient statistics dependent on $s^{(k)}$ in Eq. (A.8) with their expectations with respect to $p(s^{(k)}|h^{(k)}, \Theta)$. In the M step, we compute $\Theta = \arg\max_\Theta \langle l_c(\Theta)\rangle_Q$. Specifically, we iterate between the following two steps until convergence:

E step:

- Compute the posterior probabilities $p(s_l^{(k)}|h^{(k)})$ of the hidden states, and the matrix of co-occurrence probabilities $p(s_l^{(k)}, s_{l+1}^{(k)}|h^{(k)})$ for each motif k, using the *forward-backward* algorithm in a hidden Markov model with initial and transition probabilities defined by $\{\upsilon, \Upsilon\}$ and emission probabilities defined by $p(h_l^{(k)}|S_l^{(k)} = i) = p(h_l^{(k)}|\alpha_i)$ (i.e., Eq. (A.7)).

M step:

- Baum-Welch update for the HMM parameters $\{\upsilon, \Upsilon\}$ based on expected sufficient statistics computed from all the $p(s_l^{(k)}|h^{(k)})$ and $p(s_l^{(k)}, s_{l+1}^{(k)}|h^{(k)})$:

$$\upsilon_i = \frac{\sum_{k,l} p(S_l^{(k)} = i|h^{(k)})}{\sum_k L_k} \tag{A.9}$$

$$\Upsilon_{i,j} = \frac{\sum_{k,l} p(S_l^{(k)} = i, S_{l+1}^{(k)} = j|h^{(k)})}{\sum_{k,l}\sum_j p(S_l^{(k)} = i, S_{l+1}^{(k)} = j|h^{(k)})} \tag{A.10}$$

- Gradient ascent (one step per M-step) for the Dirichlet parameters: (To force the Dirichlet parameters to be positive, we reparameterize the Dirichlet parameters as $\alpha_{i,j} = e^{w_{i,j}}, \forall i, j$, as described by Sjölander *et al.* [1996].)

$$w_{i,j} = w_{i,j} + \eta\frac{\partial\langle l_c(\Theta)\rangle}{\partial w_{i,j}} \tag{A.11}$$

where

$$\frac{\partial\langle l_c(\Theta)\rangle}{\partial w_{i,j}} = \frac{\partial\langle l_c(\Theta)\rangle}{\partial \alpha_{i,j}}\frac{\partial \alpha_{i,j}}{\partial w_{i,j}} =$$
$$\sum_{k=1}^{K}\sum_{l=1}^{L_k}\alpha_{i,j}p(S_l^{(k)}=i|h^{(k)})\Big(\Psi(|\alpha_i|)-\Psi(|h_l^{(k)}|+|\alpha_i|)+\Psi(h_{l,j}^{(k)}+\alpha_{i,j})-\Psi(\alpha_{i,j})\Big),$$

(recall that $\Psi(x) = \frac{\partial \log\Gamma(x)}{\partial x} = \frac{\dot{\Gamma}(x)}{\Gamma(x)}$ is the digamma function) and η is the learning rate, usually set to be a small constant.

A.3 Computing the Expected Sufficient Statistics in the Global HMM

We show how to compute the expected sufficient statistics $\bar{h}$ in a global HMM, in which the emission parameters are defined by the background distribution and the motif multinomial parameters (or their estimates).

Note that the overall counting matrix equals the summation of the counting matrices of all identified motif instances (each single instance forms a matrix with four rows, one per nucleotide; each column of such a matrix has only one nonzero element, whose row index corresponds to the observed nucleotide at the position of the column and the value of this element is equal to 1):

$$h = \sum_t h(\mathbf{y}_{t:t+L-1})\mathbb{I}(\mathbf{X}_{t:t+L-1} = (1,\ldots,L)),$$

where $\mathbb{I}(\cdot)$ is an indicator function matching a sequence of states to a given motif state sequence. Taking the expectation on both sides with respect to the joint distribution $q_s(\mathbf{x})$, we have:

$$\begin{aligned}\bar{h} &= E_{q_s(\mathbf{x})}[h] \\ &= \sum_{\mathbf{x}} q_s(\mathbf{x}) \sum_{t=1}^{T-L+1} h(\mathbf{y}_{t:t+L-1})\mathbb{I}(\mathbf{x}_{t:t+L-1} = (1,\ldots,L)).\end{aligned}$$

We have to sum over all possible configurations of $\mathbf{X}$. Under the GMF approximation, $q_s(\mathbf{x})$ is a reparameterized HMM $p(\mathbf{x}|\mathbf{y},\bar{\phi}(\theta),\theta_{bg})$ (Eq. 4.29), which leads to the following simplification:

$$\bar{h} = \sum_{t=1}^{T-L+1} h(\mathbf{y}_{t:t+L-1})p(\mathbf{X}_{t:t+L-1} = (1,\ldots,L)|\mathbf{y})$$

where $p(\mathbf{X}_{t:t+L-1} = (1, \ldots, L)|\mathbf{y}) =$
$$\frac{\prod_{l=1}^{L-1} p(X_{t+l} = l+1 | X_{t+l-1} = l)\alpha(X_t = 1)\beta(X_{t+L-1} = L)\prod_{l=1}^{L-1} p(y_{t+l}|X_{t+l} = l+1)}{p(\mathbf{y})},$$

where $\alpha(x_t) \triangleq p(y_1, \ldots, y_t, x_t)$ and $\beta(x_t) \triangleq p(y_{t+1}, \ldots, y_T|x_t)$ are the two standard intermediate probabilistic terms computed in the forward-backward algorithm for HMMs. With a little algebra and using the assumption that for the global HMM state transitions within a motif are deterministic, it is easy to show that

$$p(\mathbf{X}_{t:t+L-1} = (1, \ldots, L)|\mathbf{y}) = \frac{\alpha(X_t = 1)\beta(X_t = 1)}{p(\mathbf{y})} = p(X_t = 1|\mathbf{y}),$$

which means that the posterior probability of a subsequence of states being a motif state sequence is just the posterior probability of the first indicator in the sequence being the motif-start state, which is surprisingly simple. Now,

$$\bar{h} = \sum_{t=1}^{T-L+1} h(\mathbf{y}_{t:t+L-1}) p(X_t = 1|\mathbf{y}), \tag{A.12}$$

where $p(X_t = 1|\mathbf{y})$ can be computed using the forward-backward algorithm. The time complexity of this inference is linear in the length of the sequence, and quadratic in the number of motif states. Since all within-motif state transitions are deterministic, careful bookkeeping during implementation can reduce the complexity to quadratic in the number of motif types, that is, $O(K^2T)$. For multiple input sequences, the overall expected counting matrix $\bar{h}$ is just the sum of the expected counting matrices computed from each sequence using Eq. (A.12).

A.4 Bayesian Estimation of Multinomial Parameters in the HMDM Model

We now show how to compute the Bayesian estimate of $\phi(\theta)$, the natural parameter of the multinomial distribution, in an HMDM model given the expected sufficient statistics $\bar{h}$.

