THE UNIVERSITY of EDINBURGH

A novel stochastic and entropy-based Expectation-Maximisation algorithm for transcription factor binding site motif discovery

Alastair Morris Kilpatrick

Doctor of Philosophy

Centre for Intelligent Systems and their Applications
School of Informatics
University of Edinburgh

2014

Abstract

The discovery of transcription factor binding site (TFBS) motifs remains an important and challenging problem in computational biology. This thesis presents MITSU, a novel algorithm for TFBS motif discovery which exploits stochastic methods as a means of both overcoming optimality limitations in current algorithms and as a framework for incorporating relevant prior knowledge in order to improve results.

The current state of the TFBS motif discovery field is surveyed, with a focus on probabilistic algorithms that typically take the promoter regions of coregulated genes as input. A case is made for an approach based on the stochastic Expectation-Maximisation (sEM) algorithm; its position amongst existing probabilistic algorithms for motif discovery is shown. The algorithm developed in this thesis is unique amongst existing motif discovery algorithms in that it combines the sEM algorithm with a derived data set which leads to an improved approximation to the likelihood function. This likelihood function is unconstrained with regard to the distribution of motif occurrences within the input dataset. MITSU also incorporates a novel heuristic to automatically determine TFBS motif width. This heuristic, known as MCOIN, is shown to outperform current methods for determining motif width. MITSU is implemented in Java and an executable is available for download.

MITSU is evaluated quantitatively using realistic synthetic data and several collections of previously characterised prokaryotic TFBS motifs. The evaluation demonstrates that MITSU improves on a deterministic EM-based motif discovery algorithm and an alternative sEM-based algorithm, in terms of previously established metrics. The ability of the sEM algorithm to escape stable fixed points of the EM algorithm, which trap deterministic motif discovery algorithms and the ability of MITSU to discover multiple motif occurrences within a single input sequence are also demonstrated.

MITSU is validated using previously characterised Alphaproteobacterial motifs, before being applied to motif discovery in uncharacterised Alphaproteobacterial data. A number of novel results from this analysis are presented and motivate two extensions of MITSU: a strategy for the discovery of multiple different motifs within a single dataset and a higher order Markov background model. The effects of incorporating these extensions within MITSU are evaluated quantitatively using previously characterised prokaryotic TFBS motifs and demonstrated using Alphaproteobacterial motifs. Finally, an information-theoretic measure of motif palindromicity is presented and its advantages over existing approaches for discovering palindromic motifs discussed.

Lay summary

One of the most important and challenging problems in computational biology is the discovery of transcription factor binding site (TFBS) motifs. These are short DNA sequences which are important in switching the genes of an organism on or off. This thesis presents MITSU, a new computational method for discovering TFBS motifs.

Many existing computational methods for TFBS motif discovery can perform poorly because of the way that they search for motifs. These methods are often based on a technique known as 'deterministic EM'. MITSU is based on a different technique known as 'stochastic EM', which should give improved results. Using stochastic EM should also allow biologists to add extra information about motifs, which could lead to better results. MITSU also has a number of additional features which allow it to improve over other motif discovery methods. For example, MITSU has a feature (known as MCOIN) which helps it to automatically work out the most likely length of a motif.

Tests on artificial data, and motifs which are already known in bacteria, show that the stochastic EM method used by MITSU improves on the deterministic EM method, in terms of the motifs which are discovered. Further tests show why this is the case and demonstrate some of the additional features of MITSU.

Finally, MITSU is used to discover new motifs in a particular type of bacteria known as Alphaproteobacteria. The results of these tests show a number of possible new motifs and also suggest some ways of extending MITSU. These extensions are then built into MITSU and then tested using the known bacterial motifs.

Acknowledgements

I would like to thank all those who have helped me in the course of this work, especially my supervisors Dr. Stuart Aitken and Dr. Bruce Ward for all their support over the past few years. I have learned a huge amount from them and without their help, guidance and patience, reaching this point would have taken much longer. Thanks also to Dr. Guido Sanguinetti for his useful suggestions as part of our annual reviews. I would also like to thank my examiners, Prof. Dirk Husmeier (University of Glasgow) and Dr. Ian Simpson (internal), for taking the time to read my thesis and providing feedback which has undoubtedly improved it.

Finally, thanks to my friends and family for their love and support throughout.

Declaration

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

(*Alastair Morris Kilpatrick*)

For my family.

Si je puis

Table of Contents

List of Figures

List of Tables

Notation and Definitions

N	number of input sequences
$\mathbf{X} = \{X_1, \ldots, X_N\}$	set of N input sequences (observed variables)
L_i	length of input sequence i
$X_{i,j}$	the width-W subsequence beginning at position j in sequence i
W	width of a motif
t	EM iteration number
$\mathcal{L} = \{A, C, G, T\}$	set of DNA nucleotides
θ_0	background model
$(\theta_{0,A}, \ldots, \theta_{0,T})$	parameters of the multinomial distribution describing the background model
θ_j	column j of the motif model
$(\theta_{j,A}, \ldots, \theta_{j,T})$	parameters of the multinomial distribution describing column j of the motif model
$\theta^{(t)}$	estimated parameters of θ at EM iteration t
$\mathbf{Z}$	set of indicator variables (latent variables)
$Z_{i,j}$	indicator variable for $X_{i,j}$ (1 if $X_{i,j}$ is a motif occurrence, 0 otherwise)
$U_{i,j}$	expected probability that $X_{i,j}$ is not part of a previously discovered motif occurrence
$V_{i,j}$	probabilistic erasing factor associated with $X_{i,j}$
Q_i	latent variable indicating whether or not X_i contains a motif occurrence (ZOOPS model)
$Z_{i,j}^{(t)}$, $Q_i^{(t)}$	expected values of $Z_{i,j}$, Q_i at EM iteration t
γ	prior probability of a sequence containing a motif occurrence (ZOOPS model)

(continues over)

$\phi = \{\theta, \gamma\}$	complete model parameters
$\gamma^{(t)}$, $\theta^{(t)}$	expected values of γ, θ at EM iteration t
γ', θ'	proposed values of γ, θ (for stochastic EM)
α_M	Metropolis ratio (for stochastic EM)
$\beta\ (= \beta_A, \ldots, \beta_T)$	pseudocount/Laplace estimator (equivalent to prior Dirichlet distribution)

As far as possible, indexing variables are used so that i indexes X sequences, j indexes positions within X_i, k indexes nucleotides (i.e. $k \in \{A, C, G, T\}$) and m indexes positions within a motif (i.e. $m \in \{1, \ldots, W\}$). Other uses of indexing variables should be clear from the context in which they are used.

Following convention, species names (e.g. *Escherichia coli*) are italicised and biological class names (e.g. Alphaproteobacteria) are capitalised throughout. Gene names (e.g. *ctrA*) are also italicised. Consensus and nucleotide sequences are printed in `typewriter` font.

The *information content* of a motif position is defined (following Schneider and Stephens [145]) as:

$$2 + \sum_{k \in \mathcal{L}} \theta_{j,k} \log_2 \theta_{j,k},$$

for a given position j. This assumes that all nucleotides occur equally often in the original DNA sequence. Where this is not the case, a generalisation to the *relative entropy* may be made; this is defined (following Stormo [160]) as:

$$\sum_{k \in \mathcal{L}} \theta_{j,k} \log_2 \frac{\theta_{j,k}}{\theta_{0,k}}.$$

When the parameter values of θ_0 are all equal, the information content and relative entropy are equivalent.

Publications

1. A. M. Kilpatrick, B. Ward & S. Aitken, **Stochastic EM-based TFBS motif discovery with MITSU**
   Bioinformatics, 30(12):i310-i318, 2014

2. A. M. Kilpatrick & S. Aitken, **Stochastic algorithms for motif discovery: a comparison of sampling strategies**[1]
   F1000Posters 4:705, 2013

3. A. M. Kilpatrick, B. Ward & S. Aitken, **MCOIN: A novel heuristic for determining transcription factor binding site motif width**[2]
   Algorithms for Molecular Biology 8:16, 2013

4. A. M. Kilpatrick, S. Aitken & B. Ward, **MCOIN: A novel dual heuristic for determining unknown TFBS motif width**
   F1000Posters 4:436, 2013

5. A. M. Kilpatrick, S. Aitken & B. Ward, **An information-theoretic measure of TFBS motif palindromicity**
   F1000Posters 3:33, 2012

[1] Also as an extended abstract at
`http://light.ece.ohio.edu/~reggen/2013/Alastair_Kilpatrick.pdf`.
[2] Designated 'Highly accessed' by BioMed Central.

Chapter 1

Introduction

This chapter motivates the transcription factor binding site (TFBS) motif discovery problem, defines TFBS motifs and explains why their discovery is important in computational biology. The continued interest in solving this problem is also noted (Section 1.1). This is followed by a description of the biological processes behind the problem, including the basics of gene regulation and protein-DNA interaction, focusing on bacterial gene regulation (Section 1.2). Section 1.3 presents a high-level description of the stochastic EM-based approach used to solve the TFBS motif discovery problem in this thesis, along with a hypothesis. Finally, Section 1.4 provides a summary of the thesis chapters.

1.1 The transcription factor binding site motif discovery problem

Recent advances in genome sequencing have led to a huge increase in the amount of genome data available for study. Of considerable interest to biologists are transcription factor binding site (TFBS) motifs. These are short DNA sequence patterns that have important roles in gene transcription and regulation. Discovery and further analysis of these sequences remains an important task in the wider challenge of understanding the mechanisms of gene expression (examples from the recent ENCODE project include [183, 157, 175]) and the understanding of gene regulatory networks [102]. The understanding of regulatory components and networks in pathogenic and industrial bacteria is essential for the application of systems and computational biology in medicine and biotechnology. Consequently, there is much continuing interest in developing algo-

rithms which can automatically discover TFBS motifs [9].

The experimental work in this study is focused on motif discovery in Alphaproteobacteria. This is a broad class of bacteria containing many species which have useful biotechnological and scientific properties. For example, it includes the *Caulobacter* species which is used to study cell development and is a vector for vaccine protein production, the *Magnetospirillum* species which has potential nanotechnology applications utilising nanomagnetic particles and the *Sinorhizobium* species which play a significant role in nitrogen fixation and plant survival in nutrient depleted regions. Discovery of TFBS motifs in these species would facilitate the use of these bacteria in industrial and biotechnology applications.

The information within a gene is expressed by the cellular processes of transcription and translation. Transcription is often initiated by the binding of one or more proteins known as transcription factors to the promoter region of a gene (that is, the sequence upstream of a gene's start codon). Transcription factors can regulate gene expression by either activating or repressing gene transcription. The identification of these sequences is complicated by the fact that they are often subject to natural DNA mutations, insertions and deletions. Genes with similar functions or that act in a common pathway are often regulated by a common transcriptional regulator. It is therefore expected that the upstream transcription factor binding sites for the expression of these genes should be reasonably similar (although subject to the above mutations) in terms of both pattern and width (that is, the number of nucleotides in the site); these conserved binding sites are called 'motifs'. A DNA motif is formally defined as being a DNA sequence pattern which has some biological significance. In the context of this project, these motifs are biologically significant in that they regulate transcription.

In many cases, motifs are short and reasonably well-conserved, recurring in the promoter sequences of more than one gene. However, two specialised forms are also well known [41]: palindromic motifs (also known as 'inverted repeats') and gapped motifs (also known as 'spaced dyad' motifs). Palindromic motifs are sequences whose inverse complement is the same as the original sequence (for example, `GAGATCTC`). Experimental work has concluded that DNA sequence motifs are often palindromic, or quasi-palindromic in nature [11]. It must be noted, however, that this does not mean that all sequence motifs are (quasi-)palindromic. Gapped motifs consist of two smaller well-conserved segments, separated by a gap (or 'spacer') of non-conserved bases. The gap is usually of a fixed width, but can be variable [41, 63]. This often (but not always) means that the transcription factor is a dimer which binds to the DNA at two

separate contact points over the major groove of the DNA, both of which are often the same, or palindromes of each other. Studies have suggested that gapped motifs (often with palindromic patterns) are common, especially in prokaryotes [174, 63].

The motif discovery problem is the task of discovering overrepresented sequences given a number of DNA promoter regions (usually the upstream sequences of a number of functionally related genes). These overrepresented sequences are therefore good candidates for being transcription factor binding sites. Many algorithms have been developed in order to discover motifs; some of these are reviewed in Section 2.2.

1.2 Transcription factor binding site motifs and gene regulation

This section introduces the biology behind the computational problem and provides more detail on bacterial gene regulation. A number of different prokaryotic regulatory systems are described. Finally, traditional and bioinformatics-based approaches to identifying and characterising regulatory motifs are discussed.

1.2.1 Prokaryotic regulatory systems

Regulatory elements in bacteria include sigma factors, small RNAs (sRNAs) and transcriptional regulators. Sigma factors are proteins that enable specific binding of RNA polymerase to gene promoters. Some bacteria, such as *Bacillus subtilis*, are known to use sigma factors as a major control strategy, for example, in the regulation of sporulation. In contrast, other bacteria such as *Escherichia coli* are noted for the number of transcriptional regulators that alter transcriptional activity; the majority of these can be split into two groups: one-component regulators and two-component regulators.

Two-component systems are widely occurring regulatory systems in prokaryotes [159]. These systems consist of a sensor protein and a regulatory (transcriptional activator) protein. The sensor protein senses the level of a metabolite either directly or indirectly. As the presence of the metabolite is detected, the sensor protein becomes autophosphorylated. The phosphorylated form of the sensor protein acts on the transcriptional activator protein [67, 176]. The transcriptional activator protein has at least two domains: a protein-binding domain (in order to interact with the sensor protein) and a DNA-binding domain (in order to interact with the promoter region of

the DNA). When the (phosphorylated) sensor protein and the transcriptional activator protein interact, the transcriptional activator protein also becomes phosphorylated. The phosphorylated transcriptional activator protein binds to the DNA, usually in the promoter region of the regulated gene. For positive regulation (also known as induction), the binding of the transcriptional activator enhances the interaction of the RNA polymerase-sigma factor complex, resulting in an increased rate of transcription. Biochemically, the sensor protein is a histidine kinase and the transcriptional activator is an aspartate kinase (or, aspartokinase). In some cases, the sensor protein is not a single protein but a complex of two proteins, both of which are required to form a functional sensor; these are known as split histidine kinases [4, 134]. The NtrYX regulator discussed in Section 3.5.2 is an example of a two-component system in Alphaproteobacteria.

One-component regulators are proteins in which the sensor domain and the regulator domain are contained within a single protein. These are present in many Gram-negative bacteria, including the Alphaproteobacteria. The widely occurring CRP/FNR family of regulators are known to be one-component regulators. The FnrL regulator discussed in Section 3.5.2 is a member of the FNR family and an example of a one-component system in Alphaproteobacteria.

There are cases where regulators may act on other proteins rather than on DNA [31]. Often these are systems known as phosphorelays, or cascades, where the first protein transfers a phosphate to a second protein whose activity is then altered so as to either phosphorylate another protein or to act as a DNA-binding regulator. One important Alphaproteobacterial example of a phosphorelay is the CckA-ChpT-CtrA regulatory system in *Caulobacter crescentus*. In this system, the histidine kinase CckA phosphorylates the ChpT histidine phosphotransferase. ChpT can phosphorylate either of two response regulators, CpdR or CtrA. CpdR normally inhibits CtrA; however, it is inactive when it becomes phosphorylated. Uninhibited by CpdR, CtrA is an active transcriptional regulator that controls flagellar motility and is known to control, either directly or indirectly, at least 25% of the cell-cycle regulated genes in *C. crescentus* [97]. The CtrA motif is discussed further in Section 3.5.2.

1.2.2 Operons

In prokaryotes, protein-coding genes which have a similar metabolic function are often grouped closely together in the DNA. This arrangement makes it possible to produce

a single continuous strand of 'polycistronic' mRNA which codes for several proteins, all with the same metabolic goal. The transcription of this group of genes can be controlled by a single promoter. Such a group of genes is called an operon. Well-known examples of operons in *E. coli* include the *lac* operon (which metabolises lactose) and the *trp* operon (which synthesises the amino acid tryptophan) [57, 108]. The operon arrangement is particular to prokaryotes and is important in the context of motif discovery, in that the transcription factor binding site need not be directly upstream of the gene of interest. For instance, the *E. coli lac* operon is known to consist of three genes: in order, *lacZ*, *lacY* and *lacA*. If *lacA* is the gene of interest, it would normally be assumed that the binding site controlling its expression would be directly upstream of its start codon. However, because *lacA* is part of an operon, the binding site controlling its expression is directly upstream of the first gene in the operon, namely *lacZ*. Extracting the upstream of *lacA* is therefore of little use in discovering the binding site sequence; depending on the length, the upstream sequence of *lacA* will consist of the coding sequences for *lacZ* and *lacY*. The importance of determining operon structure as part of the process for discovering novel motifs will be made clear when constructing datasets in Section 3.5.2.

1.2.3 Identifying and characterising motifs

Traditionally, regulators and their consensus sequences or motifs have been identified through biological approaches. Many regulators have been identified by the isolation of mutants that either cause a noticeable change in the regulation of the levels of other proteins or cause major defects in a number of biological systems because they regulate many different genes. For example, the CtrA regulator discussed above is a global transcriptional regulator in *C. crescentus*. It was identified through a temperature-sensitive mutant strain in which flagella production was affected [138]. CtrA is now known to regulate genes in many different systems [97]. While regulators are most often identified by mutant studies, target genes are usually identified by proteomics or microarray studies.

Once the regulatory protein has been identified, the protein structure is often analysed using bioinformatics methods. For example, analysis of the FNR regulator confirmed it to have a structure consistent with one-component regulators: the N-terminus of the protein was shown to have a sensor domain, while the C-terminus had a DNA-binding domain. The DNA-binding domain of the FNR protein forms a winged helix-

turn-helix motif, which produces a characteristic gapped TFBS motif. It may therefore be possible to draw some conclusions about the DNA-binding domain of a protein based on the binding site motif.

The fact that regulatory genes need not be located close to their target genes (and are often located far away) makes it difficult to identify *both* regulators and consensus sequences by bioinformatics methods alone. The Alphaproteobacterial BioR regulator was identified by Rodionov and Gelfand in this way [142]; however, this is unusual. The experimental work using uncharacterised data in this thesis will therefore focus on discovering motifs for regulators where the target genes are already known.

Finally, the fact that not all regulators are found in all bacteria can make the computational discovery of uncharacterised motifs more difficult. Some regulators, such as FNR and Fur, are found across a wide range of bacterial groups. However, others are relatively restricted. For instance, the CtrA regulator is found widely in the Alphaproteobacteria but does not occur in Gammaproteobacteria (for example, *E. coli*). This is particularly relevant for this project, as the Alphaproteobacteria are known to be very diverse in their metabolic abilities and therefore the regulators which control these abilities (the diversity of the Alphaproteobacteria is further discussed in Section 3.5). The BioR regulator is known only to occur in the Rhizobiales and Rhodobacterales orders within the Alphaproteobacteria class. Similarly, photosynthetic bacteria are known to have specific regulators for these functions.

1.3 A stochastic Expectation-Maximisation approach to motif discovery

The majority of TFBS discovery algorithms are probabilistic algorithms, which search the input data (usually a collection of promoter regions of coregulated genes) for sequences which are statistically overrepresented. Deterministic algorithms make up a large proportion of commonly used algorithms for motif discovery. The deterministic Expectation-Maximisation (EM) algorithm is one of the first probabilistic algorithms to be applied to motif discovery [99] and is the basis for a number of others, including the benchmark motif discovery algorithm MEME [10]. However, the EM algorithm has several well-known limitations. For example, the EM algorithm is highly sensitive to its starting parameters. Due to this sensitivity and the use of a local search strategy, the EM algorithm cannot be guaranteed to converge to the global maximum of the like-

lihood function, instead converging to an insignificant local maximum or saddle point of the likelihood function. In general, the steps of the EM algorithm can become either analytically or computationally intractable in many practical situations.

This thesis uses an approach based on the stochastic EM (sEM) algorithm. The sEM algorithm is motivated by the limitations of the deterministic EM algorithm, particularly the issues of intractability. Celeux, *et al.* [37] note that in the wider field of probabilistic modelling, the sEM algorithm is generally more successful than the EM algorithm due to stochastic perturbations, which allow the sEM algorithm to escape stable fixed points of the EM algorithm such as insignificant local maxima of the likelihood function.

To date, only two algorithms have applied stochastic variants of EM to motif discovery (the SEAM [23] and MCEMDA [24] algorithms) and the power of sEM in a motif discovery context has not been fully explored. Most notably, these algorithms are limited to the 'one occurrence per sequence' (OOPS) model, which places a constraint on the distribution of motif occurrences within the input dataset. Further, algorithms based on stochastic variants of EM have so far not implemented features commonly found in other motif discovery algorithms, including the ability to automatically determine the most likely motif width from a range of plausible values and the ability to discovery multiple different motifs within the same dataset.