First, we compute the posterior probability of a hidden state sequence $\mathbf{s}$ given $\bar{h}$. Plugging $\bar{h}$

into Eq. (2.17) and integrating over θ, we have the marginal probability:

$$p(\bar{h}, \mathbf{s}|\alpha, \upsilon, \Upsilon) = p(s_1) \prod_{l=1}^{L-1} p(s_{l+1}|s_l) \prod_{l=1}^{L} p(\bar{h}_l|s_l), \tag{A.13}$$

which is a standard (local) HMM with emission probability:

$$p(\bar{h}_l|S_l = i) = \frac{\Gamma(|\alpha_i|)}{\Gamma(|\bar{h}_l| + |\alpha_i|)} \prod_{j=1}^{4} \frac{\Gamma(\bar{h}_{l,j} + \alpha_{i,j})}{\Gamma(\alpha_{i,j})}. \tag{A.14}$$

With this fully specified HMM, we can compute the posterior probabilities of the hidden states $p(s_l|\bar{h})$ and the matrix of co-occurrence probabilities $p(s_l, s_{l+1}|\bar{h})$ using the standard forward-backward algorithm for HMMs.

Then, the Bayesian estimate of $\phi(\theta) = \ln(\theta)$ (in which $\ln(\cdot)$ is a componentwise operation) is computed as follows:

$$\begin{aligned}
\bar{\phi}_{l,j} &= \int_{\theta_l} \sum_{s_l} \ln \theta_{l,j} p(\theta_l|s_l, \alpha, \bar{h}) p(s_l|\alpha, \bar{h}) d\theta_l \\
&= \sum_{s_l} p(s_l|\bar{h}) \int_{\theta_l} \ln \theta_{l,j} p(\theta_l|\alpha_l, \bar{h}_l) d\theta_l \\
&= \sum_{i=1}^{I} p(S_l = i|\bar{h}) \big(\Psi(\alpha_{i,j} + \bar{h}_{l,j}) - \Psi(|\alpha_i| + |\bar{h}_l|)\big).
\end{aligned} \tag{A.15}$$

Appendix B

Proofs

B.1 Theorem 2: GMF approximation

For clarity, we restate the GMF theorem here, with the evidence symbol and hidden variable subscripts omitted. Our subsequent proof starts from this simplified statement.

Theorem (GMF): *For a general undirected probability model $p(\mathbf{x})$ and a clustering $\mathcal{C} : \{\mathbf{X}_{C_i}\}_{i=1}^{I}$, if all the potential functions that cross cluster borders are cluster-factorizable, then the generalized mean field approximation to $p(\mathbf{x})$ with respect to clustering $\mathcal{C}$ is a product of cluster marginals $q^{GMF}(\mathbf{x}) = \prod_{C_i \in \mathcal{C}} q_i^{GMF}(\mathbf{x}_{C_i})$ satisfying the following generalized mean field equations:*

$$q_i^{GMF}(\mathbf{x}_{C_i}) \quad = \quad p(\mathbf{x}_{C_i}|\mathcal{F}_i), \quad \forall i. \tag{B.1}$$

To prove the GMF theorem we need to use the calculus of variations [Sagan, 1992] to solve the optimization defined by Eq. (4.21). For convenience, we distinguish two subsets of nodes in a cluster i, the interior nodes and the border nodes, i.e., letting $\mathbf{X}_{C_i}$ denote the nodes in cluster C_i, we have $\mathbf{X}_{C_i} = \{\mathbf{Y}_{C_i}, \mathbf{Z}_{C_i}\}$ where $\mathbf{Y}_{C_i} \cap \mathbf{X}_{\mathcal{B}_i} = \emptyset$ (i.e., the interior nodes) and $\mathbf{Z}_{C_i} \subset \mathbf{X}_{\mathcal{B}_i}$ (i.e., the border nodes).

Proof. From Eq. (4.21), to find the optimizer of:

$$\int d\mathbf{y}_{C_i} d\mathbf{z}_{C_i} \exp\big\{ - \sum_{C_i \in \mathcal{C}} E_i'(\mathbf{y}_{C_i}, \mathbf{z}_{C_i})\big\}\big(1 - \Delta\big),$$

where $\Delta \equiv E - \sum_{C_i \in \mathcal{C}} E'_i + A(\boldsymbol{\theta})$, subject to the constraints that each E'_i defines a valid marginal distribution $q_i(\mathbf{y}_{C_i}, \mathbf{z}_{C_i})$ over all hidden variables in cluster i, we solve the Euler equations for a variational extremum, defined over Lagrangians $f(E'_i, \mathbf{x}_{C_i}) = \int d\mathbf{x}_{[\cdot \backslash i]} \big[\exp\{-\sum_j E'_j\}(1 - \Delta) - \sum_j \lambda_j \exp\{-E'_j\}\big]$ (where $\mathbf{x}_{[\cdot \backslash i]}$ refers to all hidden variables excluding those from cluster i):

$$\frac{\partial f}{\partial E'_i} - \frac{d}{d\mathbf{x}_{C_i}}\Big(\frac{\partial f}{\partial \dot{E}'_i}\Big) = 0 \qquad \forall i. \tag{B.2}$$

Since f does not depend on $\dot{E}'_i$ ($= \frac{dE'_i}{d\mathbf{x}_{C_i}}$), we have:

$$\int d\mathbf{x}_{[\cdot \backslash i]} \prod_{j \neq i} \exp\{-E'_j\}(E - \sum_j E'_j) - \lambda_i = 0$$

$\Rightarrow$

$$\begin{aligned} E'_i &= \int d\mathbf{x}_{[\cdot \backslash i]} \prod_{j \neq i} \exp\{-E'_j\}(E - \sum_{j \neq i} E'_j) - \lambda_i \\ &= C - \sum_{D_\alpha \subseteq C_i} \theta_\alpha \phi_\alpha(\mathbf{y}_{D_\alpha}) - \sum_{D_\beta \in \mathcal{B}_i} \theta_\beta \big\langle \phi_\beta(\mathbf{z}_{C_i \cap D_\beta}, \{\mathbf{z}_{C_j \cap D_\beta} : j \in I_{\beta i}\}) \big\rangle_{q_{I_{\beta i}}}, \end{aligned}$$

where $q_j = \exp\{-E'_j(\mathbf{y}_{C_j}, \mathbf{z}_{C_j})\}$ is the local marginal of cluster j; $I_{\beta i}$ denotes index set of the set of clusters other than C_i that intersect with clique D_β (i.e., all the clusters neighboring cluster i that intersect with clique β); and $q_{I_{\beta i}} = \prod_{j \in I_{\beta i}} q_j$ is the marginal over cluster set $I_{\beta i}$.

When the potential functions at the cluster boundaries factorize (say, multiplicatively) with respect to the clustering, we have:

$$\begin{aligned} E'_i &= C - \sum_{D_\alpha \subseteq \mathcal{C}_i} \theta_\alpha \phi_\alpha(\mathbf{y}_{D_\alpha}) - \sum_{D_\beta \in \mathcal{B}_i} \theta_\beta F_\beta(\phi_{\beta_i}(\mathbf{z}_{C_i \cap D_\beta}), \{\langle \phi_{\beta_j}(\mathbf{Z}_{C_j \cap D_\beta}) \rangle_{q_j} \ : \ j \in I_{\beta i}\}). \\ &= C - \sum_{D_\alpha \subseteq \mathcal{C}_i} \theta_\alpha \phi_\alpha(\mathbf{y}_{D_\alpha}) - \sum_{D_\beta \in \mathcal{B}_i} \theta_\beta \phi_{\beta_i}(\mathbf{z}_{C_i \cap D_\beta}) \prod_{j \in I_{\beta i}} \langle \phi_{\beta_j}(\mathbf{Z}_{C_j \cap D_\beta}) \rangle_{q_j}. \end{aligned}$$

So,

$$\begin{aligned} q_i(\mathbf{y}_{C_i}, \mathbf{z}_{C_i}) &= \exp\{-E'_i\} \\ &= p(\mathbf{y}_{C_i}, \mathbf{z}_{C_i} | \{\langle \phi_{\beta_j}(\mathbf{Z}_{C_j \cap D_\beta}) \rangle_{q_j}\}_{j \in I_{\beta i}, D_\beta \in \mathcal{B}_i}) \\ &= p(\mathbf{x}_{C_i} | \mathcal{F}_i), \quad \forall i. \end{aligned} \tag{B.3}$$

The presence of evidence $\mathbf{x}_{C_i,E}$ merely changes Eq. (B.3) to $q(\mathbf{x}_{C_i}) \propto p(\mathbf{x}_{C_i}, \mathbf{x}_{C_i,E} | \mathcal{F}_i)$. After normalization, this leads to $q(\mathbf{x}_{C_i}) = p(\mathbf{x}_{C_i} | \mathbf{x}_{C_i,E}, \mathcal{F}_i)$. ■

B.2 Theorem 5: GMF bound on KL divergence

Proof.

According to the GMF theorem, the GMF approximation to $p(\mathbf{x})$ is

$$\begin{aligned} q(\mathbf{x}) &= \prod_i q(\mathbf{x}_{C_i}) \\ &= \frac{1}{Z_q}\exp\Big\{\sum_i\sum_{D_\alpha\subseteq C_i}\theta_\alpha\phi_\alpha(\mathbf{x}_{D_\alpha})+\sum_i\sum_{D_\beta\subseteq\mathcal{B}_i}\theta_\beta\phi'_\beta(\mathbf{x}_{D_\beta\cap C_i})\Big\} \\ &= \frac{1}{Z_q}\exp\Big\{\sum_\alpha\theta_\alpha\phi_\alpha(\mathbf{x}_{D_\alpha})-\sum_{D_\beta\subseteq\cup\mathcal{B}_i}\theta_\beta\phi_\beta(\mathbf{x}_{D_\beta})+\sum_{D_\beta\subseteq\cup\mathcal{B}_i}k_\beta\theta_\beta\phi'_\beta(\mathbf{x}_{D_\beta\cap C_i})\Big\} \\ &= \frac{1}{Z_q}\exp\Big\{\sum_\alpha\theta_\alpha\phi_\alpha(\mathbf{x}_{D_\alpha})+\sum_{D_\beta\subseteq\cup\mathcal{B}_i}\theta_\beta\big(k_\beta\phi'_\beta(\mathbf{x}_{D_\beta\cap C_i})-\phi_\beta(\mathbf{x}_{D_\beta})\big)\Big\}, \end{aligned} \tag{B.4}$$

where $k_\beta = |I_\beta|$ is the number of clusters intersecting with clique β (note that for simplicity, we omit the argument $q_{I_{\beta i}}$ in the peripheral marginal potentials). Thus, the KL divergence from q to p is:

$$\begin{aligned} \mathrm{KL}(q\|p) &= \int_{\mathbf{x}} q(\mathbf{x})\log\frac{q(\mathbf{x})}{p(\mathbf{x})}d\mathbf{x} \\ &= \sum_{D_\beta\subseteq\cup B_i}\theta_\beta\Big(k_\beta\big\langle\phi'_\beta(\mathbf{X}_{D_\beta\cap C_i})\big\rangle_q-\big\langle\phi_\beta(\mathbf{X}_{D_\beta})\big\rangle_q\Big)-\log\frac{Z_q}{Z_p} \\ &= \sum_{D_\beta\subseteq\cup B_i}\theta_\beta(k_\beta-1)\big\langle\phi_\beta(\mathbf{X}_{D_\beta})\big\rangle_q-\log Z_q+\log Z_p. \end{aligned} \tag{B.5}$$

Now, letting $\phi_{\beta,\max}=\max_{\mathbf{x}}\phi_\beta(\mathbf{x}_{D_\beta})$, and $\phi_{\beta,\min}=\min_{\mathbf{x}}\phi_\beta(\mathbf{x}_{D_\beta})$, we have $\phi_{\beta,\min}\le\langle\phi_\beta(\mathbf{X}_{D_\beta})\rangle_q\le\phi_{\beta,\max}$. Define $a_\phi=\min_{D_\beta\subseteq\cup\mathcal{B}_i}(k_\beta-1)\phi_{\beta,\min}$, and $b_\phi=\max_{D_\beta\subseteq\cup B_i}(k_\beta-1)\phi_{\beta,\max}$. Then (since all the θs are positive by definition),

$$a_\phi\sum_{D_\beta\subseteq\cup\mathcal{B}_i}\theta_\beta\le\sum_{D_\beta\subseteq\cup\mathcal{B}_i}\theta_\beta(k_\beta-1)\big\langle\phi_\beta(\mathbf{X}_{D_\beta})\big\rangle_q\le b_\phi\sum_{D_\beta\subseteq\cup\mathcal{B}_i}\theta_\beta. \tag{B.6}$$

To bound the log partition function, we find that

$$\begin{aligned} Z_q &= \sum_{\mathbf{x}} \exp\Big\{\sum_{\alpha} \theta_\alpha \phi_\alpha(\mathbf{x}_{D_\alpha})\Big\} \times \exp\Big\{\sum_{D_\beta \subseteq \cup \mathcal{B}_i} \theta_\beta \big(k_\beta \phi'_\beta(\mathbf{x}_{D_\beta \cap C_i}) - \phi_\beta(\mathbf{x}_{D_\beta})\big)\Big\} \\ &\leq \sum_{\mathbf{x}} \exp\Big\{\sum_{\alpha} \theta_\alpha \phi_\alpha(\mathbf{x}_{D_\alpha})\Big\} \times \exp\Big\{b_Z \sum_{D_\beta \subseteq \cup \mathcal{B}_i} \theta_\beta\Big\} \\ &= Z_p \exp\Big\{b_Z \sum_{D_\beta \subseteq \cup \mathcal{B}_i} \theta_\beta\Big\}, \end{aligned} \tag{B.7}$$

where

$$\begin{aligned} b_Z &= \max_\beta (k_\beta \phi_{\beta,\max} - \phi_{\beta,\min}) \\ &= \max_\beta \big((k_\beta - 1)\phi_{\beta,\max} + (\phi_{\beta,\max} - \phi_{\beta,\min})\big). \end{aligned} \tag{B.8}$$

Similarly,

$$Z_q \geq Z_p \exp\Big\{a_Z \sum_{D_\beta \subseteq \cup \mathcal{B}_i} \theta_\beta\Big\}, \tag{B.9}$$

where

$$a_Z = \min_\beta \big((k_\beta - 1)\phi_{\beta,\min} + (\phi_{\beta,\min} - \phi_{\beta,\max})\big). \tag{B.10}$$

Thus,

$$\log Z_p + a_Z \sum_{D_\beta \subseteq \cup \mathcal{B}_i} \theta_\beta \leq \log Z_q \leq \log Z_p + b_Z \sum_{D_\beta \subseteq \cup \mathcal{B}_i} \theta_\beta. \tag{B.11}$$

Putting these together, we have

$$aW \leq \mathrm{KL}(q\|p) \leq bW, \tag{B.12}$$

where $a = \max(0,\ a_\phi - b_Z)$ and $b = b_\phi - a_Z$.