This thesis explores the power of stochastic EM in a TFBS motif discovery setting. A novel algorithm for motif discovery known as MITSU is developed [92]; this algorithm combines sEM with a derived data set which leads to an improved approximation of the likelihood function. Significantly, this likelihood function is unconstrained with regard to the number of motif occurrences in each input sequence. MITSU also implements features commonly found in other motif discovery algorithms; for example, an information-based heuristic known as MCOIN [91] is used to automatically determine the most likely motif width and a 'probabilistic erasing' method is implemented in order to discover multiple motifs within the same dataset. This thesis also explores the use of the sEM framework to incorporate relevant prior knowledge in order to improve results. MITSU is implemented in Java; a Java executable is available to download[1] and is supported on Linux and OS X.

[1] Download available at `http://www.sourceforge.net/p/mitsu-motif/`.

The algorithm developed in this thesis is used to evaluate the following hypotheses:

1. Transcription factor binding site motif discovery using stochastic Expectation-Maximisation improves on existing (deterministic) Expectation-Maximisation-based approaches, in terms of previously established metrics.

2. The incorporation of relevant prior biological knowledge through the stochastic EM framework also improves motif discovery in terms of previously established metrics.

1.4 Thesis outline

The following chapters in this thesis are summarised below.

Chapter 2 provides some relevant preliminaries and presents a critical review of probabilistic algorithms for TFBS motif discovery. This is followed by a discussion which makes the case for the stochastic EM approach adopted in this thesis.

Chapter 3 briefly describes the realistic synthetic datasets and characterised prokaryotic datasets constructed and used in the evaluation of the algorithms developed in this thesis. An analysis of previously characterised *E. coli* motifs is provided. The intergenic sequences in *E. coli* are discussed and used to motivate the incorporation of a higher order Markov background model presented in Chapter 6. Finally, a description of the previously uncharacterised Alphaproteobacterial data used in the study is also presented.

Chapter 4 begins with a derivation of the expressions central to deterministic EM in the context of TFBS motif discovery. Two issues with deterministic motif discovery algorithms are then addressed. Firstly, the EM expressions are generalised in order to improve flexibility. Secondly, a novel heuristic based on motif containment and information content (MCOIN) for determining the most likely motif width is presented and evaluated on realistic synthetic data and previously characterised prokaryotic data.

Chapter 5 begins by extending the framework of Chapter 4 to derive a set of generalised expressions for stochastic EM for motif discovery. This is followed by a practical implementation of these expressions in a novel stochastic EM-based algorithm for motif discovery (MITSU). The developed algorithm is evaluated quantitatively on realistic synthetic data and previously characterised prokaryotic data. The advantages of MITSU over deterministic EM and existing stochastic EM-based approaches for motif discovery are demonstrated.

Chapter 6 presents further validation and extension of MITSU. Further validation is carried out on characterised Alphaproteobacterial motifs before MITSU is applied to motif discovery in uncharacterised Alphaproteobacterial data. The results of these tests motivate the extension of MITSU. MITSU is extended to allow discovery of multiple different motifs and a higher order Markov background model is also incorporated. The effects of these extensions on motif discovery are evaluated on realistic synthetic data and previously characterised prokaryotic data. Finally, a novel information-theoretic of motif palindromicity is presented and its advantages over current approaches for discovering palindromic motifs discussed.

Chapter 7 summarises the results of the project and provides conclusions with regard to the aims of the study, highlighting the contributions made. Some limitations of the current approach are discussed and potential future work is also presented and discussed. The appendices present dataset listings for the tested Alphaproteobacterial datasets and more detailed experimental results.

Chapter 2

Background and Related Work

This chapter presents a review of the most important literature on transcription factor binding site motif discovery and also provides some relevant background knowledge. The Preliminaries section (2.1) begins with a description of position weight matrices (PWMs), which are used extensively throughout the thesis as a model of TFBS motifs. This section also describes sequence logos, a graphical method for representing PWMs. The performance statistics used in the thesis are also defined.

An in-depth review of the literature on TFBS motif discovery is presented (2.2), with a focus on probabilistic algorithms. This section begins with a summary of the various sequence models used in motif discovery algorithms and the assumptions made by each of these models. The different types of data required by different approaches are discussed and a case for using only promoter regions of coregulated genes (rather than adding phylogenetic data) is made. The review extends the algorithmic classification presented by Das and Dai [41]; as well as clearly deterministic and stochastic algorithms, algorithms using variational inference methods and stochastic variations on the EM algorithm are also considered.

The final section (2.3) provides a comparison of existing algorithms for motif discovery, makes the case for the stochastic EM approach adopted in this thesis and discusses how this approach relates to the existing literature.

2.1 Preliminaries

A Position Weight Matrix (PWM) is a commonly used way of representing patterns in biological sequences. Given a number of DNA sequences of equal length (Figure 2.1a), the nucleotides (A, C, G or T) occurring at each position in the sequence are

counted and then normalised by the number of sequences (Figure 2.1b), sometimes adding small pseudocounts to eliminate zero values. It is therefore possible to interpret each column of the PWM as a multinomial distribution with 4 categories, each category giving the probability of a certain base appearing at a certain position within the sequence.

One drawback of the PWM is its assumption of independence between columns (that is, each position in the binding site contributes independently to the binding affinity), which is not always true biologically [32, 124]. It has been shown that the information content of adjacent motif positions is correlated in eukaryotic motifs [56]; in Section 3.2, this phenomenon is confirmed in *E. coli* TFBS motifs. Alternative models of motifs which can account for positional dependencies include Bayesian networks (for example, [18]) and dinucleotide PWMs (for example, [148, 95]); however, these models introduce additional complexities, often increasing the number of parameters. In contrast, the assumption of independence between motif positions allows relatively simple calculations of sequence probabilities using PWMs (this reduces to a multiplication of the relevant values). This convenience and the fact that PWMs are easily interpreted as sequence logos (discussed below) are good reasons to use PWMs in this research. Further, recent quantitative analysis showed that the majority of motifs are well fitted by PWMs [185].

PWMs are generally represented graphically as a 'sequence logo' for ease of reading (Figure 2.2a). This representation uses a stack of letters representing each nucleotide at each position, the height of each letter proportional to its value in the PWM. A widely used variation scales the values in each column by the information content of that column (see page xvii), making clear which positions in the sequences are highly conserved, as positions with low conservation are scaled down (Figure 2.2b). This rescaling assumes that all four nucleotides occur equally often in the original DNA sequence. For many organisms this is usually a reasonable assumption, but for other organisms with a biased GC-content such as *Saccharomyces cerevisiae*, a correction factor is required. By scaling the values in each column by the relative entropy (see page xvii) with respect to the 'background' frequency, the significance of any sequence can be measured regardless of the distribution of nucleotides in the original DNA sequence. Figure 2.2c shows the example rescaled using relative entropy to correct for the low GC-content (38%) of *S. cerevisiae*: note that the G nucleotide in position 7 now has more information than the two T nucleotides in positions 8 and 9, even though all are perfectly conserved. This reflects the fact that nucleotides A and T are more likely

to be found in the original DNA sequence.

Given a set of known binding sites and a set of binding sites predicted by a motif discovery algorithm, the performance of that algorithm on a particular dataset may be assessed at two levels: at the *site* level (that is, assessing the number of correctly predicted sites) and at the *nucleotide* level (that is, for each site, assessing the extent to which the predicted sites overlap the known sites). Following Tompa, *et al.* [168], the following basic statistics at the site level are defined:

- *sTP*: The number of known sites overlapped by predicted sites.
- *sFN*: The number of known sites *not* overlapped by predicted sites.
- *sFP*: The number of predicted sites *not* overlapped by known sites.

Following Tompa, *et al.*, a predicted site is deemed to overlap a known site if they overlap by at least a quarter of the length of the known site (it is noted that this definition is somewhat arbitrary, but thought to be large enough that if the overlapping site were removed, there would be a noticeable change in gene expression). Performance measures at the nucleotide level can be similarly defined:

- *nTP*: The number of nucleotide positions in *both* known sites *and* predicted sites.
- *nFN*: The number of nucleotide positions in known sites but *not* in predicted sites.
- *nFP*: The number of nucleotide positions in predicted sites but *not* in known sites.

Figure 2.3 provides a graphical representation of these statistics.

In this thesis, the performance of a motif discovery algorithm is assessed through its mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*) and the area under the receiver operating characteristic (ROC) curve (AUC). *sSn* (also known as *recall* in machine learning literature [179]) measures the proportion of true positive sites which are correctly predicted as such. *sPPV* (also known as *precision*) measures the proportion of predicted positive sites which are actually true positives. In the context of motif discovery, *sSn* is defined as the fraction of true sites which are predicted and *sPPV* is defined as the fraction of predicted sites which are known to be true (see also [81]); that is: $sSn = \frac{sTP}{sTP+sFN}$ and $sPPV = \frac{sTP}{sTP+sFP}$. These measures can be similarly defined at the nucleotide level.

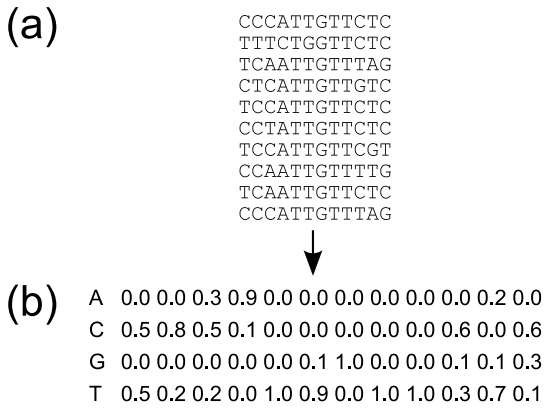

Figure 2.1: Creating a Position Weight Matrix (PWM). The nucleotide counts at each position in a set of DNA consensus sequences (a) can be used to calculate a PWM (b). Adapted from D'haeseleer [48].

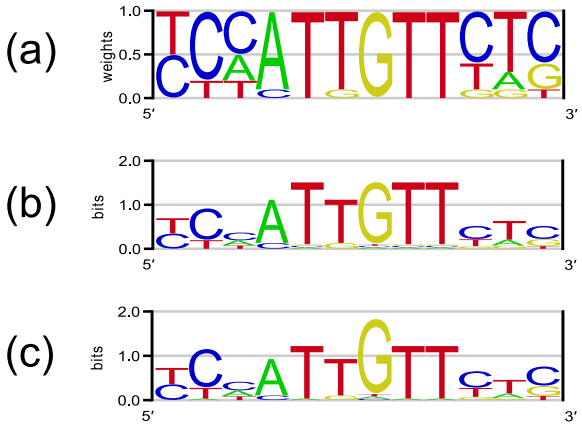

Figure 2.2: Visualising DNA motifs. The PWM in Figure 2.1 (b) can be represented graphically as a sequence logo based on the weights (a), or the weights rescaled using information content assuming an equiprobable background distribution (b) or relative entropy to adjust for biased GC-content (c). Pseudocounts have been added to (b) and (c) in order to eliminate zero values. Adapted from Dhaeseleer [48].

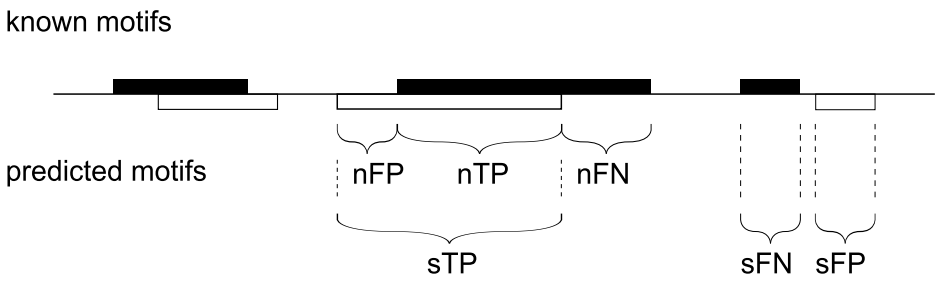

Figure 2.3: Calculating basic statistics at the site and nucleotide level from known motif occurrences (in black) and motif occurrences predicted by a motif discovery algorithm (in white). Adapted from Hu, *et al.* [81].

AUC is the integral of the ROC curve plotting sSn against the site-level false positive rate ($sFPR = \frac{sFP}{(sFP+sTN)}$). The ROC curve is constructed by computing the probability of each possible site being an occurrence of the motif $p(Z_{i,j} = 1|X_{i,j}, \theta)$ and ranking each possible site based on this value. sSn and $sFPR$ are plotted for all possible thresholds of $p(Z_{i,j} = 1|X_{i,j}, \theta)$ and AUC calculated using the trapezoid rule. This is implemented using the ROCR R package [151].

Although it is claimed that no single statistic captures 'correctness' perfectly, Tompa, *et al.* consider some single measures which average some of these quantities. Following Pevzner and Sze [133], the nucleotide level performance coefficient nPC is defined as:

- $nPC = nTP/(nTP + nFN + nFP)$.

Hu, *et al.* also use this definition, noting that nPC is a good overall performance indicator as it can easily be interpreted: it gives a probable range (in [0:1]) that the true binding sites are located around the predicted binding sites, a higher number being a better result [81].

The above classification statistics provide an indication of how well the sites predicted by a motif discovery algorithm match the true sites. In addition to these statistics, the mean absolute error (MAE) and root mean squared error (RMSE) are used to assess methods for determining motif width in Section 4.4, comparing the predicted motif width to the known width. RMSE is a commonly used measure but tends to exaggerate the effect of estimations which are further from the true value; in contrast, MAE treats all error sizes equally according to their magnitude. In most practical situations, the best estimator remains the best regardless of which error method is used [179].

2.2 Probabilistic algorithms for motif discovery

Three major classes of commonly-used motif discovery algorithm have been identified [41]; these are based on the input data required by the algorithm. The first class of algorithm requires a collection of promoter regions for coregulated genes as input and typically uses probabilistic techniques to discover motifs. The second class of algorithm requires orthologous promoter regions of a single gene from multiple species as input and uses phylogenetic footprinting to discover motifs. The third class of algorithm uses a combination of the above approaches. The latter two classes of algorithms have the disadvantage that they are generally more complex and require expert information regarding the similarity between species to achieve good results. This review will therefore concentrate on the first class of algorithm, however, some algorithms from the third class of algorithm will be discussed due to their similarity to the first class.

As noted above, the majority of algorithms in the first class are probabilistic algorithms which search the collection of promoter regions for overrepresented motifs (that is, motifs that occur more often than could be expected by chance alone). These motifs are deemed to be biologically significant. Probabilistic algorithms generally use a probabilistic sequence model (described below) to represent the input sequences. The parameters of this model are then estimated using maximum likelihood or Bayesian inference. With respect to this project, it has been noted that probabilistic approaches are particularly well-suited to motif discovery in prokaryotes, as prokaryotic motifs are generally longer than those in eukaryotes [41]. In contrast, word- or string-based enumerative methods perform better when searching for eukaryotic motifs, which are generally shorter. Probabilistic algorithms also have the advantage of requiring relatively few search parameters. However, because of a reliance on probabilistic models, these algorithms are sensitive to small changes in the input data and are also not guaranteed to find a globally optimal solution since the model parameters are usually found using some form of local search such as the Expectation-Maximization (EM) algorithm.

Most probabilistic motif discovery algorithms work on a similar principle, constructing a model which consists of two components. The ‘motif’ model component $\theta_{1\ldots W}$ describes a set of similar subsequences of fixed width W and is usually represented by a $4 \times W$ PWM, where $\theta_{j,k}$ is the probability that nucleotide k will be found

at position j in the motif:

$$\theta_{1\ldots W} = \begin{bmatrix} \theta_{1,A} & \theta_{2,A} & \cdots & \theta_{W,A} \\ \theta_{1,C} & \theta_{2,C} & \cdots & \theta_{W,C} \\ \theta_{1,G} & \theta_{2,G} & \cdots & \theta_{W,G} \\ \theta_{1,T} & \theta_{2,T} & \cdots & \theta_{W,T} \end{bmatrix} \tag{2.1}$$

The 'background' model component θ_0 accounts for every subsequence in the dataset which is not deemed to be part of the motif model and is represented by a 4×1 vector, where θ_k is the probability that nucleotide k will be found at any position in the background:

$$\theta_0 = \begin{bmatrix} \theta_A \\ \theta_C \\ \theta_G \\ \theta_T \end{bmatrix} \tag{2.2}$$

The complete model $\theta = [\theta_0, \theta_{1\ldots W}]$ is constructed using the 'observed' input data $\mathbf{X}$; it is assumed that each subsequence in the dataset arises from one of the models, but at the outset it is unknown which one. This leads to the concept of the 'latent data' (or 'missing data') $\mathbf{Z}$; in this case, the latent data to be learned is the knowledge of which model each subsequence in the dataset has arisen from. Once the models have been constructed, probabilistic motif discovery algorithms use an optimisation algorithm to simultaneously optimise both models, allowing the values of the latent data to be estimated and statistically significant motifs to be discovered.

It is possible to further classify probabilistic motif discovery algorithms based on the probabilistic method used to estimate the parameters of the model. Bishop [27] considers three main techniques: deterministic methods (such as the EM algorithm), variational methods and stochastic (or Monte Carlo) methods. Using the above classifications, it is possible to build a table of algorithms allowing comparison of both required data and algorithmic method (Table 2.1). Note that the table omits algorithms which use only phylogenetic footprinting or comparative sequence analysis such as CONREAL [22], PHYLONET [172] and PhyloScan [34]; generally, these algorithms are based on some form of alignment or consensus method and are therefore not probabilistic.

	Coregulated genes	Coregulated genes and phylogenetic footprinting
Deterministic methods	EM [99] MEME [10] The Improbizer [5] COMODE [90] cosmo [21] ALSE* [101] fdrMotif* [103] SeedSearch* [17]	OrthoMEME [135] PhyME [153] EMnEM [125]
Variational methods	LOGOS [181]	
Stochastic methods	Gibbs sampler [98] AlignACE [143, 82] MotifSampler [166] SEAM [23] MCEMDA [24] BioProspector† [107] Co-Bind† [76] GLAM2† [63]	PhyloGibbs [149]

Table 2.1: Comparing probabilistic motif discovery algorithms based on required input data and algorithmic method. * denotes discriminative algorithms (see Section 2.2.2). † denotes algorithms capable of handling gapped motifs (see Section 2.2.3).

2.2.1 Different types of sequence model

Motif discovery algorithms make different assumptions about the distribution of motif occurrences within the DNA sequences which make up the input dataset. These assumptions can affect the results of a motif discovery algorithm. The **O**ne **O**ccurrence **P**er **S**equence (OOPS) model is the simplest sequence model and assumes that each sequence in the dataset contains exactly one motif occurrence [99]. Early motif discovery algorithms were based on this model, but were limited: the performance of the OOPS model is reduced when input sequences do not contain a motif, or contain more than one occurrence of the motif. In practice, this is difficult to ensure when

constructing datasets, as gene coexpression does not imply coregulation: if the input dataset is constructed from coexpressed genes (a reasonable method to use), not all sequences in the dataset can be guaranteed to contain a motif [166]. The OOPS model was later extended to the **Z**ero or **O**ne **O**ccurrence **P**er **S**equence (ZOOPS) model, which increases flexibility by allowing sequences which do not contain a motif occurrence. The most flexible sequence model is the **T**wo **C**omponent **M**ixture (TCM) model, which assumes that there are zero or more non-overlapping occurrences of the motif in each sequence in the dataset [10].

The theoretical differences between these types of sequence model are reflected in the number of model parameters. The parameters of the OOPS model are the nucleotide frequencies for each column in the PWM (motif model), the nucleotide frequencies for the background model and a parameter for the motif width. The ZOOPS model requires an additional parameter for the prior probability of a sequence containing an occurrence of the motif. The TCM model replaces this parameter with a different parameter representing the prior probability that any position in an input sequence is the start of a motif occurrence [11]. Many recent motif discovery algorithms are based on the TCM sequence model. This allows greatest flexibility if there is some uncertainty in the distribution of motifs within the input sequences.