In the special case where $k_\beta = k$, for all β (e.g., all potentials are pairwise), $a_\phi - b_Z = (k-1)\min_{\beta,\beta'}(\phi_{\beta,\min} - \phi_{\beta',\max}) + \min_\beta(\phi_{\beta,\min} - \phi_{\beta,\max}) \geq k \min_{\beta,\beta'}(\phi_{\beta,\min} - \phi_{\beta',\max})$, and $b = b_\phi - a_Z \leq k \max_{\beta,\beta'}(\phi_{\beta,\max} - \phi_{\beta',\min}) \equiv k\Delta_\phi$. Since KL divergence is always nonnegative, we have

$$0 \leq \mathrm{KL}(q\|p) \leq k\Delta_\phi W. \tag{B.13}$$

■

Bibliography

[Akey *et al.*, 2001] J. Akey, L. Jin, and M. Xiong. Haplotypes vs single marker linkage disequilibrium tests: what do we gain? *Eur J Hum Genet*, 9:291–300, 2001.

[Alberts *et al.*, 2002] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell, 4th Edition*. Taylor and Francis, 2002.

[Anderson and Novembre, 2003] E. C. Anderson and J. Novembre. Finding haplotype block boundaries by using the minimum-description-length principle. *Am J Hum Genet*, 73:336–354, 2003.

[Attias, 2000] H. Attias. A variational Bayesian framework for graphical models. In *Advances in Neural Information Processing Systems 12*, 2000.

[Bailey and Elkan, 1994] T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proc. of the 2nd International Conf. on Intelligent Systems for Molecular Biology*, 1994.

[Bailey and Elkan, 1995a] T. L. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using EM. *Machine Learning*, 21:51–80, 1995.

[Bailey and Elkan, 1995b] T. L. Bailey and C. Elkan. The value of prior knowledge in discovering motifs with MEME. In *Proc. of the 3rd International Conf. on Intelligent Systems for Molecular Biology*, 1995.

[Barash *et al.*, 2003] Y. Barash, G. Elidan, N. Friedman, and T. Kaplan. Modeling dependencies in protein-DNA binding sites. In *Proc. of the 7th International Conf. on Research in Computational Molecular Biology*, 2003.

[Beal *et al.*, 2001] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden Markov model. In *Advances in Neural Information Processing Systems 13*, 2001.

[Benos *et al.*, 2002] P. V. Benos, A. S. Lapedes, and G. D. Stormo. Is there a code for protein-DNA recognition? Probab(ilistical)ly? *Bioassays*, 24(5):466–475, 2002.

[Berman *et al.*, 2002] B. P. Berman, Y. Nibu, B. D. Pfeiffer, P. Tomancak, S. E. Celniker, M. Levine, G. M. Rubin, and M. B. Eisen. Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome. *Proc. Natl. Acad. Sci. USA*, 99(2):757–762, 2002.

[Bernardo and Smith, 1994] J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. John Wiley, New York, 1994.

[Bishop and Winn, 2003] C. M. Bishop and J. Winn. Structured variational distributions in VIBES. In *Proceedings Artificial Intelligence and Statistics*, 2003.

[Bishop *et al.*, 2003] C. M. Bishop, D. Spiegelhalter, and J. Winn. VIBES: A variational inference engine for Bayesian networks. In *Advances in Neural Information Processing Systems 15*, 2003.

[Blackwood and Kadonaga, 1998] E. M. Blackwood and J. T. Kadonaga. Going the distance: A current view of enhancer action. *Science*, 281(5373):60–63, 1998.

[Blake and Merz, 1998] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[Blanchette and Tompa, 2003] M. Blanchette and M. Tompa. FootPrinter: A program designed for phylogenetic footprinting. *Nucleic Acids Research*, 31 (13):3840–3842, 2003.

[Blanchette *et al.*, 2002] M. Blanchette, B. Schwikowski, and M. Tompa. Algorithms for phylogenetic footprinting. *J Comput Biol*, 9 (2):211–223, 2002.

[Blei and Jordan, 2004] D. Blei and M. I Jordan. Variational methods for the Dirichlet process. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[Brudno *et al.*, 2003] M. Brudno, G. B. Do, G. M. Cooper, M. F. Kim, E. Davydov, E. D. Green, A. Sidow, and S. Batzoglou. LAGAN and multi-LAGAN: Efficient tools for large-scale multiple alignment of genomic DNA. *Genome Res*, 13 (4):721–731, 2003.

[Burge and Karlin, 1997] C. Burge and S. Karlin. Prediction of complete gene structure in human genomic DNA. *J. Mol. Biol*, 268:78–94, 1997.

[Bussemaker *et al.*, 2000] H. Bussemaker, H. Li, and E. Siggia. Building a dictionary for genomes: Identification of presumptive regulatory sites by statistical analysis. *Proc. Natl. Acad. Sci. USA*, 97, 2000.

[Bussemaker *et al.*, 2001] H. J. Bussemaker, H. Li, and E. D. Siggia. Regulatory element detection using correlation with expression. *Nat Genet.*, 27(2):167–171, 2001.

[Cardon and Stormo, 1992] L. R. Cardon and G. Stormo. Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments. *J Mol Biol.*, 223 (1):159–70, 1992.

[Chakravarti, 2001] A. Chakravarti. Single nucleotide polymorphisms: . . .to a future of genetic medicine. *Nature*, 409:822–823, 2001.

[Chiang *et al.*, 2003] D. Y. Chiang, A. M. Moses, M. Kellis, E. S. Lander, and M. B. Eisen. Phylogenetically and spatially conserved word pairs associated with gene-expression changes in yeasts. *Genome Res*, 4 (7):R43, 2003.

[Clark *et al.*, 1998] A. Clark, K. M. Weiss, D. A. Nickerson, S. L. Taylor, A. Buchanan, J. Stengard, V. Salomaa, E. Vartiainen, M. Perola, E. Boerwinkle, and C. F. Sing. Haplotype structure and

population genetic inferences from nucleotide-sequence variation in human lipoprotein lipase. *American Journal of Human Genetics*, 63:595–612, 1998.

[Clark, 1990] A. Clark. Inferences of haplotypes from PCR-amplified samples of diploid populations. *Mol. Biol. Evol*, 7:111–122, 1990.

[Clark, 2003] A. Clark. Finding genes underlying risk of complex disease by linkage disequilibrium mapping. *Curr Opin Genet Dev*, 13(3):296–302, 2003.

[Cowell *et al.*, 1999] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, 1999.

[Crick, 1970] F. Crick. Central dogma of molecular biology. *Nature*, 227(258):561–3, 1970.