2.2.2 Motif discovery algorithms based on deterministic methods

Motif discovery algorithms based on deterministic methods make up a large proportion of commonly used algorithms and use the EM algorithm, or a variant of that algorithm. The EM algorithm is a classic general optimisation technique for finding maximum likelihood solutions for probabilistic models with latent variables [46, 2]. Although the EM algorithm had been proposed many times for special cases, it was not generalised until 1977 [46] and its convergence properties were not proved until 1983 [180]. The EM algorithm is an iterative technique; some initial values for the model parameters are chosen and then two steps are carried out repeatedly until convergence is reached. In the expectation step, or E-step, the current estimates for the model parameters $\theta^{(t)}$ are used to calculate the expected value of the log likelihood function, with respect to the distribution of the latent data given the observed data; this is known as the Q function. The expected value of a random variable is the integral of that variable with respect to its probability measure. That is, if the probability distribution of X admits a

probability distribution function $f(x)$,

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x f(x) \mathrm{d}x. \tag{2.3}$$

The Q function is therefore calculated:

$$Q(\theta|\theta^{(t)}) = \mathbb{E}_{\mathbf{Z}|\mathbf{X},\theta^{(t)}}[\ln p(\mathbf{X},\mathbf{Z}|\theta)] = \int p(\mathbf{Z}|\mathbf{X},\theta^{(t)}) \ln p(\mathbf{X},\mathbf{Z}|\theta) \, \mathrm{d}\mathbf{Z}. \tag{2.4}$$

In the context of motif discovery, this can be viewed as calculating the probability for each width-W subsequence in the dataset that it is an occurrence of the motif, or equivalently estimating the position of occurrences of the motif within the input dataset. The maximisation step, or M-step, evaluates a new estimate of the parameters by maximising the expected value of the log likelihood function:

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} \, Q(\theta|\theta^{(t)}) \tag{2.5}$$

In the context of motif discovery, this can be viewed as reestimating the model parameters given the current estimates for the motif position within the input dataset. It can be shown that each iteration of the EM algorithm is guaranteed to increase the log likelihood of the model, if the likelihood is not already at a maximum. The EM algorithm is deemed to have converged when the increase in log likelihood falls below some small threshold. The EM algorithm has become popular due to its relative simplicity and stability (that is, it converges in a known and steady fashion) [25]. Due to its broad applicability, the EM algorithm has been widely used in a variety of areas, including computer vision and natural language processing. It has also been used in many different probabilistic motif discovery algorithms.

Deterministic methods using coregulated genes

The EM algorithm was first used for motif discovery by Lawrence and Reilly [99]. Although the algorithm was first used to discover protein motifs, it is equally applicable to the discovery of DNA motifs. The algorithm uses the basic EM algorithm as described by Dempster, *et al.* [46] to find a single motif from sequence data in an unsupervised manner; that is, no prior knowledge of the motif is required in order to discover it. Given a motif width to search for, Lawrence and Reilly's algorithm first estimates the start location of the motif within the input sequences and then maximises the expected likelihood of the data given the current estimates of the parameters. These two steps are repeated iteratively until convergence as described above. Unlike previous motif discovery algorithms, which required that the motif occur at the same point

```
procedure EM
  Estimate initial model parameters θ
  until EM algorithm converges, do
    E-step: reestimate motif position using current θ
    M-step: reestimate θ using current motif position
  end
  Print discovered motif
end EM
```

Algorithm 2.1: Pseudocode describing Lawrence and Reilly's original EM algorithm for motif discovery.

within each input sequence, the algorithm presented by Lawrence and Reilly is novel in that the motif position within the sequences does not have to be conserved, allowing more flexibility in choosing input sequences. However, the performance of the algorithm is reduced when given input sequences which do not contain the motif. This is due to Lawrence and Reilly only implementing the OOPS sequence model. Despite this limitation, Lawrence and Reilly successfully applied their algorithm to motif discovery in the CRP dataset (this dataset is discussed in Section 3.6.1). The algorithm has also been the basis for many more recent EM-like motif discovery algorithms such as MEME and The Improbizer. Pseudocode describing Lawrence and Reilly's EM algorithm is presented in Algorithm 2.1.

The MEME (Multiple EM for Motif Elicitation) algorithm [10] is perhaps the best known use of EM for motif discovery. It is based on the method used by Lawrence and Reilly [99] and incorporates a number of features which improve the algorithm for DNA motif discovery. Perhaps most important of these features is the introduction of the ZOOPS and TCM sequence models. As noted above, these sequence models relax the assumption made by the OOPS model that each input sequence contains exactly one motif occurrence and therefore allow MEME to be relatively robust when given some input sequences which do not contain a motif occurrence. The TCM model extends the ZOOPS model to allow for an arbitrary number of motif occurrences per sequence and therefore allows MEME to successfully predict multiple motifs within a single input sequence. The TCM model allows the greatest flexibility in the distribution of motif occurrences and is therefore of greatest benefit when searching for motif sequences in uncharacterised data. It is noted that the TCM model also has the secondary advantage in that it allows MEME to estimate the number of times a motif

appears in a dataset. The second novel feature of MEME is a 'probabilistic erasing' step, which allows more than one motif to be discovered in a dataset; once a motif has been found (it is assumed that the first discovered motif is the most significant), it is effectively 'erased' by a dynamically calculated prior distribution, which allows another run of the EM algorithm to discover another less significant motif. Clearly, the success of this feature depends on any previously discovered motifs being correct. The third novel feature of MEME attempts to address a fundamental problem of EM approaches, that of returning suboptimal solutions as a result of only finding a local maximum of the likelihood function. As noted above, the EM iteratively maximises the log likelihood of the model until convergence is reached. At this point, the gradient of the likelihood function will be 0 (or very close to 0). However, it is entirely possible (and indeed likely) that the maximum point returned by the EM algorithm is a local maximum or saddle point of the likelihood function, rather than a global maximum[1] [23]. MEME addresses this problem by trying a number of different starting points within the parameter space, running the EM algorithm for a small number of iterations at each starting point and evaluating the log likelihood of the model. The EM algorithm is then run to convergence using the starting point which looks to be most promising.

It is clearly desirable for a motif discovery algorithm to automatically determine the width of a motif; an algorithm with this ability would not require the user to know this information in advance. However, automatically determining the width of an unknown motif automatically is not a a trivial problem. Bailey and Elkan introduce a novel 'motif-width' heuristic function in later versions of MEME [11], based on the maximum likelihood ratio test. This function computes a score for each model based on the log likelihood and the number of free parameters and chooses the model with the highest score over all widths. This heuristic was replaced by an width estimator based on the E-value of the resulting multiple alignment [9]. This estimator is discussed further in Section 4.4. Pseudocode describing MEME is presented in Algorithm 2.2.

Although MEME has some limitations, it has been noted for being remarkably consistent in its results in recent comparisons of motif discovery algorithms by Das and Dai [41], Hu, *et al.* [81] and Tompa, *et al.* [168]. It has been used experimentally by Baker, *et al.* [15] and Heikkenen, *et al.* [79], amongst others, returning good results which have been proved experimentally in some cases. It is also available as a web

[1] It may be helpful here to consider the analogous 2-dimensional problem of finding the highest point in Scotland by only walking in an uphill direction: eventually a point will be reached where it is impossible to walk uphill any further, but this point is clearly not guaranteed to be Ben Nevis.

```
procedure MEME
  for N do (where N is the number of motifs to be found)
    for each motif width do
      for each TCM prior parameter value do
        Estimate initial model parameters θ, based on width and TCM prior values
        until EM algorithm converges, do
          E-step: reestimate motif position using current θ
          M-step: reestimate θ using current motif position
        end
        Test if removing outer columns from motif or implementing
              palindrome constraint improves motif score
      end
    end
    Print discovered motif which maximises motif score
    'Erase' motif from dataset by updating prior
  end
end MEME
```

Algorithm 2.2: Pseudocode describing the basic MEME algorithm.

service[2] [8]. In addition, MEME continues to be used as the basis for a number of other algorithms, for example the recent MEME-ChIP [113].

The Improbizer [5] is largely based on MEME [10], but with some small variations based on ideas proposed by Lawrence, *et al.* [98]. The Improbizer differs from MEME in its initial estimates of the motif model and in how this estimate is evolved through the iterative process. The motif model is initially estimated using non-overlapping 6-mers from the dataset, rather than being estimated by maximum likelihood as in MEME. After each iteration of the EM algorithm, a check is carried out to determine whether an addition or subtraction of a column on either side of the motif model will increase the score of the model. Due to its reliance on MEME, The Improbizer shares many of the same limitations as MEME. However, the addition of the check after each EM iteration as described above may reduce the chance of returning 'shifted' motifs. The Improbizer has been used experimentally with some success by Banerjee and Slack [16] and Gaudet, *et al.* [68], as well as in the experimental work carried out by Ao, *et al.* to identify elements that activate expression in *Caenorhabditis elegans* [5]. Like

[2] http://meme.sdsc.edu/meme4_5_0/intro.html

MEME, The Improbizer is available as a web service[3].

COMODE (COnstrained MOtif DEtection) [90] introduces the idea of information content profiles and uses these profiles to constrain a supervised search for motifs. It has been shown that the information content at each position of the motif is proportional to the number of contacts between the transcription factor protein and the DNA at the binding site [121]. That is, there is a direct relationship between information content and the structural footprint of the transcription factor upon the DNA. It follows that motifs bound by structurally similar DNA binding domains should have similar information content profiles. COMODE searches for motifs matching a given information content profile (for example, a gapped motif may have a 'high-low-high' profile). The statistical model underlying COMODE again models the data as motif and background positions. Keles, *et al.* implement the OOPS and ZOOPS models within COMODE but note that using a constrained motif model means that the M-step of the EM algorithm no longer has a closed-form solution; carrying out the M-step requires solving a nonlinear constraint problem at each EM iteration, which can be computationally intensive. This also causes problems with the TCM model as the smoothing step used in MEME cannot be included in a straightforward way; a cutting heuristic is proposed to deal with the discovery of multiple motif occurrences within a single input sequence [90] (this heuristic is discussed further when it is implemented as part of the MITSU algorithm developed in this thesis; see Section 5.3.1).

The advantage of the constraints proposed by Keles, *et al.* is that the discovery of structured but statistically subtle motifs is improved. Based on tests with yeast motifs extracted from SCPD, the supervised search is shown to perform substantially better than unconstrained search when discovering motifs with weak signals using both the OOPS and ZOOPS models. The cutting heuristic is also demonstrated on the even skipped gene in *Drosophila* and shown to discover multiple motifs within a single input sequence [90].

Bembom, *et al.* [21] focus on a number of methodological improvements to COMODE, particularly with regard to the data-adaptive selection of various model parameters, including the choice of sequence model and the Markov background order. The cutting heuristic described by Keles, *et al.* is implemented and extended; tests demonstrate that the heuristic is fairly robust with regard to different choices for the value of the cutting parameter. Following tests of various model selection methods, the Bayesian information criterion is chosen to select the most likely motif width, maxi-

[3] http://users.soe.ucsc.edu/~kent/improbizer/improbizer.html

mum likelihood is used to choose the best fitting sequence model and likelihood-based cross-validation is used to choose the order of the Markov background model. A similar constraint system to COMODE is implemented, but in a more user-friendly manner; again, it is concluded that using a constraint set to supervise the motif search improves performance in the case of low motif abundance, or structured motifs with low motif conservation. The improved version of COMODE is repackaged as cosmo, which is available as a standalone application and Bioconductor package.

Discriminative motif discovery algorithms

Despite their differences, the previously discussed motif discovery algorithms have had one similarity, in that all the input sequences have been believed to contain a motif. That is, sequences believed not to contain a motif have not been intentionally added to the input set; it is assumed that the motif discovery algorithm used will be robust enough to deal with any of these sequences if they are present (although clearly this depends greatly on the type sequence model used: see Section 2.2.1). However, there exists a class of algorithms which require sequences believed not to contain a motif as part of the input dataset. These algorithms are known as 'discriminative motif discovery algorithms' and generally use these sequences to verify and score any discovered motifs: if motifs discovered within sequences believed to contain a motif are also discovered within sequences believed not to contain a motif, they are less likely to be biologically significant. Three EM-based discriminative motif discovery algorithms are discussed below, although many more algorithms utilising other methods exist, including DEME [140], DIPS [152] and DME [155].

SeedSearch [17] takes a two stage approach, first exhaustively searching for all patterns given a motif width (for example, all width-7 patterns), then filtering out the most significant patterns using a discriminative approach and a set of sequences believed not to contain a motif. In the second stage of the algorithm, an EM approach is used to create a PWM based on the most significant patterns found in the first stage of the algorithm. Although the presented results show a great deal of variance, Barash, *et al.* note two potential improvements over algorithms such as MEME. Firstly, SeedSearch runs faster than MEME, as the second stage of the algorithm runs very quickly after the initial preprocessing has been carried out. Secondly, SeedSearch tends to return fewer spurious motifs (for example, poly A's), although this is dependent on the input sequences used.

ALSE [101] uses the EM algorithm to discover a motif which is then verified after

convergence of the EM algorithm. Although Leung and Chin claim their technique is based on the simulated annealing method, Das and Dai [41] note that there is no random walk step, so the technique does seem to be most closely related to EM. Leung and Chin suggest that using the *p*-value as a score of correctness (as in SeedSearch [17]) could be improved by performing a probabilistic analysis rather than a hyper-geometric analysis. However, it is unclear how much of the improvement in the presented results is due to this new scoring method alone. ALSE is shown to improve on MEME [10] and SeedSearch in experiments on simulated data and in tests on real biological data (using the SCPD yeast [187] and TRANSFAC eukaryotic [178] databases); however, results seem to be highly dependent on the initial parameter estimation and may not be globally optimal, due to the EM approach. ALSE has not been widely used in experiments, but is available as a download[4].

fdrMotif [103] is also a discriminative motif discovery algorithm based on an EM approach. However, unlike ALSE, fdrMotif aims to combine the optimisation and the statistical scoring of the discovered motif model, rather than optimising and then scoring afterwards. fdrMotif is based on the MEME framework, but with a variation on the normalisation procedure in the E-step of the EM algorithm which tests for motif significance. Like ALSE, fdrMotif is reliant on 'many sets' of sequences believed not to contain a motif and generates random sequences if not enough are available. Li, *et al.* [103] compare fdrMotif to MEME based on tests using eukaryotic ChIP and 'simulated ChIP' data. On both types of data, fdrMotif compares well to MEME in terms of precision and slightly outperforms MEME in terms of sensitivity [168]. However, there does not seem to be a clear advantage, except for not having to score motifs after they have been discovered. Like ALSE, fdrMotif does not appear to have been used experimentally besides the tests run by Li, *et al.* [103].

SeedSearch [17], ALSE [101] and fdrMotif [103] have not been widely used in bioinformatic analyses, besides the tests carried out by their respective authors. Although discriminative algorithms clearly have promise and counterintuitively, sequences which do not contain a motif can be helpful in motif discovery, the improvement they give over non-discriminative methods appears to be marginal. As with the other EM-based algorithms, there is no guarantee that any discovered motifs are globally optimal. Both Li, *et al.* [103] and Redhead and Bailey [140] reach a similar conclusion regarding the improvements of discriminative methods over traditional non-discriminative methods, Redhead and Bailey noting that although their discriminative

[4] http://alse.cs.hku.hk

algorithm DEME gives good results on synthetic data, on real-world data DEME was shown to be as good as, but not better than non-discriminative algorithms. Algorithms implementing a discriminative approach remain an active area of research, for example, the recent DLocalMotif algorithm [119].

Deterministic algorithms using coregulated genes and phylogenetic footprinting

Like The Improbizer [5], OrthoMEME [135] is based on the MEME [10] framework and uses a generalisation of MEME in an attempt to solve the 'two-species heterogeneous data problem'. In this problem, the input sequences to the algorithm consist of a collection of coregulated genes from one genome together with their orthologous genes in another closely related genome. Although there are clear advantages to this heterogeneous approach, it raises the additional issue of how to treat the data from different species. OrthoMEME uses an additional parameter to denote the species of the input sequence, but is otherwise very similar to MEME. Like MEME, OrthoMEME tries to address the problem of comparing motif scores over different widths. However, it does this by simply dividing the log likelihood of the model by the motif width, based on the somewhat simplistic assumption that log likelihood scales linearly with increasing motif width (as noted above, model comparison over different motif widths is complicated by the issue of different numbers of model free parameters). OrthoMEME has been used very little experimentally, although Prakash, *et al.* [135] present results of experiments on a number of eukaryotic genomes.

PhyME [153] is a generalisation of OrthoMEME [135] that allows for input sequences from any number of different related genomes (given enough computing power and processing time). The idea behind PhyME is to combine two important aspects of motif significance - overrepresentation and cross-species conservation - into one probabilistic score. Unlike the algorithms discussed previously, which have generally used a PWM-based approach to model the motif of interest, PhyME uses a Hidden Markov Model (HMM). The parameters of the HMM are initialised and then trained using the Baum-Welch algorithm (a particular form of the Generalised EM (GEM) algorithm), which converges to maximum likelihood parameters using EM. In order to incorporate phylogenetic relationships within this model, PhyME uses a probabilistic evolutionary model as part of the computation of subsequence probabilities (that is, the probability of generating a width-W subsequence from the current model). This evolutionary model was initially proposed by Sinha, *et al.* to model binding site evolution in *Drosophila* species [154]. Although PhyME is relatively complex in training the

HMM parameters, it scales well with increasing numbers of input sequences, allowing a large number of input sequences to be analysed simultaneously in a relatively short time. Like MEME, PhyME carries out a small number of iterations at different HMM parameter starting values and calculates the likelihood of the model each time, before running the most promising starting values to convergence in an attempt to discover the global maximum likelihood. Sinha, *et al.* [153] present the results of PhyME on a number of eukaryotic datasets; the algorithm has also been used experimentally by de Jong [43]. PhyME is available to download on request from the authors.[5]

Like OrthoMEME, EMnEM (Expectation Maximization on Evolutionary Mixtures [125]) is based on the MEME framework and is claimed to be an evolutionary extension of MEME. EMnEM combines the mixture model used by MEME and other variant algorithms with a probabilistic model of evolution, which considers DNA sequences from different species to have been generated by an unseen ancestral sequence. Different subsequences of this ancestral sequence are assumed to evolve at different rates. Subsequences with a slow evolution rate are more conserved and therefore deemed to be motifs; subsequences with a higher evolution rate are less conserved and are deemed to be part of the background. EMnEM differs from all of the previously discussed algorithms in that it requires input sequences to be aligned in order to give a smaller search space. However, this in turn constrains the input data to sequences from closely related species which can be aligned; both Moses, *et al.* and Das and Dai [41] note that this may not be possible with species at large evolutionary distances. In tests, EMnEM was compared with MEME on data from five *Saccharomyces* genomes taken from the SCPD database [187] and was shown to consistently rank the 'correct' motif higher than MEME. MEME did find the motifs, but ranked them behind 'false positive' results; it is claimed that the use of phylogenetic information can help reduce the number of false positives, allowing EMnEM to outperform MEME in this regard. EMnEM is available as a software download[6] but has not been used widely in experiments.

Approaches for chromatin immunoprecipitation (ChIP) data

Chromatin immunoprecipitation (ChIP) coupled either with microarray (ChIP-chip) or massively parallel DNA sequencing (ChIP-seq) is a relatively new and powerful method for experimentally mapping the genome-wide binding sites of transcription

[5] http://veda.cs.uiuc.edu/cgi-bin/phyme/download.pl
[6] http://www.moseslab.csb.utoronto.ca/alan/emnem.html

factors. The resolution of binding regions identified by ChIP-seq is usually on the order of a few hundred base pairs. These regions are ideal candidates for scanning with motif discovery algorithms. However, it has been noted that traditional motif discovery algorithms cannot handle the increased volumes of data output by ChIP-x experiments. The primary limiting factor is computational efficiency. Bailey has noted that many algorithms (including MEME) do not scale well to very large (ChIP-scale) datasets; the running time of MEME is noted to scale non-linearly with dataset size, which makes it impractical with very large datasets [7]. Ma, *et al.* also draw a similar conclusion [112]. A secondary problem of choosing initial parameter values is also noted by Bi; it is claimed that EM convergence becomes very slow when poor initial parameter values are chosen and that this problem becomes worse as the dataset size increases to ChIP-scale [23]. Recent trends in computational motif discovery have therefore attempted to address the need for more powerful and robust approaches in handling high throughput data (for example, ChIP-seq).

The number of motif discovery algorithms specifically designed for ChIP data has increased dramatically in recent years. While some aspects of traditional algorithms are retained (for example, DREME [7] uses a discriminative approach), many are specific to their application on ChIP data. For instance, traditional motif discovery algorithms generally assume no prior knowledge regarding motif position (that is, all positions within a sequence are *a priori* equally likely to be motif occurrences). However, algorithms designed for ChIP data often employ positional information as transcription factors are more likely to bind to areas near the peaks of the ChIP intensity profile. Examples of this include CentriMO [14], ChIPMunk [94] and POSMO [112].

Discussion of deterministic methods

Although motif discovery algorithms based on deterministic methods are similar in that they all use the EM algorithm in some form, there are differences between them which allow some algorithms to outperform others. As noted above, Lawrence and Reilly's EM algorithm [99] is limited in that it only implements an OOPS sequence model, which reduces the performance of the algorithm when sequences which do not contain a motif are included in the input dataset. As the OOPS model assumes that *every* sequence contains a motif, a genuine motif in a few of the input sequences may be disregarded in favour of a spurious motif that occurs in all of the input sequences. This is clearly not desirable, although the situation is realistic: it is often not possible to say for certain whether a sequence contains a motif or not. The increased flexibility

of the TCM sequence model introduced by Bailey and Elkan [10] allows increased performance in such situations and is more likely to return the genuine motif. As a result, the TCM model has been implemented in subsequent deterministic motif discovery algorithms (SeedSearch is an exception, implementing only the ZOOPS model [17, 101]). Lawrence and Reilly's EM algorithm is also limited in terms of finding the global maximum likelihood and as a result is more likely to converge to a local maximum in the likelihood. Again, this results in a less significant motif being returned. Combined with the fact that EM can only discover one motif from a dataset, this severely limits the performance of the algorithm. As noted above, Lawrence and Reilly's EM algorithm was the first motif discovery algorithm which could deal with unaligned input sequences. This feature has been used in almost all motif discovery algorithms since, with the exception of EMnEM [125], which requires sequences to be aligned before motif discovery can begin.

MEME [10] and The Improbizer [5] deal with the issue of finding the global maximum in different ways. As noted above, MEME tests a number of different starting points in the parameter space (for both the model parameters and the TCM prior (or 'mixing') parameter). It is clear to see how this can improve the search for a global maximum; however, the success of such an approach depends greatly on the complexity of the likelihood function. In contrast, The Improbizer takes each non-overlapping 6-mer from the dataset as a starting point, adding or removing columns as required to improve the score function. Theoretically, the approach used by MEME should be more thorough in its exploration of the parameter space due to the different start points for the TCM prior parameter used. Although the value of this parameter changes with each iteration of EM, choosing different initial values for this parameter should improve performance, especially if the likelihood function is expected to be complex. PhyME [153] works on a similar principle to MEME, trying out different initial HMM parameter values in an attempt to find the global maximum likelihood.

Like MEME, The Improbizer can search for multiple motifs in a single dataset by using an 'erasing' prior distribution. If it is expected that promoter regions contain more than one motif (as noted above, gene expression can be initiated by cooperative binding of multiple transcription factors), it makes sense to search for more than one motif. The success of this sequential approach to multiple motif discovery is dependent on previously discovered motifs being correct; for instance, if a motif is only partially discovered and erased on one pass, the part of the motif which was not erased may affect motif discovery in subsequent passes of the algorithm. The use of the sequential

approach would appear to make MEME more powerful than The Improbizer when searching for multiple motifs, as a more thorough examination of the parameter space is made, making it more likely that any previously discovered motifs would be correct. The discriminative motif discovery algorithms noted only search for single motifs. This is likely a consequence of their approach; similar to Lawrence and Reilly's EM algorithm, it must be known in advance that the input sequences in the 'positive' set contain a motif (and the sequences in the 'negative' set contain no motif occurrences). To know in advance that an input sequence definitely contains more than one *known* motif is very unlikely, if not impossible, when dealing with uncharacterised data.

As noted above, automatically determining the width of an unknown motif is a hard problem, as models of different widths cannot be directly compared using only the likelihood function of the models [11]. This is a consequence of the different numbers of free parameters of the models; the maximum value of the likelihood function is bound to increase by the addition of more parameters (that is, increasing motif width). Bishop [27] makes a similar observation regarding the phenomenon of 'overfitting' and the numbers of free parameters when comparing models. MEME currently uses a score function based on the E-value of the resulting multiple alignment [9]. In contrast, OrthoMEME [135] uses a much more simplistic scoring function, dividing the likelihood by the width of the motif model. While this may work as a rough approximation, it does not take into account the issues of overfitting and would not be able to handle alternative motif forms such as palindromic motifs, where the motif width and the number of free parameters do not increase together (as a guide, the number of free parameters in palindromic motif models is around half that of regular motifs of the same width). The scoring function used by MEME takes these alternative motif forms into account and strongly penalises models of higher widths. Heuristics to determine the most likely motif width are discussed further in Section 4.4.

Although discriminative algorithms such as fdrMotif [103] seem to offer little improvement over non-discriminative algorithms in terms of results in *de novo* motif discovery, they have some interesting features which could be adapted to improve non-discriminative algorithms. fdrMotif and EMnEM [125] use similar methods to control the number of false positive results (that is, sites incorrectly identified as a motif). This in turn increases the proportion of correct motifs returned by the algorithm. fdrMotif uses an additional parameter called the false discovery rate, which was originally developed to control the number of false positive results in multiple hypothesis testing. Given a PWM representing a subsequence, the log likelihood score is computed and

the subsequence deemed to be significant if the score is equal to or greater than a certain threshold. EMnEM uses a simpler but similar method, calculating the log ratio of the likelihood of the discovered motif to the likelihood of a sequence known not to be a binding site. If the log ratio is greater than zero, the discovered motif has a greater likelihood than the false positive and should be returned. Controlling the number of returned false positive results would clearly improve the overall performance of motif discovery algorithms, even though such an approach could be overly cautious. Of all of the motif discovery algorithms reviewed by Tompa, *et al.* [168], Weeder [131] gave the best results. This is thought to be due at least in part to the cautious use of the algorithm; the overall performance of Weeder was increased by not returning results which may possibly be wrong.

Algorithms based on variational methods

Although deterministic methods have been used successfully for motif discovery, they often have issues of computational and analytical intractability with regard to the two steps in the EM algorithm. The E-step is usually the more problematic, as it requires the evaluation of the expectation of the log likelihood with respect to the posterior distribution of the latent variables $p(\mathbf{Z}|\mathbf{X},\theta)$ (Equation 2.4). Although this is relatively straightforward for simple problems, for many practical problems, it is not possible to evaluate the posterior distribution or evaluate the expectation of the log likelihood with respect to this distribution. There are a number of reasons why it may not be possible to evaluate the posterior distribution. Most often, it is that the posterior distribution takes a complex form which is not analytically tractable. However, the posterior distribution may also have so many dimensions that it becomes impossible to work with directly. In addition to these problems, to evaluate the posterior distribution, the distribution must be normalised, which requires some form of integration or summation, depending on the type of variables involved. The required integration may not have an analytical solution, or the dimensionality of the problem may be too high to compute numerical solutions. In contrast, summations over all possible states of discrete variables are always possible in theory, but the exponential number of states may be prohibitively expensive to compute in practice, depending on the dimensionality of the problem. In situations where evaluation of the posterior distribution is impossible, approximate approaches must be used. Approximate approaches fall into two categories: deterministic and stochastic. Deterministic approximate approaches will be discussed in the next section; stochastic approximate approaches will be discussed in the follow-

ing section.

Deterministic approximate approaches are based on analytical approximations to the posterior distribution, for example by assuming that the posterior distribution takes a form which can be factorised in a particular way, or by assuming that it has a parametric form which can also be simplified. Deterministic approximate approaches generally have the advantages that they are not computationally demanding and scale well to large problems; indeed, such approaches have wide applicability. Clearly, the main disadvantage of a deterministic approximate approach is that the success of the approximation relies on being able to find an appropriate and tractable approximating distribution, which is not always a trivial problem. A family of deterministic approximation methods known as variational inference will be discussed.

Variational inference aims to approximate the complicated posterior distribution with a simpler distribution, chosen from a restricted family of distributions [27]. The family of simpler distributions is necessarily restricted so that it comprises of only tractable distributions (that is, distributions that can be integrated); however, care must be taken to make the family of distributions rich and flexible enough to provide a good approximation to the posterior distribution. The optimal form of the simpler distribution is obtained by minimising the difference between the posterior distribution and the simpler approximating distribution. This difference is typically measured using the Kullback-Leibler (KL) divergence[7], which has some useful information theoretic properties and allows one to compute the similarity of two distributions. Minimising the difference between the two distributions allows a lower bound on the log likelihood to be maximised, in a similar manner to the maximisation of the likelihood function seen previously [182, 181].

Formally, given observed variables $\mathbf{X}$ and latent variables $\mathbf{Z}$, the goal of variational inference is to find an approximation for the intractable posterior distribution $p(\mathbf{Z}|\mathbf{X})$ (following Bishop [27], it is assumed for simplicity that model parameters θ are absorbed into $\mathbf{Z}$). In order to approximate the posterior distribution, an approximating distribution $q(\mathbf{Z})$ is introduced. $q(\mathbf{Z})$ can be any tractable distribution. The log

[7]The KL divergence is known as a 'divergence' measure rather than a 'distance' measure because of its asymmetric properties. That is, the KL divergence measured from one probability distribution $p(x)$ to another $q(x)$ is not equal to the KL divergence measured from $q(x)$ to $p(x)$. Bishop [27] notes that using the reversed form of the KL divergence results in a related but quite different deterministic approximation method known as 'expectation propagation' (EP). EP is not discussed here as Bishop also notes that using EP for multimodal distributions can lead to poor approximations, in particular where EP is applied to mixtures.

marginal probability $\ln p(\mathbf{X})$ can be written[8]:

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + D_{\mathrm{KL}}(q||p), \tag{2.6}$$

where

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X},\mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z} \tag{2.7}$$

and the KL divergence of p from q is defined (in the continuous case) as:

$$D_{\mathrm{KL}}(q||p) = -\int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}. \tag{2.8}$$

If it were possible to choose any distribution as $q(\mathbf{Z})$, then $D_{\mathrm{KL}}(q||p)$ would reduce to zero when $q(\mathbf{Z})$ is equal to the posterior distribution $p(\mathbf{Z}|\mathbf{X})$. However, because $p(\mathbf{Z}|\mathbf{X})$ is intractable, it is only possible to minimise $D_{\mathrm{KL}}(q||p)$. Doing so allows $q(\mathbf{Z})$ to be as close as possible to $p(\mathbf{Z}|\mathbf{X})$ while remaining tractable.

Beal and Ghahramani [20] incorporate variational inference into the EM framework in the variational Bayesian EM algorithm (VBEM; also simply known as the variational EM, or VEM, algorithm). Comparing the VEM algorithm to the EM algorithm, Beal and Ghahramani note that instead of maximising the likelihood as in the EM algorithm, the VEM algorithm maximises a lower bound on the likelihood. In all other ways, the two algorithms are identical, except that it is expressed in the terms of the expected parameters. Indeed, it is noted that the VEM M-step computes a distribution over parameters, rather than a maximum likelihood estimate and that if the distribution is restricted to a point estimate (that is, a Dirac delta function), the VEM algorithm reduces to the ordinary EM algorithm.

VBEM has been applied to other research in computational biology (for example, modelling transcriptional regulatory networks [106]), but only forms the basis of one motif discovery algorithm. LOGOS (Local and Global motif Sequence model, [181]) uses variational EM for motif discovery; unlike the deterministic algorithms discussed in Section 2.2.2 which used a single probabilistic model, LOGOS takes a different approach and consists of two interacting submodels [181, 174], each modelling a different aspect of the input data. A local alignment model known as Hidden Markov Dirichlet-Multinomial (HMDM, [182]) is used to incorporate biological prior knowledge and positional dependencies within a single motif subsequence. As in PhyME, a Hidden Markov model (HMM) is used to globally model all the motifs within the

[8] A similar decomposition is used for extensions of the EM algorithm such as the ECM (expectation conditional maximization) algorithm; using such a decomposition allows both steps of the EM algorithm to be viewed as different maximisations.

dataset. The use of the HMDM model is novel in that it allows prior knowledge to be used to return more biologically plausible motifs. For example, as noted above, motif positions with a high information content are more likely to be adjacent to positions which also have a high information content [56]. Xing, *et al.* use this positional clustering as one of the main motivations for the HMDM model. Use of the HMDM model should theoretically model these features better than regular models, which assume independence between motif positions. While the HMDM model is useful for these motif-level features, this model alone cannot predict the locations of motifs within the dataset so the HMM model is used for this purpose. As noted above, it is intuitively reasonable that incorporating prior knowledge as in the HMDM model can help improve the results of motif discovery algorithms. However, incorporating this knowledge increases the complexity of the full data model. Variational EM is therefore used to approximate the complex distribution in the E-step with a simpler tractable distribution, the optimal form of which is obtained by minimising the KL divergence as described above.

Like many of the EM-based motif discovery algorithms discussed above, LOGOS is capable of discovering multiple motifs in a dataset [181, 88]. However, unlike EM-based algorithms, LOGOS discovers multiple motifs simultaneously through the HMM. Xing, *et al.* argue that simultaneous discovery of multiple motifs is a better strategy than sequential discovery, particularly when motif concentrations are high. It is argued that in sequential discovery, less significant motifs are treated as background positions, which can potentially cause suboptimal estimation of both the background *and* the motif model parameters.

Xing, *et al.* test LOGOS on semi-realistic data and real eukaryotic data (yeast data from the SCPD database [187] and *Drosophila* data). In these tests, LOGOS appears to outperform MEME and AlignACE (see below). Xing, *et al.* claim that the results of their tests suggest that algorithms such as MEME or AlignACE are not powerful enough to handle non-trivial motif detection. However, Xing, *et al.* note that the datasets used are unusually large (input sequences are up to 5,000nt in length) and that results may vary depending on the quality of the dataset. In addition, LOGOS is not tested on prokaryotic data; this would be a study worth carrying out. LOGOS is not available as a web service or software implementation so it has not been included in independent evaluations of motif discovery algorithms [41, 81, 168] or used experimentally other than in the tests carried out by Xing, *et al.*. As a result, it is not possible to compare LOGOS to other motif discovery algorithms. While LOGOS has

some advantages and it is clearly an interesting technique, there are some disadvantages. Firstly, as noted above, approximate methods are inherently inexact; this may become a problem depending on the complex distribution that is being approximated. Clearly, the closer the approximating distribution is to the real distribution, the better the results will be. However, the form of the approximating distribution is bound by the set of tractable distributions. The relative complexity of LOGOS is another issue; however, this would appear to be the cost of incorporating prior biological knowledge into a motif discovery algorithm.

Published evaluations of deterministic motif discovery algorithms

Two extensive evaluations of motif discovery algorithms [168, 81] have included algorithms based on deterministic methods. Tompa, *et al.* [168] compare 13 motif discovery algorithms, including MEME [10] and The Improbizer [5]. These algorithms were tested using datasets created from the TRANSFAC [178] database. Overall, MEME was shown to outperform The Improbizer in motif discovery in all datasets, particularly in datasets containing yeast transcription factors. MEME also outperforms The Improbizer in terms of average site performance and higher precision, although Tompa, *et. al* urge caution in comparing algorithms solely on precision as this statistic is undefined for datasets where no motif was predicted. Despite MEME's higher performance as noted, MEME and The Improbizer are comparable in terms of overall nucleotide level specificity (that is, the number of motif positions correctly predicted). This means that although the motifs that MEME discovers are good, MEME completely misses other motifs. In contrast, although motifs discovered by The Improbizer are less good, it picks up more motifs overall. Tompa, *et al.* note that there are limitations to their evaluation, due to the variations in the tested algorithms (for instance, varying assumptions regarding the input data). Perhaps most importantly, because TRANSFAC only contains eukaryotic transcription factors, the datasets used were restricted to eukaryotic data. It is noted that it would be beneficial to carry out a similar evaluation using prokaryotic data. Hu, *et al.* [81] carry out such an evaluation (using *E. coli* data) on 5 motif discovery algorithms, however MEME is the only deterministic algorithm evaluated. Although this means that a comparison of deterministic algorithms using prokaryotic data cannot be made here, Hu, *et al.* do compare MEME to other motif discovery algorithms. MEME is found to have the best sensitivity and has the highest motif level success rate. MEME is also deemed to be one of the best algorithms in the test in terms of scalability (finding motifs as the input dataset becomes

larger). Although Tompa, *et al.* note that there is no 'gold standard' for comparing motif discovery algorithms, it is clear from the results presented by the creators of other deterministic algorithms that MEME is the current benchmark deterministic motif discovery algorithm. This is perhaps not surprising, since MEME is one of the oldest deterministic motif discovery algorithms and has been steadily improved since its original description. Clearly, later motif discovery algorithms have not had as much time to make an impact.

Model Representation

Many of the above motif discovery algorithms use similar representations of the motif and background model. With the exception of PhyME [153], the probabilistic algorithms use a PWM to represent the motif model and a similar single column vector representing the background model. It may seem that using a PWM to represent a motif is overly simplistic and by doing so, much of the detailed features of the real-world motif are lost. However, PWMs have been shown to be a very good approximation of the real-world situation; along with their simplicity and readability, this makes them a good method of model representation [114]. Some of the algorithms above use additional heuristics in order to better model DNA sequence motifs, which often conform to known alternate forms. These alternate forms can cause problems for motif discovery algorithms which do not include some additional heuristic to deal with them. Lawrence and Reilly [99] note that alternate model representations for dealing with palindromic sequences and gapped motifs can be made by changing the formulae at the heart of their EM method, although there are no clear details on exactly how this should be done. Bailey and Elkan [11] explain in more detail how DNA palindromes are represented in MEME. MEME models DNA palindromes by enforcing a constraint on the parameters of corresponding columns of the PWM, while still allowing columns to be independent. An additional heuristic allows MEME to automatically choose whether to enforce the palindrome constraint or not, depending on whether or not it improves the motif score function. Although The Improbizer [5] is very similar to MEME, Ao, *et al.* do not mention any special representations for handling palindromic or gapped motifs.

It is perhaps also worth noting here how the motif discovery algorithms work with the input dataset. Most algorithms follow the method proposed by Lawrence and Reilly [99], where given a set of input sequences and motif width W, the set of input sequences is split up into every possible width-W subsequence, each of which is sim-

plistically assumed to be independent and a potential motif. However, PhyME [153] uses a different splitting technique, parsing the input DNA sequences into a set of non-overlapping occurrences of the motif and background models. This seems to imply that a 'hard', or deterministic assignation is made to each input sequence 'occurrence' and not a probabilistic one.

Issues with deterministic methods

Although deterministic methods have been used successfully for motif discovery, some limitations remain. Analysis has shown that the EM algorithm converges in a known and predictable way towards a maximum in the likelihood function [180]. This behaviour would be perfectly adequate if there were a single point of maximum likelihood. However, the likelihood function is seldom a smooth function with one maximum point, but usually a complex function with a number of local maxima and saddle points, all of which can act as convergence points for the EM algorithm. In the context of motif discovery, numerous local maxima can correspond to biologically significant motifs when more than one motif is present in the dataset, however, ideally algorithms should discover the most significant motif, corresponding to the global maximum point in the likelihood function, followed by other motifs in order of significance. Algorithms such as MEME [10] and PhyME [153] attempt to find the global maximum likelihood point by running the EM algorithm from a number of different initial starting parameters, but this still cannot guarantee convergence to the global maximum, depending on the complexity of the likelihood function. Hu, *et al.* [81] note that the effectiveness of a 'multi-start' approach is limited for large search spaces such as those found in datasets with long sequences and that experimentation is required to judge how much of an impact complex likelihood functions have on the performance of motif discovery algorithms.

It seems intuitive that incorporating additional information regarding motif sequences will significantly improve the results of motif discovery algorithms [128]. Indeed, experiments have shown the value of incorporating various forms of additional prior information to MEME. Bailey and Elkan [11] demonstrate that adding a heuristic for automatically identifying (quasi-)palindromic motifs and adding information regarding which motif elements share common properties (this is demonstrated using amino acids in protein analysis but is applicable to a lesser extent in DNA) can improve the ability of MEME to discover motif sequences. Bailey, *et al.* [9] demonstrate that using position-specific prior distributions to incorporate additional information

can substantially improve the results returned by MEME. However, the addition of prior knowledge increases the complexity of the probabilistic model, which can lead to either the E-step or the M-step, or both steps, in the EM algorithm becoming either analytically or computationally intractable. Two relatively simple extensions of EM attempt to address the issues of intractability in either the E-step or M-step of the EM algorithm, as discussed in the next section.

Extensions of the EM algorithm

The expectation conditional maximization (ECM) algorithm [120] addresses the problem of an intractable M-step by carrying out several computationally simpler constrained maximisations of the likelihood function rather than one maximisation over all the model parameters [25]. For example, the model parameters may be grouped into a number of arbitrary sets, then each set maximised separately while the other groups are held fixed. The ECM algorithm is a special case of the generalized EM (GEM) algorithm. The GEM algorithm does not attempt to maximise the likelihood function in the M-step, but instead aims to change the model parameters in such a way that the likelihood function is increased (typically using some form of nonlinear optimisation strategy such as the conjugate gradients algorithm). It can be shown that the GEM algorithm converges to a maximum and each iteration of the GEM algorithm is guaranteed to increase the likelihood function. However, the GEM algorithm usually converges significantly slower than the EM algorithm [84].

It is also possible to similarly generalise an intractable E-step of the EM algorithm, by viewing both the E-step and the M-step of the EM algorithm as increasing the same function [130]. This is particularly useful, as the E-step often includes high-dimensional integration, which is hard to determine [25]. In this 'incremental' view of the EM algorithm, known as the incremental EM (IEM) algorithm [130], the distribution for only one of the model parameters is recalculated in each E-step. Again, it can be shown that the IEM algorithm converges in the same way as the EM algorithm. Although deterministic exact approaches are used as alternatives to the EM algorithms in areas such as image segmentation and optical communication (GEM) and analysing robotic sensor data (IEM), they have not so far been used in motif discovery algorithms. This is likely because the other issue of EM (finding a global maximum) is more pressing and requires alternative approaches to solve. These approaches will be discussed in the following sections.