[Daly *et al.*, 2001] M. J. Daly, J. D. Rioux, S. F. Schaffner, T. J. Hudson, and E. S. Lander. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29(2):229–232, 2001.

[Davidson, 2001] E. H. Davidson. *Genomic Regulatory Systems*. Academic Press, 2001.

[Efron, 1996] B. Efron. Empirical Bayes methods for combining likelihoods (with discussion). *J. Amer. Statist. Assoc.*, 91:538–565, 1996.

[Eisen, 2003] M. Eisen. Structural properties of transcription factor-DNA interactions and the inference of sequence specificity. submitted, 2003.

[El-Hay and Friedman, 2001] T. El-Hay and N. Friedman. Incorporating expressive graphical models in variational approximations: Chain-graphs and hidden variables. In *Proceedings of the 17th Annual Conference on Uncertainty in AI*, 2001.

[Escobar and West, 2002] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 2002.

[Eskin *et al.*, 2003] E. Eskin, E. Halperin, and R.M. Karp. Efficient reconstruction of haplotype structure via perfect phylogeny. *Journal of Bioinformatics and Computational Biology*, 1:1–20, 2003.

[Excoffier and Slatkin, 1995] L. Excoffier and M. Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12(5):921–7, 1995.

[Falkner *et al.*, 1994] J. Falkner, F. Rendl, and H. Wolkowitz. A computational study of graph partitioning. *Mathematical Programming*, 66(2):211–239, 1994.

[Ferguson, 1973] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1:209–230, 1973.

[Fine *et al.*, 1998] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden Markov model: Analysis and applications. *Machine Learning*, 32:41–62, 1998.

[Frech *et al.*, 1993] K. Frech, G. Herrmann, and T. Werner. Computer-assisted prediction, classification, and delimitation of protein binding sites in nucleic acids. *Nucleic Acids Res.*, 21 (7):1655–1664, 1993.

[Frieze and Jerrum, 1995] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. In Egon Balas and Jens Clausen, editors, *Integer Programming and Combinatorial Optimization*, volume 920, pages 1–13. Springer, 1995.

[Frith *et al.*, 2001] M. C. Frith, U. Hansen, and Z. Weng. Detection of cis-element clusters in higher eukaryotic DNA. *Bioinformatics*, 17:878–889, 2001.

[Fujioka *et al.*, 1999] M. Fujioka, Y. Emi-Sarker, G. L. Yusibova, T. Goto, and J. B. Jaynes. Analysis of an even-skipped rescue transgene reveals both composite and discrete neuronal and early blastoderm enhancers, and multi-stripe positioning by gap gene repressor gradients. *Development*, 126(11):2527–38, 1999.

[Gabriel *et al.*, 2002] S. B. Gabriel, S. F. Schaffner, H. Nguyen, et al. The structure of haplotype blocks in the human genome. *Science*, 296:2225–2229, 2002.

[Gelman, 1998] A. Gelman. Inference and monitoring convergence. In W. E. Gilks, S. Richardson, and D. J. Spiegelhalter, editors, *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, Boca Raton, Florida, 1998.

[Ghahramani and Beal, 2001] Z. Ghahramani and M.J. Beal. Propagation algorithms for variational Bayesian learning. In *Advances in Neural Information Processing Systems 13*, 2001.

[Ghahramani and Jordan, 1997] Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–273, 1997.

[Gilbert, 2003] S. F. Gilbert. *Developmental Biology, Seventh Edition*. Sinauer Associates, 2003.

[Gilks *et al.*, 1996] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.

[Goemans and Williamson, 1995] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42:1115–1145, 1995.

[Goto *et al.*, 1989] T. Goto, P. Macdonald, and T. Maniatis. Early and late periodic patterns of even skipped expression are controlled by distinct regulatory elements that respond to different spatial cues. *Cell*, 57(3):413–22, 1989.

[Greenspan and Geiger, 2003] D. Greenspan and D. Geiger. Model-based inference of haplotype block variation. In *Proceedings of RECOMB 2003*, 2003.

[GuhaThakurta and Stormo, 2001] D. GuhaThakurta and G. D. Stormo. Identifying target sites for cooperatively binding factors. *Bioinform.*, 17:608–621, 2001.

[Gupta and Liu, 2003] M. Gupta and J. S. Liu. Discovery of conserved sequence patterns using a stochastic dictionary model. *J. Amer. Statist. Assoc.*, 98, 2003.

[Gusfield, 2002] D. Gusfield. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions. In *Proceedings of RECOMB 2002*, pages 166–175, 2002.

[Gusfield, 2004] D. Gusfield. An overview of combinatorial methods for haplotype inference. Technical Report, UC Davis, 2004.

[Haldimann *et al.*, 1996] A. Haldimann, M. K. Prahalad, S. L. Fisher, S. Kim, C. T. Walsh, and B. L. Wanner. Altered recognition mutants of the response regulator PhoB: A new genetic strategy for studying protein-protein interactions. *Proc. Natl. Acad. Sci. USA*, 93:14361–14366, 1996.

[Halfon *et al.*, 2002] M. S. Halfon, Y. Grad, G. M. Church, and A. M. Michelson. Computation-based discovery of related transcriptional regulatory modules and motifs using an experimentally validated combinatorial model. *Genome Research*, 12:1019–1028, 2002.

[Halperin and Eskin, 2002] E. Halperin and E. Eskin. Haplotype reconstruction from genotype data using imperfect phylogeny. Technical Report, Columbia University, 2002.

[Harding *et al.*, 1989] K. Harding, T. Hoey, R. Warrior, and M. Levine. Autoregulatory and gap gene response elements of the even-skipped promoter of Drosophila. *EMBO J.*, 8(4):1205–12, 1989.

[Helden *et al.*, 2000] J. Van Helden, A. Rios, and J. Collado-Vides. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Res.*, 28:1808–1818, 2000.

[Hertz and Stormo, 1996] G. Z. Hertz and G. D. Stormo. Escherichia coli promoter sequences: Analysis and prediction. *Meth. Enzymol.*, 273:30–42, 1996.

[Hertz and Stormo, 1999] G. Z. Hertz and G. D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinform.*, 15:563–577, 1999.

[Hodge *et al.*, 1999] S. E. Hodge, M. Boehnke, and M. A. Spence. Loss of information due to ambiguous haplotying of SNPs. *Nat Genet*, 21:360–361, 1999.

[Huang *et al.*, 2004] H. Huang, M. Kao, X. Zhou, J. S. Liu, and W. H. Wong. Determination of local statistical significance of patterns in Markov sequences with application to promoter element identification. *Journal of Computational Biology*, 11 (1), 2004.

[Hughes *et al.*, 2000] J. D. Hughes, P. W. Estep, S. Tavazoie, and G. M. Church. Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae. *J. Mol. Biol*, 296(5):1205–14, 2000.

[Hugot *et al.*, 2001] J. P. Hugot, M. Chamaillard, H. Zouali, S. Lesage, J. P. Cezard, J. Belaiche, S. Almer, C. Tysk, G. A. O'Morain, M. Gassull, V. Binder, Y. Finkel, A. Cortot, R. Modigliani, P. Laurent-Puig, C. Gower-Rousseau, J. Macry, J. F. Colombel, M. Sahbatou, and G. Thomas. Association of NOD2 leucine-rich repeat variants with susceptibility to Crohn's disease. *Nature*, 411 (6837):599–603, 2001.