2.2.3 Motif discovery algorithms based on stochastic methods

As noted above, stochastic approximate methods are a widely-used alternative to variational methods when evaluation of the posterior distribution is impossible. These methods are collectively known as Monte Carlo methods[9] [27, 115]. As noted in Section 2.2.2, evaluation of the posterior distribution is required in order to evaluate the expectation of the log of the complete likelihood with respect to the posterior distribution of the latent variables. Monte Carlo techniques approach this problem in a slightly different way based on numerical sampling: the evaluation of the posterior distribution is not of direct interest, but the evaluation of a function (for example, the expectation) with respect to the posterior distribution is of interest. The general idea behind Monte Carlo methods is to obtain a set of independent samples from the posterior distribution. These samples allow the required expectation to be approximated by a finite sum, which can be calculated quite easily. This approximation gets better and better with increasing numbers of independent samples due to the law of large numbers; given infinite computational time, Monte Carlo methods generate exact results (the approximation arising from the fact that only finite computational time is available in any practical situation) [27]. Simple Monte Carlo strategies for evaluating the expectation of functions such as importance sampling and rejection sampling suffer from a number of severe limitations in problems with more than one or two dimensions. In higher dimensional problems, a framework known as Markov chain Monte Carlo (MCMC), which scales well with increasing dimensionality is often used.

The most commonly used Monte Carlo method for motif discovery is Gibbs sampling [70]. Given a complex multivariate distribution $p(\mathbf{z}) = p(z_1, \ldots, z_M)$, from which direct sampling is impossible, Gibbs sampling allows samples to be taken from $p(\mathbf{z})$ by considering a series of conditional distributions; at each step, the value of one of the variables z_i is replaced by a value drawn from the distribution of that variable conditioned on the remaining variables $p(z_i|\mathbf{z}_{\setminus i})$. For example, suppose that $p(\mathbf{z}) = p(z_1, z_2, z_3)$ and the values of the variables at step t are $z_1^{(t)}$, $z_2^{(t)}$ and $z_3^{(t)}$. At step t+1, $z_1^{(t)}$ is replaced by a sample from the conditional distribution:

$$z_1^{(t+1)} \sim p(z_1|z_2^{(t)}, z_3^{(t)}). \tag{2.9}$$

[9] Such methods were developed at Los Alamos during the development of the hydrogen bomb in the late 1940s and named after the Monte Carlo Casino, where inventor Stanisław Ulam's uncle often gambled.

Next, $z_2^{(t)}$ is replaced by a sample from the conditional distribution

$$z_2^{(t+1)} \sim p(z_2|z_1^{(t+1)}, z_3^{(t)}). \tag{2.10}$$

Finally, $z_3^{(t)}$ is replaced by a sample from the conditional distribution

$$z_3^{(t+1)} \sim p(z_3|z_1^{(t+1)}, z_2^{(t+1)}). \tag{2.11}$$

The variables now have values $z_1^{(t+1)}$, $z_2^{(t+1)}$ and $z_3^{(t+1)}$. The procedure is repeated for steps t+2,...,T. Although from the algorithm description the practicality of using Gibbs sampling clearly depends on being able to find appropriate conditional distributions, this is usually possible [27].

Stochastic methods using coregulated genes

Gibbs sampling was first used for motif discovery by Lawrence, *et al.* [98, 161]. Like the EM approach described in [99], Lawrence, *et al.* describe a Gibbs sampling algorithm to discover protein motifs, although their algorithm is equally applicable to the discovery of DNA motifs. Like Lawrence and Reilly's EM-based algorithm [99] and MEME [10], Lawrence, *et al.*'s algorithm iteratively updates a motif model and a background model. Indeed, it is noted that the algorithm can be seen as a stochastic analog of the EM procedure used in deterministic algorithms [98]. The Gibbs sampling algorithm also evolves a separate data structure which contains the start positions for the best motif alignment within the dataset (the algorithm uses the OOPS sequence model). Given a motif width W to search for, the algorithm randomly chooses potential motif starting points a within each input sequence and uses these as initial values. Having chosen these initial starting points, the algorithm iteratively carries out two steps. In the first step, a single input sequence z is chosen, either randomly or by cycling through each sequence in some order. The parameters of the motif and background models are calculated based on the current values of a for all input sequences except z. The second step of the algorithm considers all width-W subsequences within z to be a possible instance of the motif and calculates the probability of each subsequence given both the current motif and background models. A new subsequence is then chosen from z using a weighted probability and the corresponding a updated. This procedure is carried out iteratively to discover a motif. The main idea is that the more accurate the motif model constructed in the first step, the more accurate the motif location calculated in the second step will be, and vice versa. The basic algorithm is described in pseudocode in Algorithm 2.3.

```
procedure Gibbs sampling
  Randomly choose initial motif starting points a
  until a motif is found do
    Step 1: choose sequence z at random, update θ_m and θ_b from all other a
    Step 2: calculate probability of all other width-W subsequences in z
        and choose new a with weighted probability
  end
  Print N discovered motifs
end Gibbs sampling
```

Algorithm 2.3: Pseudocode describing Lawrence, *et al.*'s Gibbs sampling algorithm for motif discovery.

The reliance on a random 'jump' means that stochastic methods (including Gibbs sampling) do not converge steadily and predictably as deterministic methods do. This presents a problem in that it is generally difficult to tell how many iterations should be carried out. While Gibbs sampling has no sufficient convergence criteria, necessary convergence criteria do exist, for example, the Gelman-Rubin potential scale reduction factor (PSRF) [69]. Gelman and Rubin's approach to diagnosing MCMC convergence involves obtaining several parallel Markov chains and comparing variances, between chains and within each chain. The PSRF represents a factor of difference between these variances; as the PSRF decreases towards 1, the chains are more likely to have converged to the same target distribution. In their Gibbs sampling-based motif discovery algorithm, Lawrence, *et al.* propose a similar heuristic approach based on the recurrence of the same result using different initial conditions [98]. Like the extended versions of MEME, Lawrence, *et al.*'s algorithm attempts to automatically determine the best motif width by comparing a number of different models over different motif widths. As noted by Bailey and Elkan [12], this cannot be done by simple comparison; Lawrence, *et al.* use a similar comparison method based on the number of model free parameters, a quantity known as the 'information per parameter', which allows models over different motif widths to be compared.

Like LOGOS [181], Lawrence, *et al.*'s algorithm allows for simultaneous discovery of multiple motifs, although the details of how this is done are not presented. Again, it is argued that simultaneous discovery of multiple motifs is preferable to sequential discovery as simultaneous discovery allows information about one motif to aid the discovery of other motifs. The main disadvantage of Lawrence, *et al.*'s algorithm is

that during the optimisation process, it can often get locked into a local maximum that is a 'shifted'[10] version of the motif (for example, if the motif is the 8 bases starting at position a, the algorithm can often return the 8 bases starting at position a-1 or a+1). Lawrence, *et al.* suggest that this situation can be avoided by inserting a 'shifting' step every M iterations, comparing the likelihood of the current motif model with the likelihood of the shifted motif model and switching to one of the shifted patterns if it has a higher likelihood. Lawrence, *et al.* successfully apply their algorithm to a number of proteins including lipocalins and prenyltransferases [98]. It has also been used in studies including Petersen, *et al.* [132] and has formed the basis of other similar motif discovery algorithms.

AlignACE (Aligns Nucleic Acid Conserved Elements, [143]) is based on Lawrence, *et al.*'s Gibbs sampling algorithm but adds a number of extensions to the basic algorithm. Perhaps most interestingly, simultaneous searching for multiple motifs is replaced by a MEME-like approach where the best motif is found and then subsequently masked [143, 174]. Roth, *et al.* claim that such an approach allows for a more efficient search for subtle motifs [143]. A maximum *a priori* log likelihood score is used to judge motifs discovered during the run of the algorithm; this score is used to gain a measure for how overrepresented the motif is within the input sequences. In a similar way to the discriminative algorithms discussed above, AlignACE provides a measure which takes into account the whole genome sequence from which the input sequences were taken. Again, any motifs returned should be overrepresented in the input sequences but relatively much less common in the genes which make up the rest of the genome [168]. Finally, AlignACE considers both strands of DNA, not just the single strand given as input to the algorithm. This means that when potential transcription factor binding sites are examined, either the site or its reverse complement (but not both) are added to the current alignment [143]. Roth, *et al.* [143] demonstrate AlignACE by applying to three extensively studied regulatory systems in *S. cerevisiae*. Hughes, *et al.* [82] also used AlignACE to analyse groups of genes in the *S. cerevisiae* genome; motifs found in this experiment were later confirmed by laboratory experiments. In addition to these tests, AlignACE has been used in other experiments including a search for motifs in *E. coli* by Grainger, *et al.* [71] and in *S. cerevisiae* and *E. coli* in two studies by Wade, *et al.* [169, 170]. It is also available as a software

[10] Lawrence, *et al.* use the term 'phase shifted' to describe such motifs. Following Bailey and Elkan [10], the term 'shifted' will be used here, as it does not carry any notions of periodicity.

download[11].

Like AlignACE, MotifSampler [166] is based on the basic Gibbs sampling algorithm presented by Lawrence, *et al.* but makes two modifications in order to improve results. Unlike the previously discussed motif discovery algorithms, MotifSampler does not use a single nucleotide frequency distribution as the background model. Instead, MotifSampler uses a higher order Markov process to construct a background model which is designed to better represent the data; this model has been shown to improve motif discovery when using a Gibbs sampling approach [165]. While not used in other motif discovery algorithms, Thijs, *et al.* [166] note that this method is used in contemporary gene detection algorithms (including GLIMMER [44], HMMgene [93] and GeneMark.hmm [110]). The background model is usually created using all non-coding sequences from the full genome of the species being studied. Although this clearly requires the full genome data to be available before the algorithm can be applied, Thijs, *et al.* claim that the background model can be created from only the input dataset if required (based on the assumption that the dataset contains only non-coding sequences). In addition to improving motif discovery, use of the higher background model means that the algorithm runs faster, as a result of the background model remaining constant throughout the algorithm. This is in contrast to other motif discovery algorithms, which simultaneously iteratively improve the background model and the motif model. The second modification to the original Gibbs sampling approach is the introduction of a probability distribution to estimate the number of copies of a motif in a sequence. This estimation is represented by adding another latent parameter, which is estimated along with the motif model. By application of Bayes' theorem, the expected number of copies of the motif in each input sequence can be calculated. Although Thijs, *et al.* do not explicitly mention particular sequence models, the addition of this parameter implies the use of the TCM sequence model (see Section 2.2.1). The MotifSampler algorithm is described in Algorithm 2.4.

Thijs, *et al.* demonstrate MotifSampler on two main datasets, using plant and bacterial data [166]. The bacterial dataset was constructed using FNR regulated genes from six different species, including *Rhodobacter* and *Sinorhizobium* species. MotifSampler was shown to successfully detect the FNR binding site with high probability. Besides these tests, MotifSampler has been used by Le Crom, *et al.* to discover motif sequences in yeast [100]. It has also been made available as a web service[12].

[11] http://arep.med.harvard.edu/mrnadata/mrnasoft.html

[12] http://bioinformatics.psb.ugent.be/webtools/MotifSuite/motifsampler.php

procedure MotifSampler
  Compute *m*th-order background model θ_b from full genome data
  Randomly choose initial motif starting points *a*
  for *N* **do** *(where N is the number of motifs to be found)*
   until a motif is found **do**
    Step 1: choose sequence *z* at random, update motif model θ_m from all other *a*
    Step 2: calculate probability of all other width-*W* subsequences in *z*
      and choose new *a* with weighted probability
   end
   Print discovered motif
   Mask discovered motif
  end
end MotifSampler

Algorithm 2.4: Pseudocode describing the MotifSampler algorithm

Stochastic variants of deterministic methods

Monte Carlo methods have also been incorporated into deterministic algorithms, as an approximation to the E-step of the EM algorithm, in a procedure known as the Monte Carlo EM (MCEM) algorithm [173]. This algorithm takes advantage of the fact that the expectation of a random variable (2.3) can be approximated as a finite sum over samples from its probability distribution function, that is:

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x f(x) \mathrm{d}x \approx \frac{1}{L} \sum_{l=1}^{L} X^{(l)}. \tag{2.12}$$

This approximation becomes exact in the limit $L \to \infty$. In the MCEM algorithm, the integral to be calculated in the E-step is replaced with a finite sum over a number of samples from the posterior distribution [173, 87]:

$$Q(\theta|\theta^{(t)}) \approx \frac{1}{L} \sum_{l=1}^{L} \ln p(\mathbf{X}, \mathbf{Z}^{(l)}|\theta). \tag{2.13}$$

This sampling step (or S-step) replaces the E-step in the algorithm. The M-step remains the same, but is often known as the update, or U-step. A particular instance of the MCEM algorithm known as 'stochastic EM' draws just one sample in each E-step [27]. This can be viewed as a 'hard' assignment of data points to either the motif or background model. Celeux, *et al.* note that stochastic variations on the deterministic EM algorithm are generally more successful for two main reasons. Firstly, the

stochastic perturbations resulting from the sampling step of these variations guide the algorithm away from stable fixed points of the EM algorithm, such as saddle points and insignificant local maxima of the likelihood function. This is achieved by allowing a non-zero probability of accepting updated parameters with a lower likelihood than the current parameters at each EM iteration. Secondly, the 'underlying EM dynamics' resulting from the deterministic U-step mean that the algorithms generally converge in a relatively small number of iterations, in comparison to full stochastic methods [37].

Stochastic EM has been utilised for motif discovery in the SEAM (Stochastic EM-type Algorithm for Motif-finding) algorithm [23]. Here, stochastic EM is used in an attempt to overcome the limitations inherent in the deterministic EM algorithm, primarily the problem of converging to a local maximum of the likelihood function. One clear advantage of using stochastic EM is that the basic multinomial mixture model used in deterministic EM can be retained, while the stochastic sampling step allows the algorithm to escape local optima of the likelihood function and discover motifs which are statistically more significant. Notably, SEAM also modifies the deterministic update step of the sEM algorithm to either accept or reject the samples drawn in the S-step, using the Metropolis algorithm to test how good the new samples are. The use of the Metropolis algorithm in the U-step of the SEAM algorithm means that the U-step is no longer deterministic (as in the original sEM algorithm). Therefore, while sEM as it is originally defined may converge faster than fully stochastic approaches (for instance, Gibbs sampling), SEAM cannot be guaranteed to have this property.

SEAM only implements the OOPS sequence model; although other sequence models are discussed, they are not implemented. Bi demonstrates the performance of SEAM using two yeast datasets and three *E. coli* datasets, including the 'gold standard' CRP dataset [23]; like Lawrence and Reilly's EM algorithm [99], although SEAM was designed to discover protein motifs, it is equally applicable to the discovery of DNA motifs. SEAM is shown to improve performance over a deterministic EM-based algorithm in searching for a global optimum in the likelihood function [23]. SEAM is not available as a software download or as a web service and has not been used experimentally besides the tests carried out by Bi.

MCEMDA (Monte Carlo EM Motif Discovery Algorithm) [24] takes a similar approach to SEAM, but implements Monte Carlo EM rather than stochastic EM, drawing three samples from each sequence in the E-step instead of one.[13] Like SEAM,

[13] Although three samples are drawn at each iteration, only the best is used. This is not quite true to the original spirit of MCEM, which would take an average: this approach is noted, but ultimately not implemented as it was thought to be too inefficient.

procedure SEAM
  Randomly choose initial motif starting points $z^{(t=0)}$
  Calculate initial model θ and value of entropy function $G(\theta)$
  until a stationary distribution (equilibrium) **do**
    S-step:
    for each input sequence **do**
      Compute conditional likelihood for each position, sample a position
    end
    Form a new sample z', compute θ' and $G(\theta')$
    U-step:
    Draw $u \sim \text{Unif}[0,1]$
    Update $z^{(t+1)} = z'$ if $u \leq \exp(-\Delta G)$
  end
  Return optimal alignment z^* and associated motif model θ^*
end SEAM

Algorithm 2.5: Pseudocode describing the SEAM algorithm

MCEMDA only implements the OOPS sequence model. Perhaps the most notable point discussed is how to choose the number of samples to be taken at each iteration. It is noted that if only one sample is taken, this reduces to the stochastic EM algorithm, as implemented in SEAM. If the number of samples is increased, the algorithm behaviour becomes increasingly deterministic (as the number of samples increases, the algorithm tends to behave like the EM algorithm). However, it is difficult to determine a theoretical value for the number of samples which should be taken, as this varies according to the dataset used. MCEMDA is tested on a small number of motifs, including three *E. coli* motifs from RegulonDB. However, it is unclear what advantage MCEMDA offers over SEAM. Again, MCEMDA is not available as a software download or as a web service and has not been used experimentally besides the tests carried out by Bi [24].

Algorithms capable of detecting gapped motifs

While all of the previously discussed algorithms have been successful to some extent in detecting ungapped motifs, relatively few attempts have been made to design algorithms capable of detecting gapped motifs. Bi notes that most motif finding methods assume a contiguous motif and thus do not explore the properties of a discontinuous structured motif [26] The main reason for this is that motif discovery becomes much

harder when gaps are introduced, due to an 'explosion' in the number of possible variations [63]. In their review of motif discovery algorithms, Wei and Yu [174] note only two probabilistic algorithms with this capability: BioProspector [107] and Co-Bind [76]. However, Frith, *et al.* note a number of non-probabilistic algorithms such as PRATT [89], SAM [83] and HMMER [55]. BioProspector and Co-Bind are discussed below, as well as later algorithm GLAM2 [63], which was published after Wei and Yu's review.

BioProspector [107] is similar to MotifSampler in that it also extends Lawrence, *et al.*'s Gibbs sampling algorithm by introducing a higher order background model in order to better capture the characteristics of the local DNA environment [107]. BioProspector uses up to third order Markov background models, but also gives the option of using a 0th order background model as used by other algorithms. Like MotifSampler, BioProspector implements the TCM sequence model to allow for the case of zero or multiple motif copies per sequence, giving more flexibility than Lawrence, *et al.*'s algorithm. Liu, *et al.* claim that both of these additions greatly improve the performance of the algorithm [107]. Perhaps more importantly, BioProspector also introduces a strategy for searching for gapped motifs; rather than a single PWM representing the motif model θ_m, BioProspector generally uses two PWMs (that is, θ_{m1} and θ_{m2}). The first PWM is initialised by randomly choosing a starting point within each sequence (as in Lawrence *et al.*'s algorithm); the second is then initialised using the alignment position a fixed gap away from the first. Where the gapped motif is also palindromic, only one PWM need be used. Wei and Yu note that this feature is important, as gapped motifs are prevalent in prokaryotes [174]. Liu, *et al.* successfully use BioProspector to find motifs in three real-world datasets: *S. cerevisiae* (RAP1 protein), *B. subtilis* ('TATA box' gapped motif) and *E. coli* (CRP protein). BioProspector was particularly successful in discovering the gapped TATA box motif, despite low conservation in the data, using the two PWM strategy described above. This strategy was also applied to the CRP data. While algorithms analysing this data in the past had used a single PWM model with reasonable results, the two PWM model used by BioProspector was shown to greatly improve results [107]. Besides these tests, BioProspector has been used by Mukherjee, *et al.* [127] to discover motif binding sites in yeast and is also available as a software download[14].

Co-Bind (Cooperative Binding, [76]) is specifically designed to model gapped motifs, or cooperatively acting transcription factors within close proximity to each other.

[14] http://ai.stanford.edu/~xsliu/BioProspector/

Like BioProspector, Co-Bind uses a two PWM approach to model both parts of the gapped motif and maximises the joint likelihood of both PWMs to discover gapped motifs. GuhaThakurta and Stormo note that such an approach is particularly useful in the case where two halves of a gapped motif are insignificant (either in terms of probability or information content) individually, but significant when taken together. That is, Co-Bind is able to identify both halves of gapped motifs which would not have been found if searched for individually by conventional motif discovery algorithms. Like the previously described Gibbs sampling algorithms, given some random initialisation, the two PWMs are iteratively improved to detect the motif. As noted above, due to the random nature of Monte Carlo strategies, it is very hard to decide on convergence criteria for algorithms based on Gibbs sampling. GuhaThakurta and Stormo solve this problem by simply running Co-Bind for a fixed number of iterations [76]. It is assumed that the number of iterations is large enough that the algorithm will be at or near convergence after they have been carried out. While this greatly simplifies the problem, it is clear that the number of iterations must be chosen carefully. Too few iterations would mean that the algorithm is far from convergence by the limit (Bishop [27] and Gamerman and Lopes [66] demonstrate that this is entirely possible, dependent on the model and the initial parameter values), while too many iterations would mean many wasted cycles after convergence has been reached. GuhaThakurta and Stormo demonstrate Co-Bind on semi-synthetic and real data extracted from the SCPD database; four sets of yeast genes shown experimentally to be regulated by two factors were constructed. In both tests, Co-Bind was shown to improve on the performance of BioProspector (the only comparable algorithm) in the identification of motifs with small gaps. Co-Bind was also shown to discover weak-signal motifs which could not be found using other motif discovery algorithms. Co-Bind has been used experimentally by Pramila, *et al.* [136] to identify motifs in yeast, which were later confirmed experimentally [75]. It is also available as a software download[15].

GLAM2 (Gapped Local Alignment of Motifs, [63]) is a generalisation of Lawrence, *et al.*'s original Gibbs sampling algorithm as described above. However, GLAM2 uses a technique known as simulated annealing to optimise the initial estimations for the model parameters. As noted above, the original Gibbs sampling algorithm had a tendency to return shifted motifs as a consequence of the random optimisation procedure [98]. GLAM2 attempts to avoid this by using simulated annealing, which generally increases the likelihood of the motif model but sometimes decreases it in order to es-

[15] http://stormo.wustl.edu/software.html

cape from local maxima in the likelihood function. GLAM2 is novel in that as well as allowing for longer gaps in motifs, it also allows for arbitrary insertions and deletions of single bases within motifs. Allowing for such natural mutations is important as they can negatively impact the result of a motif discovery algorithm. Frith, *et al.* demonstrate GLAM2 by searching for previously discovered protein kinase substrate motifs within the PROSITE protein database [150] and a gapped DNA motif in the mouse genome [63]. GLAM2 is available as a software download[16] and since 2009, GLAM2 has been used as part of the online MEME Suite [8], allowing the discovery of gapped motifs.

Stochastic algorithms using coregulated genes and phylogenetic footprinting

Like the deterministic algorithms discussed in Section 2.2.2, stochastic algorithms which can take advantage of phylogenetic footprinting can be effective if the relevant expert knowledge is available. PhyloGibbs [149] is a Gibbs sampling algorithm that takes phylogenetic relationships between species into account, in the same way as the deterministic algorithms PhyME and EMnEM. In many ways, PhyloGibbs can be seen as a stochastic analogue of PhyME, as both algorithms use the same evolutionary model for the evolution of binding sites. PhyloGibbs and PhyME are also similar in that the input dataset need not be already aligned (as noted earlier, EMnEM requires a global multiple alignment as input; Siddharthan, *et al.* claim that, as well as being inflexible, such an approach can adversely affect the performance of a motif discovery algorithm [149]). Clearly, the main difference between these algorithms is that PhyloGibbs uses a Gibbs sampling approach rather than an EM-like approach. This allows the simultaneous discovery of multiple motifs, in the same way as other Gibbs sampling-based algorithms. Like GLAM2, PhyloGibbs uses simulated annealing in an attempt to find a global optimum in the likelihood function. Siddharthan, *et al.* test PhyloGibbs on synthetic data and real data from five *Saccharomyces* species; PhyloGibbs is shown to outperform PhyME, EMnEM and benchmark non-phylogenetic algorithm MEME. Besides these tests, PhyloGibbs has been used experimentally by Galgano, *et al.* [64] to discovery motif sequences in human genes and Ferreira, *et al.* [62] to discover motif sequences in tomato genes. It also available as a software download[17].

[16] http://acb.qfab.org/acb/glam2/

[17] http://www.phylogibbs.unibas.ch/cgi-bin/phylogibbs.pl

Discussion of stochastic methods

Like the deterministic motif discovery algorithms discussed in Section 2.2.2, stochastic motif discovery algorithms have some important differences despite their apparent similarities. Lawrence, *et al.*'s basic Gibbs sampling method [98] and AlignACE [143] both implement the OOPS sequence model. As noted above, this limits the performance of the algorithm when input sequences do not contain a motif. Later algorithms MotifSampler and BioProspector implement the TCM sequence model, which improves the flexibility of the algorithm in such situations and allows an estimate for the distribution of motifs within the dataset to be made.

MotifSampler [166] and BioProspector [107] are also novel in that they implement a higher order (usually 3rd order) Markov background model rather than the single nucleotide frequency distribution models used by other motif discovery algorithms. Although there are some drawbacks to using a Markov background model (most notably that the full genome data must be available in order to create such a model), doing so improves the performance of an algorithm in motif discovery. Algorithmic efficiency is also improved as the background model need only be calculated once then reused, rather than be initially estimated and then updated at each iteration of the algorithm [166]. This is particularly important in stochastic algorithms, where in general, many more iterations are required in comparison to deterministic algorithms. The use of higher order Markov background models is discussed further in Section 6.3.

While all of the stochastic algorithms discussed above are capable of searching for multiple motifs, there are differences in their methods for doing so. The original Gibbs sampling method introduced by Lawrence, *et al.* [98] makes use of the fact that stochastic algorithms allow for simultaneous discovery of multiple motifs, that is, several PWMs are initialised and iteratively updated to discover a number of different motifs. However, later algorithms based on Lawrence, *et al.*'s approach use a MEME-like method, repeatedly discovering a single motif and then probabilistically masking or erasing it. While Lawrence, *et al.* [98] claim that simultaneous discovery of multiple motifs allows knowledge of one motif to inform discovery of other motifs, it is subsequently claimed by Roth, *et al.* that sequential discovery allows for more efficient motif discovery, particularly with regard to subtle motifs [143].

Recent studies have suggested that gapped motifs are reasonably common, especially in prokaryotes [174, 63]. This agrees with analysis of the dataset of characterised *E. coli* motifs used in this study (Section 3.2); of the 20 motifs within this dataset, 8

were judged to be gapped, based on observation of the information content profile of the motif. The ability to discover gapped as well as ungapped motifs is therefore desirable. The novel two-PWM approach introduced by BioProspector [107] appears to work well and has also been implemented in Co-Bind [76] and GLAM2 [63]. However, it is still limited in that the gap between the two PWMs is fixed. Removing this constraint would allow more flexible discovery of gapped motifs; it is possible that this could be implemented in a similar way to the automatic determination of motif width procedure in other motif discovery algorithms.

Published evaluations of stochastic motif discovery algorithms

The evaluation of motif discovery algorithms carried out by Tompa, *et al.* [168] included AlignACE [143] and MotifSampler [166] (as well as GLAM2's predecessor GLAM). In tests using TRANSFAC data, AlignACE was found to be comparable to MEME [10] in almost every aspect other than sensitivity and nucleotide-level correlation in the mouse and yeast datasets, where MEME performed slightly better. MotifSampler outperformed MEME in nucleotide-level correlation in the yeast dataset, gaining the second best score (after Weeder). However, MotifSampler performed slightly worse than MEME and AlignACE in terms of site-level precision. No explanation is given for this, however it is noted that in contrast to the other evaluated algorithms, MotifSampler's performance increased when using real data rather than synthetic data. It is possible that this is due to the introduction of a higher order Markov background model, which may work better on the background sequences in real data rather than the randomly generated background in synthetic data. GLAM was shown to be worse than almost all other algorithms, although Tompa, *et al.* point out that it is possible that GLAM returned more potential motifs than other algorithms, returning more weak motifs and decreasing its relative performance. As noted above, the datasets used by Tompa, *et al.* were restricted to eukaryotic data. Hu, *et al.* [81] carry out tests using prokaryotic data on AlignACE, MotifSampler and BioProspector. Like Tompa, *et al.*, Hu, *et al.* show that AlignACE is comparable to MEME and that MotifSampler and BioProspector perform slightly better in terms of motif discovery. It is possible that this is a result of the higher order Markov background model. Hu, *et al.* also compare the performance of the algorithms on datasets of different sizes to compare scalability. BioProspector and MEME are shown to be the best probabilistic algorithms when input sequence length increases; however, Hu, *et al.* note that in general, Gibbs sampling strategies tend to become inefficient with increasing sequence length. Given that Bio-

Prospector is also a Gibbs sampling-based algorithm, this conclusion may need further investigation.

Model representation

In general, the stochastic algorithms discussed represent the motif and background models in the same way as the deterministic algorithms discussed in Section 2.2.2. As noted, the simplicity and readability of PWMs make them an ideal representation of both models. The stochastic algorithms capable of discovering gapped motifs introduce a variation on this representation, using two PWMs to represent a gapped motif (each PWM represents one of the contact points of the dimer). This two-PWM approach, introduced by Liu, *et al.* and implemented in BioProspector [107], has been used successfully in the discovery of gapped motifs. The success of this approach has led other algorithms to incorporate similar approaches. As noted above, current implementations of the two-PWM approach for gapped motif discovery are somewhat limited in that the 'gap' between PWMs is fixed, either as part of the algorithm or as a user-defined parameter. However, in practice, the size of the gap may be variable, although constrained to a narrow range due to the shape of the DNA molecule. It would therefore be desirable for the width of the gap to be automatically determined, in much the same way as MEME and other algorithms automatically determine the width of a motif sequence. Of the algorithms capable of discovering gapped motifs, only BioProspector mentions using a different representation for palindromic motifs. In this case, only one PWM is required. It is assumed here that similar model constraints are used in BioProspector as in MEME and that some heuristic process exists which allows BioProspector to change between model representations as required.

Issues with stochastic methods

Although stochastic methods such as Gibbs sampling remove some of the limitations of deterministic methods such as the EM algorithm, they bring their own limitations. Unlike the EM algorithm, which has been shown to converge in a known and predictable way [180], stochastic methods are unpredictable in their convergence due to a random step [27]. This random step clearly helps to avoid local optima but also makes it hard to determine when convergence has occurred. This is a very difficult problem in general; Robert and Casella [141] summarise convergence diagnostic techniques. Concerns over convergence detection and the time taken to reach convergence led Xing,

et al. to use variational methods over stochastic methods in LOGOS [181]. The time taken to reach convergence is also a valid concern; Bishop notes that the Metropolis-Hastings algorithm (of which Gibbs sampling is a special case) can have very slow convergence rates depending on the distribution [27]. Although more sophisticated Monte Carlo methods (for example, slice sampling [129] or hybrid Monte Carlo) have been introduced in an attempt to increase the rate of convergence, the nature of the random processes which make up these methods means that it is hard to choose one technique which is consistent and predictable in its convergence in the same way that deterministic methods are.

2.3 Comparing probabilistic approaches to motif discovery

It can be seen from the above discussions that each of the three algorithmic techniques have both advantages and disadvantages. Although used in LOGOS [181], variational inference is ruled out here for a number of reasons. Firstly, stochastic methods can more closely approximate the true distribution given enough time (based on the law of large numbers), in comparison with variational inference, which will always remain approximate. How close this approximation is to the true distribution depends heavily on the chosen approximating distribution (recall that this must be chosen from the set of tractable distributions). The choice of the approximating distribution is usually based on some assumption of how the complex posterior distribution factorises; given this, variational methods are deemed to lack generality for the situation proposed. If prior knowledge is to be incorporated into the model, some assumption must also be made about the form of this knowledge. Replacing some form of prior knowledge with another form of prior knowledge means that the original assumptions regarding the factorisation must be updated and this may involve changing the approximating distribution, possibly every time another form of knowledge is incorporated; clearly, one system which can be used regardless of the knowledge to be incorporated is desirable.

Deterministic and stochastic motif discovery algorithms generally have complementary advantages and disadvantages. Deterministic algorithms have the advantages that they are relatively simple and are well understood, converging to a solution in a steady, well-defined way. This convergence is usually reasonably fast and always converges to the same model, given the same input data. However, the converged model

may be a local optimum of the likelihood function and therefore the returned motif may not be the most significant in the dataset. Deterministic algorithms are also limited in that it is hard to incorporate prior knowledge in anything but the simplest cases. They also only admit sequential discovery of multiple motifs, although as discussed, this may be a better strategy than simultaneous motif discovery. Stochastic algorithms are also relatively simple (although perhaps slightly more complex than deterministic algorithms) and perhaps have the most important advantage that they are able to escape local optima in order to return the most significant motif in the dataset. The stochastic methods discussed also have the advantage that they are much more flexible in incorporating prior knowledge; doing so has been shown to improve results. Unlike deterministic algorithms, stochastic algorithms allow either sequential or simultaneous motif discovery and are the only current algorithms that can handle gapped motifs (although a similar deterministic implementation of the method for discovering gapped motifs may be possible). The main potential disadvantages of stochastic algorithms stem from the random processes at the heart of the method. Most importantly, convergence of stochastic algorithms is not well defined and generally much slower than convergence of deterministic algorithms; is is entirely possible that stochastic algorithms spend an unpredictable number of iterations jumping around the search space, effectively leaving the algorithm on a plateau before converging. It should also be noted that unlike deterministic algorithms, stochastic algorithms generally give slightly different results given the same input data and additional parameters for different random seeds. This means that such algorithms usually have to be run multiple times to achieve a consensus of results.

Stochastic EM appears to be a promising approach for motif discovery. Unlike deterministic EM, the sampling step in stochastic EM allows the algorithm to escape local maxima of the likelihood function and return improved results. Given that it is expected that the likelihood function in the motif discovery problem is generally complex and multimodal, this would appear to be advantageous.

The 'ideal' motif discovery algorithm

Given the above discussion, it is possible to construct a list of desirable features that would be included in the 'ideal' motif discovery algorithm.

- **Implementation of the TCM sequence model** - Implementation of this model would remove the requirement that each input sequence contain at most one

motif occurrence, which cannot be guaranteed when analysing uncharacterised data.

- **Incorporation of prior knowledge** - A framework for easily incorporating various forms of prior knowledge would be most beneficial; ideally, this would allow prior knowledge to be simply 'slotted in'. The HMDM model as used to encode knowledge of motif-level site dependencies in LOGOS could be viewed as a form of prior knowledge. Other forms of knowledge may be possible.

- **Ability to discover multiple motifs** - Again, this makes sense as it is assumed the dataset contains multiple motifs. Ideally, multiple motifs will be returned in order of significance. Consensus seems to suggest that sequential discovery is the best strategy for discovering multiple motifs; it can be implemented in either deterministic or stochastic algorithms and seems to have an advantage over simultaneous discovery of multiple motifs.

- **Automatic determination of motif width** - Implementation of this would eliminate one of the user parameters. This will require some model comparison method, such as that used in MEME.

- **Control of the false discovery rate** - In evaluation, discriminative algorithms and 'cautiously run' algorithms such as Weeder show the effect of controlling the number of false positives, either by comparison with random sequences or by more sophisticated methods.

- **Implementation of a higher order background model** - Using a higher order Markov background model has been shown to improve the results of algorithms such as MotifSampler and BioProspector. By constructing the background model only once, this is also likely to improve sequential motif discovery and algorithmic efficiency.

- **Ability to discover gapped motifs** - If it is assumed that gapped motifs are prevalent, it makes sense to implement a method for their discovery. This may require a stochastic approach. It would also be desirable to implement a heuristic that automatically switches between gapped and non-gapped motifs.

Chapter 3

Data

This chapter describes the datasets created for the evaluation of the motif discovery algorithms, beginning with the construction of large data collections of DNA sequences containing motifs at varying conservations (Section 3.1). This is followed by a description of the *E. coli* data collection; these datasets contain previously characterised motif sequences from the RegulonDB database (Section 3.2). An additional collection of prokaryotic datasets created from ChIP data is then described in Section 3.3. Section 3.4 presents a collection of *E. coli* intergenic sequence data; this data is used in order to create more realistic synthetic datasets and is used in the evaluation of a higher order Markov background model in Section 6.3. Section 3.5 gives a brief description of the Alphaproteobacteria class of bacteria and provides a summary of the Alphaproteobacteria data used in the study. Particular attention is paid to the operon structure of this data, whose motifs have not been previously characterised. Finally, additional datasets are described in Section 3.6.

3.1 Realistic synthetic data

Five realistic synthetic data collections, each consisting of 1,000 datasets, were created in order to test the MCOIN heuristic presented in Section 4.4 and the MITSU algorithm presented in Chapter 5. Each dataset contained 20 input sequences of length 200 nucleotides (nt). Input sequences were created by extracting 200nt from the EcoGene [144] database of *E. coli* intergenic sequences, representing 'background' positions. Datasets were created so that each data collection had different mean levels of motif conservation, ranging from 0.51 to 2.00 bits/col: Motif positions within each sequence were chosen at random and a synthetic motif inserted. Synthetic motifs were cre-

ated by choosing nucleotides (A, C, G, T) at random and randomly mutating positions in the motif occurrences so that the levels of conservation at each position could be controlled. This 'point mutation' method of constructing datasets allows motif conservation to be varied and has also been used in other studies (for example, by Li, *et al.* [102]). A motif width of 12nt was used in all synthetic datasets. A comparison of methods for determining motif width in Bembom, *et al.* used datasets containing real (human) motifs with a minimum mean information content of 0.76 bits/col [21]; the realistic synthetic data used in this study contains many motifs at lower levels of motif conservation, as analysis of known *E. coli* TFBS motifs indicated that significant numbers of motifs had mean conservation levels of less than 0.76 bits/col.

3.2 *E. coli* datasets

Twenty datasets incorporating known *E. coli* TFBS sequences were created (Table 3.1). Background sequences were created as for realistic synthetic data. Positions within each 200nt input sequence were chosen at random and a known TFBS sequence inserted. Known *E. coli* TFBS sequences were extracted from RegulonDB [65] for insertion in the background positions. One notable concern was that, if the sole evidence for the TFBS sequences was computational prediction, this may introduce some circularity to the predictions made in this study. That is, the predictions made in this study may simply reproduce results which were previously computationally predicted. However, TFBS data from RegulonDB is supported by literature with experimental evidence; in the majority of cases this evidence stems from classical experimentation such as DNA footprinting and/or site mutation expression analysis and is not supported solely by human or computational inference[1].

Each motif occurrence is embedded in one background sequence, hence the number of motif occurrences in RegulonDB defined the number of input sequences. The mean number of input sequences was 15, ranging from 2 to 99 input sequences (median: 9). Using known motif occurrences in this manner preserves the true biological conservation of the motif. The mean motif conservation was 1.13 bits/col, ranging from 0.49 to 2.00 bits/col (median: 1.04 bits/col). The mean motif width was 16nt, ranging from 10 to 21nt (median: 17nt).

The data collection was split into two groups based on mean information content

[1]The experimental evidence for the TFBS sequences extracted from RegulonDB is provided as supplementary material in [91].

High conservation			Low conservation		
Name	W^*	N	Name	W^*	N
Ada	13	4	ArgR	18	35
CaiF	16	8	DeoR	16	7
CueR	19	3	FruR	18	18
IlvY	21	4	Fur	19	99
LacI	21	3	GntR	20	17
MalI	12	2	MalT	10	20
MelR	18	11	Nac	15	18
MetR	13	7	RcsB	14	11
PurR	16	20			
SoxR	19	2			
TorR	10	10			
XylR	18	4			

Table 3.1: Summary of characterised *E. coli* TFBS motifs used in tests with real data. For each motif, transcription factor name is given, along with known width (W^*, nt) and number of motif occurrences (N).

per column. The split was made at a value of 1 bit/col, producing a 'high conservation' group containing 12 datasets and a 'low conservation' group containing 8 datasets. In the 'high conservation' group, the mean number of input sequences was 7, ranging from 2 to 20 input sequences. The median number of input sequences was 4. The mean motif conservation was 1.36 bits/col, ranging from 1.02 to 2.00 bits/col (median: 1.31 bits/col). The mean motif width was 16nt, ranging from 10 to 21nt (median: 17nt). In the 'low conservation' group, the mean number of input sequences was 28, ranging from 7 to 99 input sequences (median: 18). The mean motif conservation was 0.78 bits/col, ranging from 0.49 to 0.99 bits/col (median: 0.79 bits/col). The mean motif width was 16nt, ranging from 10 to 20nt (median: 17nt). Table 3.1 illustrates some of the diversity within the chosen *E. coli* motifs. The sequence logos of selected motifs (Figure 3.1) illustrate the diversity in terms of motif conservation.

Eisen has noted that "transcription factors rarely contact a single base without interacting with adjacent bases" [56]. It follows that there should be some correlation between positions of high information content within a motif. That is, motif positions with a high information content will frequently be adjacent to positions which also have

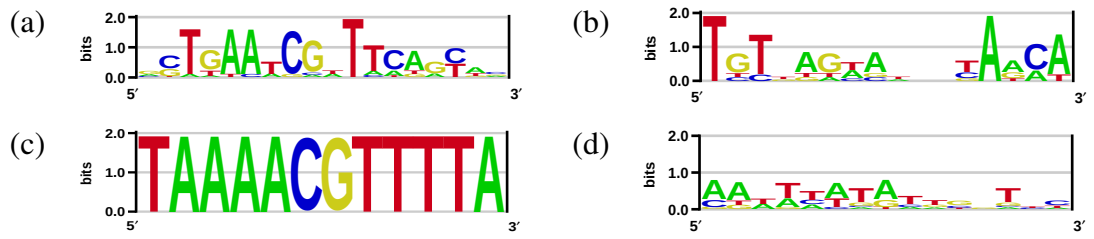

Figure 3.1: Diversity of *E. coli* motifs: Sequence logos for four *E. coli* motifs illustrate the diversity of motifs in terms of information content profile. (a) FruR has a number of perfectly conserved positions in the centre of the motif, flanked by positions which are less well-conserved. (b) The gapped motif of DeoR illustrates the opposite: two well-conserved segments are separated by an unconserved 'gap'. (c) All positions in the MalI motif are perfectly conserved. (d) The Nac motif has few well-conserved positions.

a high information content and similarly, positions with a low information content will frequently be neighboured by positions which also have a low information content. This correlation in positional information content has been shown in eukaryotic motifs from the TRANSFAC database [56]. Figure 3.2 confirms that this correlation is also present in the characterised motifs within the *E. coli* datasets (Pearson product-moment correlation coefficient: $r = 0.55$, $p < 2.20 \times 10^{-16}$). As the information content of a given motif position increases, the mean information content of its neighbouring positions also increases. The importance of this clustering for motif discovery will be noted in Chapters 4 and 5.