[Ishwaran and James, 2001] H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 90:161–173, 2001.

[Jaakkola and Jordan, 2000] T. S. Jaakkola and M. I. Jordan. Bayesian logistic regression: A variational approach. *Statistics and Computing*, 10:25–37, 2000.

[Jordan *et al.*, 1999] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 105–161. Kluwer Academic Publishers, 1999.

[Jordan, 2004] M. I. Jordan. Graphical models. *Bayesian Statistics*, special issue:in press, 2004.

[Kappen and Wiegerinck, 2002] H. J. Kappen and J. Wiegerinck. A novel iteration scheme for the cluster variation method. In *Advances in Neural Information Processing Systems 14*, 2002.

[Karchin *et al.*, 2002] R. Karchin, K. Karplus, and D. Haussler. Classifying G-protein coupled receptors with support vector machines. *Bioinformatics*, 18(1):147–159, 2002.

[Karisch and Rendl, 1998] S. E. Karisch and F. Rendl. Semidefinite programming and graph equipartition. In P. M. Pardalos and H. Wolkowicz, editors, *Topics in Semidefinite and Interior-Point Methods*, volume 18, pages 77–95. AMS, 1998.

[Kechris *et al.*, 2004] K. J. Kechris, E. van Zwet, P. J. Bickel, and M. B. Eisen. Detecting dna regulatory motifs by incorperating positional trends in information content. *Genome Biology*, 5:R50, 2004.

[Keles *et al.*, 2003] S. Keles, M. J. van der Laan, S. Dudoit, B. Xing, and M. B. Eisen. Supervised detection of regulatory motifs in DNA sequences. *Statistical Applications in Genetics and Molecular Biology*, 2 (1), 2003.

[Keles *et al.*, 2004] S. Keles, M. J. van der Laan, and C. Vulpe. Regulatory motif finding by logic regression. *Bioinformatics*, page in press, 2004.

[Kellis *et al.*, 2003] M. Kellis, N. Paterson, M. Endrizzi, B. Birren, and E. S. Lander. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, 423:241–254, 2003.

[Kenneth and Clark, 2002] M. W. Kenneth and A. G. Clark. Linkage disequilibrium and the mapping of complex human traits. *TRENDS in Genetics*, 18(1):19–24, 2002.

[Kikuchi, 1951] R. Kikuchi. Theory of cooperative phenomena. *Phys. Rev.*, 81:988, 1951.

[Kimmel and Shamir, 2004] G. Kimmel and R. Shamir. Maximum likelihood resolution of multi-block genotypes. In *Proceedings of RECOMB 2004*, pages 847–56, 2004.

[Krogh *et al.*, 1994] A. Krogh, M. Brown, I.S. Mian, K. Sjolander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *J Mol Biol.*, 235:1501–1531, 1994.

[Kruglyak and Nickerson, 2001] L. Kruglyak and D. A. Nickerson. Variation is the spice of life. *Nature Genetics*, 27:234–236, 2001.

[Lafferty *et al.*, 2001] J. Lafferty, F. Pereira, and A. McCallum. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.

[Lange, 2002] K. Lange. *Mathematical and Statistical Methods for Genetic Analysis*. Springer, 2002.

[Lari and Young, 1990] K. Lari and S. J. Young. The estimation of stochastic context free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4, 1990.

[Lauritzen and Sheehan, 2002] S. L. Lauritzen and N. A. Sheehan. Graphical models for genetic analysis. TR R-02-2020, Aalborg University, 2002.

[Lawrence and Reilly, 1990] C. Lawrence and A. Reilly. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, 7:41–51, 1990.

[Lawrence *et al.*, 1993] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

[Lazzeroni, 2001] L. C. Lazzeroni. A chronology of fine-scale gene mapping by linkage disequilibrium. *Stat Methods Med Res*, 10:57–76, 2001.

[Leisink and Kappen, 2001] M. A. R. Leisink and H. J. Kappen. A tighter bound for graphical models. In *Advances in Neural Information Processing Systems 13*, 2001.

[Lewin, 2003] B. Lewin. *Genes VIII*. Prentice Hall, 2003.

[Li *et al.*, 2003] L. Li, E. I. Shakhnovich, and L. A. Mirny. Amino acids determining enzyme-substrate specificity in prokaryotic and eukaryotic protein kinases. *Proc. Natl. Acad. Sci. USA*, 100(8):4463–4468, 2003.

[Lin *et al.*, 2002] S. Lin, C. J. Cutler, M. E. Zwick, and A. Chakravarti. Haplotype inference in random population samples. *Am J Hum Genet*, 71:1129–1137, 2002.

[Liu *et al.*, 1995] J. S. Liu, A. F. Neuwald, and C. E. Lawrence. Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *J. Amer. Statist. Assoc*, 90:1156–1169, 1995.

[Liu *et al.*, 2001] X. Liu, D. L. Brutlag, and J. Liu. Bioprospector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. In *Proc. of Pac Symp Biocomput*, pages 127–138, 2001.

[Liu *et al.*, 2002] X. S. Liu, D. L. Brutlag, and J. S. Liu. An algorithm for finding protein-DNA binding sites with applications to chromatin immunoprecipitation microarray experiments. *Nat Biotechnol*, 20(8):835–9, 2002.

[Liu, 1994] J. S. Liu. The collapsed Gibbs sampler with applications to a gene regulation problem. *J. Amer. Statist. Assoc*, 89:958–966, 1994.

[Lockless and Ranganathan, 1999] S W. Lockless and R. Ranganathan. Evolutionarily conserved pathways of energetic connectivity in protein families. *Science*, 286(5438):295–299, 1999.

[Loots *et al.*, 2002] G. G. Loots, I. Ovcharenko, L. Pachter, I. Dubchak, and E. M. Rubin. rVista for comparative sequence-based discovery of functional transcription factor binding sites. *Genome Res*, 12 (5):832–839, 2002.

[Ludwig *et al.*, 2000] M. Z. Ludwig, C. Bergman, N. H. Patel, and M. E. Kreitman. Evidence for stabilizing selection in a eukaryotic enhancer element. *Nature*, 403(6769):564–7, 2000.

[Markstein *et al.*, 2002] M. Markstein, P. Markstein, V. Markstein, and M. S. Levine. Genome-wide analysis of clustered Dorsal binding sites identifies putative target genes in the Drosophila embryo. *Proc. Natl. Acad. Sci. USA*, 99(2):763–768, 2002.

[Mehldau and Myers, 1993] G. Mehldau and E. W. Myers. A system for pattern matching applications on biosequences. *Computer Applications in the BioSciences*, 9(3):299–314, 1993.

[Michelson, 2002] A. M. Michelson. Deciphering genetic regulatory codes: A challenge for functional genomics. *Proc. Natl. Acad. Sci. USA*, 99:546–548, 2002.

[Minka, 2001] T. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the 15th Annual Conference on Uncertainty in AI*, 2001.

[Moriyama and Kim, 2003] E. N. Moriyama and J. Kim. Protein family classification with discriminant function analysis. In *Proceedings of Stadler Genetics Symposium*, 2003.