3.3 Diverse prokaryotic datasets from ChIP data

In order to create a more diverse database from current genome-wide data, additional datasets containing previously characterised motifs from a range of prokaryotic species were constructed. One particular motivation for including these additional datasets is that they ensure that results are not unique to *E. coli* motifs in particular and that tested methods are widely applicable to prokaryotic TFBS motifs. The motifs in this collection were determined through ChIP methods. As with the *E. coli* motifs described in Section 3.2, the prokaryotic motifs used here are generally supported by experimental evidence such as electrophoretic mobility shift assays (EMSA). It is noted that the use of ChIP methods alone is not a solution to the circularity problem mentioned in Section 3.2, as many studies which have used ChIP methods for motif discovery simply per-

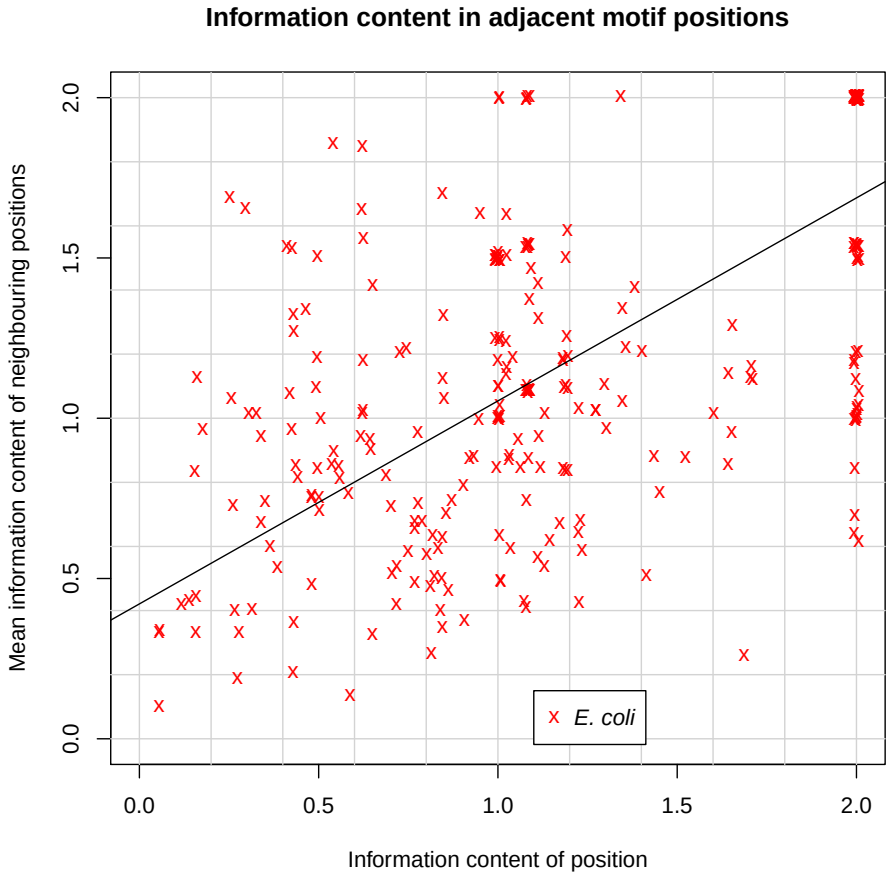

Figure 3.2: Information content for a given motif position vs. the mean information content of the neighbouring positions within known *E coli* motifs. A small amount of jitter has been added in order to help distinguish multiple data points at the same coordinates. The least squares regression line is printed in black.

form computational discovery (generally using MEME or AlignACE) on the (∼500nt) peaks returned by ChIP analysis.

Species	Name	W^*	N
E. coli	CRP	22	34
E. coli	LexA	20	25
E. coli	PurR	16	28
E. coli	RutR	16	19
V. cholerae	Fur	21	55
V. cholerae	RpoN	15	37
M. tuberculosis	DosR	18	24
M. tuberculosis	LexA	18	23
B. subtilis	Spo0A	12	94

Table 3.2: Summary of known prokaryotic TFBS motifs used in tests with real data. For each motif, the species and transcription factor name is given, along with known width (W^*, nt) and number of motif occurrences (N).

Nine datasets incorporating known prokaryotic motifs discovered by ChIP methods were created (Table 3.2). Motifs from diverse species including *E. coli* [72, 170, 39, 147], the Gammaproteobacterium *Vibrio cholerae* [42, 52], the Actinobacterium *Mycobacterium tuberculosis* [111, 156] and the Bacillus *Bacillus subtilis* [123] were used. Background sequences for the *E. coli* datasets were created as for the datasets described in Section 3.2. Background sequences for other species were created by randomly choosing nucleotides, altering the weighting to reflect GC-content as required. Again, positions within each 200nt input sequence were chosen at random and a known TFBS sequence inserted.

As with the *E. coli* datasets, each known motif occurrence is embedded in one background sequence, hence the number of occurrences for each motif defined the number of input sequences; the mean number of input sequences was 38, ranging from 19 to 94 input sequences (median: 28). Again, the use of known motif occurrences allows the true motif conservation to be retained. The mean motif conservation was 0.99 bits/col, ranging from 0.56 to 1.25 bits/col (median: 1.04 bits/col). The mean motif width was 16nt, ranging from 10 to 22nt (median: 16nt). Like the *E. coli* TFBS motifs, the prokaryotic motifs display a great diversity, as demonstrated in Figure 3.3.

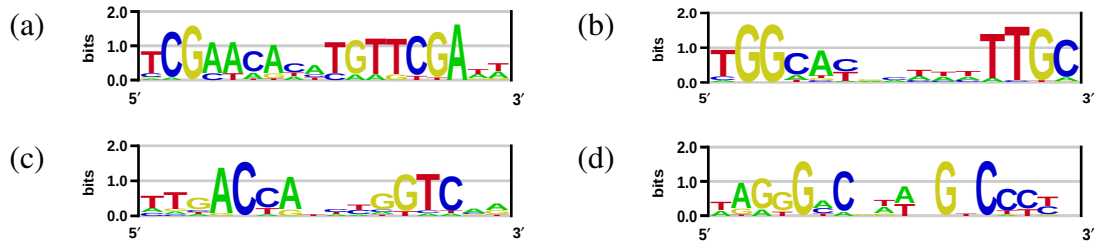

Figure 3.3: Diversity of prokaryotic motifs: Sequence logos for four TFBS motifs illustrate the diversity of motifs in terms of information content profile. (a) The *M. tuberculosis* LexA motif is reasonably well-conserved at all positions. (b) The *V. cholerae* RpoN motif is clearly gapped, with two well-conserved sections separated by a number of unconserved positions. Both the *E. coli* RutR motif (c) and the *M. tuberculosis* DosR motif (d) display high levels of palindromicity.

3.4 Intergenic data

As noted in Section 3.1, background positions in the realistic synthetic datasets were created by extracting *E. coli* intergenic sequences from the EcoGene database [144]. Retaining the actual intergenic sequences creates more realistic test data in comparison to randomly choosing nucleotides with given probabilities. 2,509 intergenic sequences were extracted; although the majority of these sequences were relatively short, there were some longer sequences. The minimum sequence length was 47nt and the maximum length was 960nt; the mean sequence length was 172.98nt. Figure 3.4 presents a histogram of intergenic sequence lengths.

Intergenic sequences account for approximately 9.5% of the *E. coli* genome. Analysis of the intergenic sequences shows that there are some significant differences between the *E. coli* intergenic sequences and the full *E. coli* genome, in terms of nucleotide distribution. For example, the GC-content of the intergenic sequences is calculated to be 40.3%, in comparison to the full genome, which has a GC-content of 50.7%. There are also some noticeable differences in the frequencies of particular di- and trinucleotides. Perhaps unsurprisingly given the differences in GC-content, the frequencies of the AA, TA and TT dinucleotides are increased in the intergenic sequences: $p(X_{i,j} = A|X_{i,j-1} = A)$, $p(X_{i,j} = A|X_{i,j-1} = T)$ and $p(X_{i,j} = T|X_{i,j-1} = T)$ are increased to 0.3553, 0.2461 and 0.3557 from 0.2958, 0.1858 and 0.2975 respectively. Similarly, $p(X_{i,j} = C|X_{i,j-1} = G)$ and $p(X_{i,j} = G|X_{i,j-1} = C)$ are decreased to 0.2537 and 0.2163 from 0.3262 and 0.2939 respectively. The relative frequency of the rarest trinucleotide in the full genome, CTA, increased in the intergenic sequences:

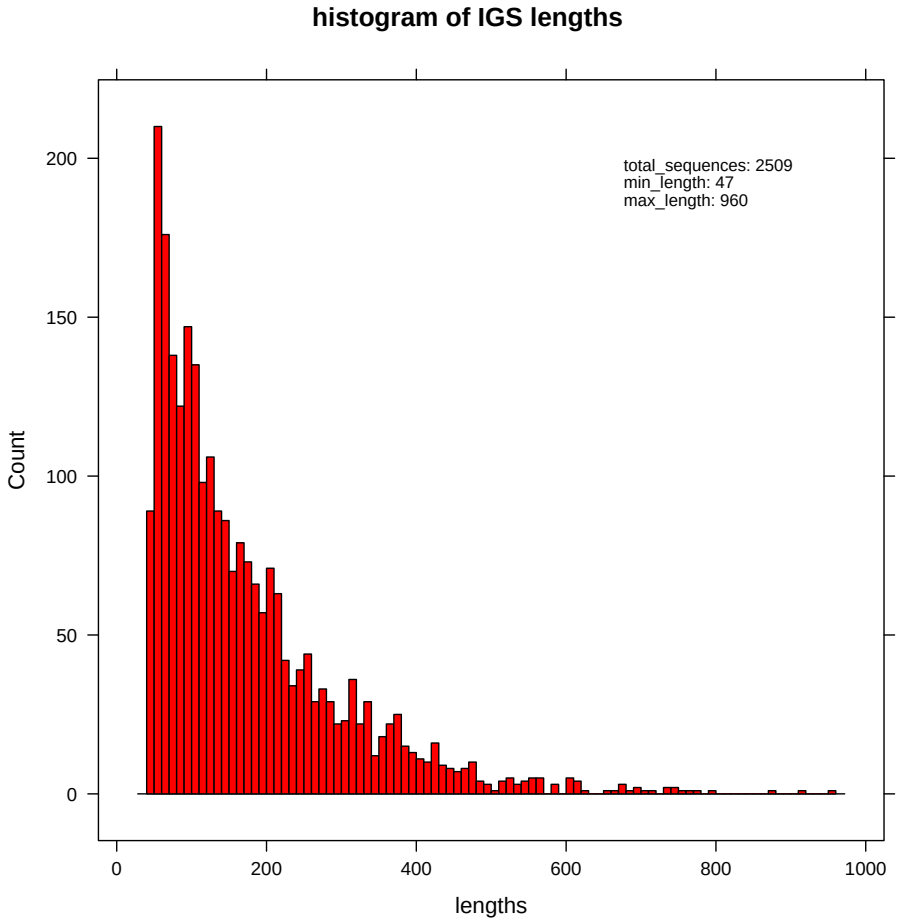

Figure 3.4: Histogram of *E. coli* intergenic sequence lengths.

$p(X_{i,j} = A|X_{i,j-1} = T, X_{i,j-2} = C)$ increased from 0.1134 to 0.1755. The most significant change was for the trinucleotide CTG: $p(X_{i,j} = G|X_{i,j-1} = T, X_{i,j-2} = C)$ decreased from 0.4359 in the full genome to 0.2928 in the intergenic sequences. This difference in nucleotide distribution has been noted previously; poly-A, poly-T and poly-AT repeats have been shown to occur often in intergenic sequences [86]. These repeats can present a problem when attempting to discover motifs with a high AT-content.

The differences in nucleotide (and particularly di- and trinucleotide) composition between the full genome and the intergenic sequences motivates the construction of a background model which is more complex than a simple 0th order, or 'frequency', model. It is a reasonable assumption that transcription factor binding sites to be discovered will be located within intergenic regions. Therefore, it has been argued that a background model constructed using intergenic data should be more successful at modelling positions which are not part of motif occurrences (for example, by Thijs, *et al.* [166]). The construction, implementation and evaluation of more complex background models in the context of both deterministic and stochastic EM is described in Section 6.3.

3.5 Alphaproteobacteria datasets

Alphaproteobacteria is a class of bacteria in the phylum Proteobacteria. Its members are very diverse, with a wide range of genome sizes and metabolic abilities. Genome sizes, measured in megabasepairs (Mb), vary from intracellular pathogens such as *Rickettsia prowazekii* (1.1Mb) and *Bartonella quintana* (1.58Mb), through average size genomes in *Caulobacter crescentus* (4.02Mb) and *Sinorhizobium meliloti* (3.65Mb) to the large genome of *Bradyrhizobium japonicum* (10.3Mb). The GC-content of Alphaproteobacterial genomes also varies widely, from 57-70%. Over the past decade, genome sequences for around 260 Alphaproteobacterial species[2] have been determined and show great diversity in genome size and architecture; this is partially due to the rapid adaptation of the bacteria to deal with different habitats[3].

Genes are not necessarily conserved between closely-related species and as few as 33% of genes in one species may have a homolog in another Alphaproteobacterial

[2]For which their genomes have been completely sequenced and published, according to the GOLD database.

[3]Due to their great diversity, the Proteobacteria are named after Proteus, the Greek god of the sea capable of assuming many different forms [158].

species. There are also many species-specific genes, reflecting adaptation to particular environments. Looking at the correlation between increase in genome size and number of genes in a particular functional group there are dramatic increases in numbers for genes involved in adaptation (for example, energy metabolism, transport and regulatory functions) whilst some species with some of the smallest genomes have virtually no regulatory genes [30]. Approximately 200 protein-coding and non-coding genes are present in all Alphaproteobacterial genomes [30].

Many Alphaproteobacterial species are aquatic; this includes the SAR11 clade and *Rhodobacter* species, which together are estimated to comprise as much as 30-50% of all bacteria in the ocean surface waters. The well characterised *C. crescentus* is a fresh water aquatic bacterium.

Despite their differences, many of the free living Alphaproteobacteria found in microaerobic habitats share the following characteristics: a capacity for autotrophic growth, multiple and complex chemotaxis systems, iron transport systems, adaptations for growth at low nutrient concentrations and the capacity for membrane invagination. The photosynthetic bacteria *Rhodobacter sphaeroides*, *Rhodobacter capsulatus* and *Rhodospirillum centenum* are all capable of autotrophic growth with hydrogen.

3.5.1 Selected Alphaproteobacteria species

Regarding the aims of this project, the heterogeneity of the Alphaproteobacterial species presents a number of potential problems, hence the choice of Alphaproteobacterial species to be studied is important. As noted above, genome sizes are highly diverse; this becomes important when selecting species for testing, as small genomes will have both fewer regulatory genes and fewer target genes. Ideally, the majority of species selected for analysis should contain the regulatory region of interest (the regulators studied are described in Section 3.5.2) and a reasonable number of target genes. The diversity of metabolism within the Alphaproteobacteria means that regulatory genes for particular functions (such as photosynthesis) will be restricted to a narrow evolutionary group. A phylogenetic analysis of Alphaproteobacterial species should aid selection of species in this case. However, it should be noted that there is also a bias in terms of the species that have been sequenced and therefore are available for study: particular species with specific functions of interest (for instance, photosynthetic bacteria and symbionts) will be highly represented amongst the available genomes.

Table 3.3 presents the species initially selected for study. Two species from each of

the *Acidiphilium*, *Bartonella*, *Brucella*, *Caulobacter*, *Ehrlichia*, *Gluconacetobacter*, *Methylobacterium*, *Orientia*, *Rhizobium*, *Rhodopseudomonas*, *Rickettsia*, *Wolbachia* and *Zymomonas* genera were chosen, covering a wide range of species from the two main Alphaproteobacterial subclasses, Caulobacteridae and Rickettsidae (the remaining subclass, Magnetococcidae, contains relatively few species).

The genes coding for the prokaryotic 16S ribosomal RNA (16S rRNA) are often used in phylogenetic analysis due to their high level of conservation between species. Alignment of these genes allows the construction of a phylogenetic tree, which shows the inferred evolutionary relationships between species; species joined together in the tree are inferred to have descended from a common ancestor. The 16S rRNA sequences for the 26 Alphaproteobacterial species in Table 3.3 were extracted from the greengenes database [47]; these sequences were then aligned using ClustalW and a phylogenetic tree (Figure 3.5) created using the neighbour-joining method in MEGA [164]. While Ferla, *et al.* [61] have noted there is some disagreement on the Alphaproteobacterial phylogeny, the constructed tree displays a high degree of similarity with accepted Alphaproteobacterial phylogenetic trees [77, 177], with clear clusters corresponding to biological order. The tree in Figure 3.5 is observed to split into two main branches based on subclass, with one branch comprising the genera from the Rickettsidae subclass (*Ehrlichia*, *Orientia*, *Rickettsia* and *Wolbachia*) and the other comprising the genera from Caulobacteridae.

3.5.2 Alphaproteobacterial TFBS motifs

In this section, two datasets containing previously characterised Alphaproteobacterial motifs are constructed. These datasets are used to validate the motif discovery algorithm developed in Chapter 5 (MITSU). A further dataset containing uncharacterised data is also constructed; MITSU will be applied to this dataset in order to make some novel predictions about the TFBS motif and consensus sequence for this previously uncharacterised regulator.

Characterised Alphaproteobacterial motifs

The characterised CtrA and FnrL motifs will be used to validate MITSU. Construction of datasets containing these motifs is outlined below; in the cases of CtrA and FnrL, the genes controlled by these regulators have been determined through previous studies.

Species	NCBI Accession number
Acidiphilium cryptum str. JF5	CP000697
Acidiphilium multivorum str. AIU301	AP012035
Bartonella clarridgeiae str. 73	FN645454
Bartonella quintana str. Toulouse	BX897700
Brucella abortus str. S19	CP000887 - CP000888
Brucella melitensis str. M28	CP002459 - CP002460
Caulobacter crescentus str. CB15	AE005673
Caulobacter sp. str. K31	CP000927
Ehrlichia canis str. Jake	CP000107
Ehrlichia ruminantium str. Welgevonden	CR925678
Gluconacetobacter diazotrophicus str. PAI 5	AM889285
Gluconobacter oxydans str. 621H	CP000009
Methylobacterium extorquens str. CM4	CP001298
Methylobacterium nodulans str. ORS 2060	CP001349
Orientia tsutsugamushi str. Boryong	AM494475
Orientia tsutsugamushi str. Ikeda	AP008981
Rhizobium etli str. CFN 42	CP000133
Rhizobium leguminosarum bv *trifolii* str. WSM 2304	CP001191
Rhodopseudomonas palustris str. BisA53	CP000463
Rhodopseudomonas palustris str. BisB18	CP000301
Rickettsia rickettsii str. Iowa	CP000766
Rickettsia typhi str. Wilmington	AE017197
Wolbachia pipientis str. wPip	AM999887
Wolbachia sp. str. wRi	CP001391
Zymomonas mobilis str. ATCC 10988	CP002850
Zymomonas mobilis str. ZM4 ATCC 31821	AE008692

Table 3.3: Species name and NCBI accession number for the 26 Alphaproteobacterial species initially selected for study. While the majority of the Alphaproteobacteria have only one chromosome, *Brucella abortus* str. S19 and *Brucella melitensis* str. M28 both have two chromosomes. Accession numbers for plasmids are not shown. Note that *Methylobacterium extorquens* str. CM4 is synonymous with *Methylobacterium chloromethanicum* str. CM4. As of 2005, the former name is recommended; however, gene locus IDs remain based on the latter (beginning 'Mchl').