[Murphy *et al.*, 1999] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the 15th Annual Conference on Uncertainty in AI*, 1999.

[Nazina and Papatsenko, 2004] A. G. Nazina and D. A Papatsenko. Statistical extraction of eukaryotic cis-regulatory modules using exhaustive assessment of local word frequency. (http://homepages.nyu.edu/ dap5/CV/word_frequency.pdf), 2004.

[Neal and Hinton, 1998] R. M. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.

[Neal, 2000] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *J. Computational and Graphical Statistics*, 9(2):249–256, 2000.

[Niu *et al.*, 2002] T. Niu, S. Qin, X. Xu, and J. Liu. Bayesian haplotype inference for multiple linked single nucleotide polymorphisms. *American Journal of Human Genetics*, 70:157–169, 2002.

[Page and Holmes, 1998] R. D. M. Page and E. C. Holmes. *Molecular Evolution: A Phylogenetic Approach*. Blackwell, Oxford, 1998.

[Papatsenko *et al.*, 2002] D. A. Papatsenko, V. J. Makeev, A. P. Lifanov, M. Regnier, A. G. Nazina, and C. Desplan. Extraction of functional binding sites from unique regulatory regions: The Drosophila early developmental enhancers. *Genome Research*, 12:470–481, 2002.

[Patil *et al.*, 2001] N. Patil, A. J. Berno, D. A. Hinds, et al. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294:1719–1723, 2001.

[Pearl, 1988] J. Pearl. *Probabilistic Reasoning in Intelligent System: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[Pennacchio and Rubin, 2001] L. A. Pennacchio and E. M. Rubin. Genomic strategies to identify mammalian regulatory sequences. *Nature Reviews Genetics*, 2(2):100–109, 2001.

[Pritchard, 2001] J. K. Pritchard. Are rare variants responsible for susceptibility to complex disease? *Am J Hum Genet*, 69:124–137, 2001.

[Ptashne and Gann, 1997] M. Ptashne and A. Gann. Transcriptional activation by recruitment. *Nature*, 386:569–577, 1997.

[Ptashne, 1988] M. Ptashne. How eukaryotic transcriptional activators work. *Nature*, 335:683–689, 1988.

[Puffenberger *et al.*, 1994] E. G. Puffenberger, E. R. Kauffman, S. Bolk, T. C. Matise, S. S. Washington, M. Angrist, J. Weissenbach, K. L. Garver, M. Mascari, and R. Ladda et al. Identity-by-descent and association mapping of a recessive gene for Hirschsprung disease on human chromosome 13q22. *Hum Mol Genet*, 3 (8):1217–25, 1994.

[Quandt *et al.*, 1995] K. Quandt, K. Frech, H. Karas, E. Wingender, and T. Werner. MatInd and MatInspector: New fast and versatile tools for detection of consensus matches in nucleotide sequence data. *Nucleic Acids Res*, 23 (23):4878–84, 1995.

[Rabiner and Juang, 1986] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–15, January 1986.

[Rajewsky *et al.*, 2002] N. Rajewsky, M. Vergassola, U. Gaul, and E. D. Siggia. Computational detection of genomic cis-regulatory modules, applied to body patterning in the early Drosophila embryo. *BMC Bioinformatics*, 3:30:1–13, 2002.

[Rasmussen, 2000] C. E. Rasmussen. The infinite Gaussian mixture model. In *Advances in Neural Information Processing Systems 12*, 2000.

[Ren *et al.*, 2000] B. Ren, F. Robert, J. Wyrick, O. Aparicio, E. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. Volkert, C. Wilson, S. Bell, and R. Young. Genome-wide location and function of DNA binding proteins. *Science*, 290 (5500):2306–2309, 2000.

[Rendl and Wolkowicz, 1995] F. Rendl and H. Wolkowicz. A projection technique for partitioning the nodes of a graph. *Annals of Operations Research*, 58:155–180, 1995.

[Rioux *et al.*, 2001] J. D. Rioux, M. J. Daly, M. S. Silverberg, K. Lindblad, H. Steinhart, Z. Cohen, T. Delmonte, K. Kocher, and K. Miller et al. Genetic variation in the 5q31 cytokine gene cluster confers susceptibility to Crohn disease. *Nat Genet*, 29 (2):223–8, 2001.

[Risch, 2000] N. J. Risch. Searching for genetic determinants in the new millennium. *Nature*, 405(6788):847–56, 2000.

[Rubin, 2001] G. M. Rubin. The draft sequences: Comparing species. *Nature*, 409:820–821, 2001.

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

[Sachidanandam *et al.*, 2001] R. Sachidanandam, D. Weissman, S. C. Schmidt, et al. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 291:1298–2302, 2001.

[Sackerson *et al.*, 1999] C. Sackerson, M. Fujioka, and T. Goto. The even-skipped locus is contained in a 16-kb chromatin domain. *Dev Biol.*, 211(1):39–52, 1999.

[Sagan, 1992] H. Sagan. *Introduction to the Calculus of Variations*. Dover Publications, 1992.

[Saul and Jordan, 1996] L. K. Saul and M. I. Jordan. Exploiting tractable substructures in intractable networks. In *Advances in Neural Information Processing Systems 8*, 1996.

[Schneider and Stephens, 1990] T. D. Schneider and R. M. Stephens. Sequence logos: A new way to display consensus sequences. *Nucl. Acids Res.*, 18:6097–6100, 1990.

[Schneider *et al.*, 1986] T. D. Schneider, G. D. Stormo, L. Gold, and A. Ehrenfeucht. Information content of binding sites on nucleotide sequences. *J Mol Biol.*, 188(3):415–31, 1986.

[Schwartz *et al.*, 2003] S. Schwartz, W. J. Kent, A. Smit, Z. Zhang, R. Baertsch, R. C. Hardison, D. Haussler, and W. Miller. Human-mouse alignments with BLASTZ. *Genome Res*, 13 (1):103–107, 2003.

[Segal *et al.*, 2001] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller. Rich probabilistic models for gene expression. In *Ninth International Conference on Intelligent Systems for Molecular Biology*, 2001.

[Segal *et al.*, 2003a] E. Segal, Y. Barash, I. Simon, N. Friedman, and D. Koller. From promoter sequence to expression: a probabilistic framework. In *Proceedings of RECOMB 2002*, pages 263–272, 2003.

[Segal *et al.*, 2003b] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):166–76, 2003.

[Shalon *et al.*, 1996] D. Shalon, S. J. Smith, and P. O. Brown. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Research*, 6(7):639–45, 1996.

[Sharan *et al.*, 2003] R. Sharan, I. Ovcharenko, A. Ben-Hur, and R. Karp. Creme: A framework for identifying cis-regulatory modules in human-mouse conserved segments. *Bioinformatics*, Suppl. 1:i283–i291, 2003.

[Sigrist *et al.*, 2002] C. Sigrist, L. Cerutti, N. Hulo, A. Gattiker, L. Falquet L, M. Pagni, A. Bairoch, and P. Bucher. PROSITE: A documented database using patterns and profiles as motif descriptors. *Brief Bioinform*, 3:265–274, 2002.

[Sinha and Tompa, 2000] S. Sinha and M. Tompa. A statistical method for finding transcription factor binding sites. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 344–354, 2000.

[Sjölander *et al.*, 1996] K. Sjölander, K. Karplus, M. Brown, R. Hughey, A. Krogh, I.S. Mian, and D. Haussler. Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology. *Computer Applications in the Biosciences*, 12, 1996.

[Small *et al.*, 1996] S. Small, A. Blair, and M. Levine. Regulation of two pair-rule stripes by a single enhancer in the Drosophila embryo. *Dev Biol.*, 175(2):314–24, 1996.

[Stanojevic *et al.*, 1991] D. Stanojevic, S. Small, and M. Levine. Regulation of a segmentation stripe by overlapping activators and repressors in the Drosophila embryo. *Science*, 254(5036):1385–7, 1991.

[Stephens and Donnelly, 2000] M. Stephens and P. Donnelly. Inference in molecular population genetics. *Journal of the Royal Statistical Society, Series B*, 62:605–655, 2000.

[Stephens and Donnelly, 2003] M. Stephens and P. Donnelly. A comparison of Bayesian methods for haplotype reconstruction from population genotype data. *American Journal of Human Genetics*, 73:1162–1169, 2003.

[Stephens *et al.*, 2001] M. Stephens, N. Smith, and P. Donnelly. A new statistical method for haplotype reconstruction from population data. *American Journal of Human Genetics*, 68:978–989, 2001.

[Stoneking, 2001] M. Stoneking. Single nucleotide polymorphisms: From the evolutionary past. . . *Nature*, 409:821–822, 2001.

[Stormo and Fields, 1998] G. D. Stormo and D. S. Fields. Specificity, free energy and information content in protein-DNA interactions. *Trends in Biochemical Sciences*, 23:109–113, 1998.

[Stormo, 2000] G. D. Stormo. DNA binding sites: representation and discovery. *Bioinformatics*, 16 (1):16–23, 2000.

[Stryer, 1995] L. Stryer. *Biochemistry (4th. edition)*. W. H. Freeman and Company, 1995.

[Sturm, 1999] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999. Special issue on Interior Point Methods (CD supplement with software).

[Swendsen and Wang, 1987] R. Swendsen and J-S Wang. Non-universal critical dynamics in Monte Carlo simulation. *Physical Review Letters*, 58:86–88, 1987.

[Tanner and Wong, 1987] M. A. Tanner and W. H. Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82:528–540, 1987.

[Tavare and Ewens, 1998] S. Tavare and W.J. Ewens. The Ewens sampling formula. *Encyclopedia of Statistical Sciences*, Update Volume 2.:230–234, 1998.

[Teh *et al.*, 2004] Y. Teh, M. I. Jordan, M. Beal, and D. Blei. Hierarchical Dirichlet processes. Technical Report 653, Department of Statistics, University of California, Berkeley, 2004.

[Thijs *et al.*, 2001] G. Thijs, M. Lescot, K. Marchal, S. Rombauts, B. De Moor, P. Rouz, and Y. Moreau. A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics*, 17 (12):1113–112, 2001.

[Thompson, 1981] E. A. Thompson. Pedigree analysis of Hodgkin's disease in a Newfoundland genealogy. *Annals of Human Genetics*, 45:279–292, 1981.

[Ureta-Vidal *et al.*, 2003] A. Ureta-Vidal, L. Ettwiller, and E. Birney. Comparative genomics: genome-wide analysis in metazoan eukaryotes. *Nature Reviews Genetics*, 4:251–262, 2003.

[van Helden *et al.*, 1998] J. van Helden, B. Andre, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.*, 281:827–42, 1998.

[Vandenberghe and Boyd, 1996] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[Venter *et al.*, 2001] C. Venter, M. D. Adams, E. W. Myers, et al. The sequence of the human genome. *Science*, 291:1304–51, 2001.

[Wainwright and Jordan, 2003] M. J. Wainwright and M. I. Jordan. Variational inference in graphical models: The view from the marginal polytope. *Invited paper; Allerton Conference on Communication, Control, and Computing*, Oct. 2003.

[Wasserman and Sandelin, 2004] W. W. Wasserman and A. Sandelin. Applied bioinformatics for the identification of regulatory elements. *Nature Reviews Genetics*, 5:276–287, 2004.

[Weiss and Clark, 2002] K. Weiss and A. Clark. Linkage disequilibrium and the mapping of complex trains. *Trends. in Genetics*, 18(1):19–24, 2002.

[West *et al.*, 1994] M. West, P. Muller, and M. D. Escobar. Hierarchical priors and mixture models, with application in regression and density estimation. *Aspects of Uncertainty: A Tribute to D V Lindley*, 1994.

[Wiegerinck, 2000] W. Wiegerinck. Variational approximations between mean field theory and the junction tree algorithm. In *Proceedings of the 16th Annual Conference on Uncertainty in AI*, 2000.

[Wingender *et al.*, 2000] E. Wingender, X. Chen, R. Hehl, H. Karas, I. Liebich, V. Matys, T. Meinhardt, M. Pruss, I. Reuter, and F. Schacherer. TRANSFAC: An integrated system for gene expression regulation. *Nucleic Acids Res.*, 28:316–319, 2000.

[Xing and Karp, 2004] E. P. Xing and R. M. Karp. MotifPrototyper: a profile Bayesian model for motif family. *Proc. Natl. Acad. Sci. USA*, 101(29):10523–28, 2004.

[Xing *et al.*, 2001] E. P. Xing, D. Wolf, I. Dubchak, S. Spengler, M. Zorn, C. Kulikowski, and I. Muchnik. Automatic discovery of sub-molecular sequence domains in multi-aligned sequences: A dynamic programming algorithm for multiple alignment segmentation. *Journal of Theoretical Biology*, 212(2):129–139, 2001.

[Xing *et al.*, 2003a] E. P. Xing, M. I. Jordan, R. M. Karp, and S. Russell. A hierarchical Bayesian Markovian model for motifs in biopolymer sequences. In *Advances in Neural Information Processing Systems 15*, 2003.

[Xing *et al.*, 2003b] E. P. Xing, M. I. Jordan, and S. Russell. A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of the 19th Annual Conference on Uncertainty in AI*, 2003.

[Xing *et al.*, 2004a] E. P. Xing, M. I. Jordan, and S. Russell. Graph partition strategies for generalized mean field inference. In *Proceedings of the 20th Annual Conference on Uncertainty in AI*, 2004.

[Xing *et al.*, 2004b] E. P. Xing, W. Wu, M. I. Jordan, and R. M. Karp. Logos: A modular Bayesian model for *de novo* motif detection. *Journal of Bioinformatics and Computational Biology*, 2(1):127–154, 2004.

[Xing *et al.*, 2004c] E.P. Xing, R. Sharan, and M.I Jordan. Bayesian haplotype inference via the Dirichlet process. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[Yedidia *et al.*, 2001a] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *Advances in Neural Information Processing Systems 13*, 2001.

[Yedidia *et al.*, 2001b] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *Distinguished Lecture track, the 17th International Joint Conference on AI*, 2001.

[Zhang *et al.*, 2002] K. Zhang, M. Deng, T. Chen, M. Waterman, and F. Sun. A dynamic programming algorithm for haplotype block partitioning. *Proc. Natl. Acad. Sci. USA*, 99(11):7335–39, 2002.