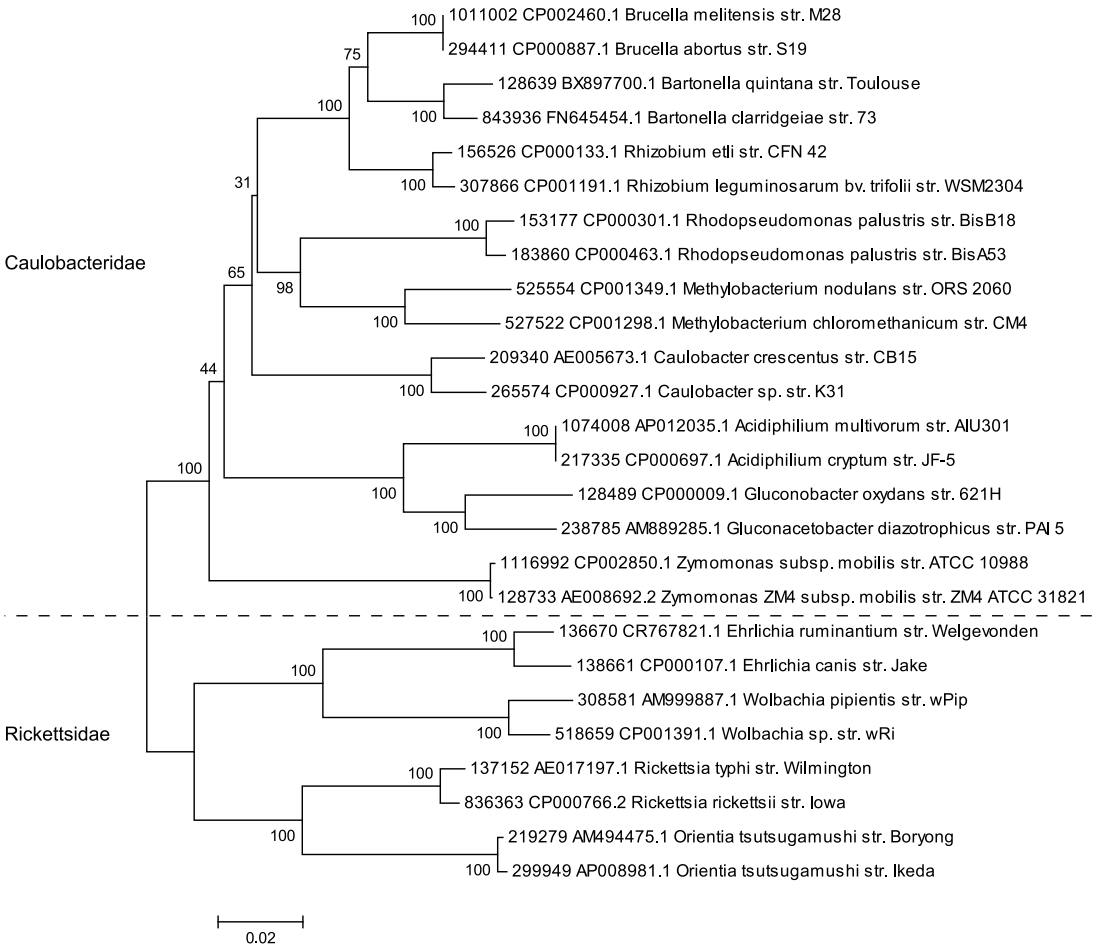

Figure 3.5: Phylogenetic tree created from the 16S rRNA regions of the 26 selected Alphaproteobacterial species listed in Table 3.3. The tree was created using the neighbour-joining method in MEGA. The numbers on the nodes indicate bootstrap scores (100 replicates) observed in the analysis. The phylogenetic tree splits into two main branches corresponding to the Alphaproteobacterial subclasses Caulobacteridae and Rickettsidae (dotted line)

CtrA The cell cycle transcriptional regulator A (CtrA) transcription factor is a well-characterised one-component regulator found in the Alphaproteobacteria but not the Gammaproteobacteria. It plays an important role in controlling the cell cycle in the model organism *Caulobacter crescentus*. CtrA has been shown to, either directly or indirectly, control at least 25% of the 553 cell cycle-regulated genes in *C. crescentus* [96]. CtrA is also thought to be important in chemotaxis [97]. The phosphorylated form of the CtrA transcription factor directly controls genes involved in cell division, DNA methylation and flagellar and pili biogenesis; it is also known to repress the initiation of DNA replication [96]. The CtrA transcription factor is known to bind to DNA as a dimer, with a well-defined consensus sequence `TTAA-N7-TTAAC`; in addition, Laub, *et al.* have identified a second (ungapped) consensus sequence `TTAACCAT` which is thought to be a possible extension of the 3' half-site [96].

Laub, *et al.* report 55 genes which have been experimentally determined to be directly regulated by CtrA in *C. crescentus* CB15 [96]. These genes can be hierarchically clustered based on their expression profiles, yielding five clusters. Almost all of the genes in the two largest clusters (28 out of 30 genes in clusters D and E in Laub, *et al.*'s study) have expression profiles which suggest that their induction is driven predominantly by CtrA [96]. Two datasets named CtrA-cD and CtrA-cE were created using the 200nt upstream sequence for each gene in clusters D and E, respectively; each cluster contained 15 genes. A third dataset, CtrA-cDE, was created by combining the datasets for clusters D and E. Tables A.1 and A.2 in Appendix A (pages 203 and 204) list the gene product annotations for the genes in the CtrA-cD and CtrA-cE datasets, respectively. The results of experiments using the CtrA datasets are presented in Section 6.1.

FnrL The *E. coli* fumarate and nitrate reductase (FNR) transcription factor is important in the expression of a number of genes involved in anaerobic metabolism. FnrL is a one-component homologue of the FNR regulator found in the Alphaproteobacteria [54, 184]. In combination with the PrrBA two-component system and the AppA/PpsR antirepressor/repressor system, FnrL has been shown to control expression of photosynthesis genes in *R. sphaeroides* [184]. Like CtrA, the FnrL transcription factor is known to bind to DNA as a dimer, with a canonical consensus sequence `TTGAT-N4-ATCAA` [54].

Dufour, *et al.* note that 63 genes have been experimentally determined to be regulated by FnrL in *R. sphaeroides* 2.4.1 [54]. Table A.3 in Appendix A (page 205) lists

the gene product annotations for these genes. The genome of *R. sphaeroides* consists of two chromosomes and five naturally occurring plasmids. Chromosome 1 is much larger than chromosome 2; 57 of the genes in Table A.3 are found in chromosome 1 but only 6 in chromosome 2. Following Dufour, *et al.*, the FnrL-63 dataset was constructed by extracting the 300nt upstream sequences for each gene.

A subset of 20 genes from the above dataset have also been shown to be part of the predicted core FNR regulon that is conserved across many Alphaproteobacterial species [54]. Table A.4 in Appendix A (page 207) lists the gene product annotations for these genes. As before, the 300nt upstream sequences for each gene were extracted to construct the FnrL-20 dataset.

Having determined the *R. sphaeroides* genes which are part of the predicted core FNR regulon, datasets were created using the homologous genes in the Alphaproteobacterial species selected for study (Table 3.3). The Alphaproteobacterial genera *Bartonella*, *Ehrlichia*, *Rickettsia* and *Wolbachia* are known not to possess proteins in the CRP/FNR protein superfamily and were therefore not studied [54]. Further, it is interesting to search for the FnrL motif in species which were not studied by Dufour, *et al.*; this leaves 7 species, namely *A. multivorum* str. AIU301, *B. abortus* str. S19, *B. melitensis* str. M28, *O. tsutsugamushi* str. Boryong, *O. tsutsugamushi* str. Ikeda, *Z. mobilis* str. ATCC 10988 and *Z. mobilis* str. ZM4 ATCC 31821. The genes used in these datasets are listed in Appendix A. The results of experiments using the FnrL datasets are presented in Section 6.1.

Previously uncharacterised Alphaproteobacterial motifs

The NtrX regulator is chosen for tests aimed at the discovery of novel motifs from previously uncharacterised data. The lack of previously identified regulatory regions and motifs makes the construction of datasets more complex. Although the regulons controlled by NtrX are known, a list of NtrX-controlled genes has not been determined.

NtrX The nitrogen assimilation transcription regulator (NtrYX) two-component system is involved in the adaptation of *B. abortus* to oxygen-limited conditions [35]. The NtrY sensor protein is activated under low oxygen tension and the regulatory protein NtrX regulates the expression of denitrification genes, allowing respiration of nitrate instead of oxygen [45]. NtrX also regulates the expression of high-affinity cytochrome oxidases, which enable efficient respiration at low oxygen concentration [109]. High-affinity cytochrome oxidases are known to be essential for bacterial virulence under

anaerobic conditions. In *Brucella* species, the NtrYX pathway is involved in bacterial virulence, causing brucellosis[4]. NtrYX has also been linked to the virulence of the Betaproteobacterium *Bordetella bronchiseptica*.

In other bacterial species, NtrX is known to regulate several additional systems. For instance, in the photosynthetic bacterium *Rhodobacter capsulatus*, it has been demonstrated that NtrX is involved in the regulation of the photosynthetic genes *puf* and *puc*, which are upregulated under low oxygen tension [73]. Recent proteomic analysis of *Brucella suis* has shown that NtrX increases the expression of several enzymes including those involved in fatty acid oxidation and citrate fermentation [40]. A recent study has also suggested that NtrX plays a part in the expression of genes controlling flagellum formation in *Sinorhizobium meliloti* [171]. The NtrX transcriptional activator protein is found reasonably widely in Alphaproteobacteria; however, neither a regulatory motif nor a consensus sequence has yet been defined for NtrX.

Homologous genes were determined using the BLAST tool. The non-redundant protein sequences of the selected Alphaproteobacterial species were searched using a protein-protein BLAST (blastp, version 2.2.29+). The default parameters were used, with the exception of the 'expect threshold' parameter, which was set to 10^{-8}, in order to ensure that only close homologs were returned. If no results were returned for a particular species, no homologs were judged to be present.

Homologs of the *ntrX* gene were revealed to be present in 24 of the Alphaproteobacteria selected for study (Table 3.3). Only the species in the *Wolbachia* genus did not have a homologous gene. Having determined the species in which *ntrX* is present, further BLAST searches were carried out in order to discover genes known to be regulated by *ntrX*. Dahouk, *et al.* [40] have determined ten regulons to be regulated by *ntrX* in *Brucella suis*, namely those coding for nitrate reductase (*narGHIJK* operon), nitrite reductase (*nirKV* operon), nitric oxide reductase (*norBCDEFQ* operon), nitrous oxide reductase (*nosDFLRXYZ* operon), cytochrome oxidases (*cydCDAB* and *ccoNOQP* operons), nitrogen fixation (*nifA*), succinoglycan (*exoB*, *exoY*, *exoK*, *exoN*, and *exoU*), flagellin (*flaA* and *flaD*) and the regulatory genes *visN* and *visR*. Although these regulons have been determined in *Brucella suis*, it is unlikely that all of the Alphaproteobacteria selected for study contain all of these genes. BLAST protein searches are used to determined which regulons are present in each species. Beyond determining whether a species contains *ntrX*-regulated genes, it is also important to determine the

[4]Also known by many other names, including Malta fever, Mediterranean fever and Rock fever of Gibraltar (in humans) and Bang's disease (in animals).

order of these genes, particularly in cases where genes form part of an operon (the operon mechanism is described in Section 1.2). In such cases, the *ntrX* binding site is most likely to be upstream of the first gene in the operon. However, this gene need not be the same in all species, therefore extra care is required in order to determine the upstream sequences likely to contain motif occurrences. Following Dufour, *et al.* [54], operon predictions were obtained from the VIMSS database[5], where they were predicted using the computational method for predicting prokaryotic operons described by Price, *et al.* [137].

Homologs of the *narG* (nitrate reductase) gene were determined to be present in six of the Alphaproteobacteria selected for study (Table 3.4). *narG* was determined to be one of several subunits in the *nar* operon, although not necessarily the first; *narK* is determined to be the first gene in the operon equally often. It is noted that the operon is also often predicted to include the *surA* gene (coding for peptidyl-prolyl cis-trans isomerase) and a gene coding for a hypothetical protein; it is unclear whether these genes are genuinely part of the *nar* operon, or just predicted to be part of the operon .

The *nirK* (nitrite reductase) gene was found in four of the Alphaproteobacterial species (Table 3.5). The *nirK* gene is predicted to work alone, instead of as part of an operon.

The *norB* (nitric oxide reductase) gene is found in three species (Table 3.6). A clear *nor* operon is only observed in *R. etli* str. CFN 42; however, the structure of the predicted operon in *B. abortus* str. S19 suggests that the *clpA/B* gene may be synonymous with *norQ* and that the first unnamed gene in *B. abortus* str. S19 and *R. palustris* str. BisA53 (denoted '*?*') is *norC*. Similarly, it is possible that the second unnamed gene in *R. palustris* str. BisA53 is *norQ*.

Similarly, the *nosD* (nitrous oxide reductase) gene was only found in three species (Table 3.7). The operon structure is more complex in this case, although the first gene in the operon is *nosR* for both *Rhodopseudomonas* species. As for *narG* above, it is unclear whether the hypothetical protein predicted as part of the operon in *B. abortus* S19 is genuinely part of the *nos* operon. If not, the first gene would again be *nosR*.

The *cydC* (cytochrome oxidase) gene is found widely in the selected Alphaproteobacterial species, occurring in 11 species (Table 3.8). It is noted that the *cyd* operon occurs twice in *A. cryptum* str. JF-5, suggesting that the genes have been duplicated. However, the gene product annotation for each occurrence is different, which suggests that while the duplicated genes may have once had the same function, they have

[5] www.microbesonline.org

Species	Gene product annotation	Locus ID	Operon
B. abortus str. S19	NarG, respiratory nitrate reductase, alpha subunit	BAbS19_II08330	*narKGYJIsurAhp*
B. melitensis str. M28	nitrate reductase subunit alpha	BM28_B0294	*narKGHJIsurAhp*
M. nodulans str. ORS 2060	nitrate reductase subunit alpha	Mnod_2128	*narGYJIsurAhp*
Caulobacter sp. str. K31	nitrate reductase subunit alpha	Caul_3864	*narKKGYJIsurAhp*
A. cryptum str. JF-5	nitrate reductase subunit alpha	Acry_1581	*narGYJIsurAnarK*
A. multivorum str. AIU301	respiratory nitrate reductase subunit alpha	ACMV_16270	*narGYJVhphp*

Table 3.4: Alphaproteobacterial species determined to contain *narG* genes. Genes coding for hypothetical proteins are denoted '*hp*'.

Species	Gene product annotation	Locus ID	Operon
B. abortus str. S19	Copper-containing nitrite reductase precursor	BAbS19_II08720	*nirK*
B. melitensis str. M28	Copper-containing nitrite reductase precursor	BM28_B0251	*nirK*
R. palustris str. BisA53	nitrite reductase, copper-containing	RPE_4071	*nirK*
R. etli str. CFN 42	nirK nitrite reductase	RHE_PF00525	*nirK*

Table 3.5: Alphaproteobacterial species determined to contain *nirK* genes.

Species	Gene product annotation	Locus ID	Operon
B. abortus str. S19	Cytochrome c oxidase, subunit I	BAbS19_II08830	*?norBclpA/BnorD*
R. palustris str. BisA53	cytochrome c oxidase, subunit I	RPE_0621	*?norB?*
R. etli str. CFN 42	nitric oxide reductase protein	RHE_PF00516	*norCBQD*

Table 3.6: Alphaproteobacterial species determined to contain *norB* genes. Unnamed genes are denoted '*?*'.

Species	Gene product annotation	Locus ID	Operon
B. abortus str. S19	Carbohydrate binding and sugar hydrolysis	BAbS19_II08560	*hpnosRZDccmAnosYLapbE*
R. palustris str. BisB18	periplasmic copper-binding	RPC_0429	*nosRZDccmAnosYLapbE*
R. palustris str. BisA53	periplasmic copper-binding protein	RPE_3096	*nosRZDccmAnosYLapbE*

Table 3.7: Alphaproteobacterial species determined to contain *nosD* genes. Genes coding for hypothetical proteins are denoted '*hp*'.

evolved to perform different functions. The operon structure is highly variable. Although *cydDC* is a common operon structure (either on its own or in combination with other genes), *cydC* occurs without *cydD* in three species.

Similar to *cydC*, the *ccoN* (cytochrome oxidase) gene is found widely in the selected Alphaproteobacterial species, occurring in eight species (Table 3.9). Although the operon structure is similarly complex, *ccoN* is usually the first gene within the operon. It is noted that the operon appears twice in both *M. nodulans* str. ORS 2060 and *R. etli* str. CFN 42. In the latter case, the operon appears once in plasmid D and once in plasmid F; however, the genes have different names. This suggests that the *fixNOQP* genes in *R. etli* str. CFN 42 (and *B. abortus* str. S19) are synonymous with the *ccoNOQP* genes. The operons in the *Methylobacterium* and *Rhodopseudomonas* species all contain *ccNO?cccA*. Based on the operon structure of the remaining species, the unnamed gene (marked '*?*') may be *ccoQ* and similarly *cccA* may be synonymous with *ccoP*. As with the analysis of *narG*, it is unclear whether the *napH*, *fixH*, *zntA* and *fixS* genes are genuinely part of the *cco* operon, or incorrectly predicted to be part of the operon.

Homologs of the *nifA* (nitrogen fixation) gene were determined to be present in eight of the Alphaproteobacteria selected for study (Table 3.10). Like the *nirK* gene, the *nifA* gene was confirmed not to be part of an operon.

Of the *exo* genes, only *exoN*, *exoU* and *exoY* were found, occurring in only one of the selected species. Results for the flagellin-coding genes *flaAD* and the regulatory genes *visNR* were inconclusive in the Alphaproteobacterial species initially selected for study. A BLAST protein search for the *Campylobacter flaA* gene (performed as described above) produced some partially conserved proteins, but no significant results. Similarly, a search for *visN* returned a partial match against the LuxR gene family in the *Rhizobium* species, but no significant results in the initially selected species. The *exo*, *fla* and *vis* regulons were therefore not studied further.

The occurrences of regulons controlled by *ntrX* in the selected Alphaproteobacterial species are summarised in Table 3.11. Datasets were constructed for the *nar*, *nir*, *nor*, *nos*, *nif*, *cyd* and *cco* regulons using the 200nt upstream region for the genes determined to be first in each operon in each species. Tables A.5-A.11 in Appendix A list the gene product annotations for the genes in these datasets. Further datasets were constructed using combinations of these datasets. A cytochrome oxidase dataset consisting of 22 sequences was constructed by combining the *cyd* and *cco* datasets. A 'nitrogen' dataset was constructed by combining the *nar*, *nir*, *nor*, *nos* and *nif* datasets. This

Species	Gene product annotation	Locus ID	Operon
B. abortus str. S19	ABC transporter, ATP-binding/permease protein	BAbS19_II06790	*?cydDCAappBybgT*
R. palustris str. BisB18	ABC transporter related	RPC_1019	*caiClivKHcydClivF*
M. extorquens str. CM4	ABC transporter related	Mchl_1558	*cydDC*
R. palustris str. BisA53	ABC transporter related	RPE_0595	*acslivKHcydClivF*
G. oxydans str. 621H	Transport ATP-binding protein CydD	GOX2410	*cydDC*
G. diazotrophicus str. PAI 5	ABC transporter, CydDC cysteine exporter (CydDC-E) family, permease/ATP-binding protein CydC	Gdia_3257	*rpoErpoEhpbaeScydDC*
R. leguminosarum bv. *trifolii* str. WSM2304	ABC transporter, CydDC cysteine exporter (CydDC-E) family, permease/ATP-binding protein CydC	Rleg2_6535	*?cydDCAappBybgT*
R. etli str. CFN 42	putative multidrug ABC transporter, ATP-binding and permease protein	RHE_PF00035	*cydC*
C. crescentus str. CB15	ABC transporter, ATP-binding protein Cydc	CC0760	*cydDC*
Caulobacter sp. str. K31	ABC transporter related	Caul_0632	*cydDCntrBfixJK*
A. cryptum str. JF-5	ABC transporter, transmembrane region, type 1	Acry_0553	*cydDC*
A. cryptum str. JF-5	Beta tubulin, autoregulation binding site	Acry_1636	*cydDC*

Table 3.8: Alphaproteobacterial species determined to contain *cydC* genes. Genes coding for hypothetical proteins are denoted '*hp*'. Unnamed genes are denoted '*?*'.

Species	Gene product annotation	Locus ID	Operon
B. abortus str. S19	Cytochrome c oxidase cbb3-type	BAbS19_I03630	*ccoNfixOccoQPnapHfixHzntAfixS*
R. palustris str. BisB18	cytochrome c oxidase, cbb3-type, subunit I	RPC_0015	*ccoNO?cccAnapHfixHzntAfixS*
M. nodulans str. ORS 2060	cytochrome c oxidase, cbb3-type, subunit I	Mnod_2111	*ccoNO?cccA*
M. nodulans str. ORS 2060	cytochrome c oxidase, cbb3-type, subunit I	Mnod_5230	*ccoNO?cccA*
R. palustris str. BisA53	cytochrome c oxidase, cbb3-type, subunit I	RPE_0018	*ccoNO?cccAnapHfixHzntAfixS*
R. leguminosarum bv. *trifolii* str. WSM2304	cytochrome c oxidase, cbb3-type, subunit I	Rleg2_5015	*ccoNOQcccAhp*
R. etli str. CFN 42	fixNf cytochrome C oxidase, fixN chain protein	RHE_PF00507	*fixNfOfQfPf*
R. etli str. CFN 42	cytochrome C oxidase, fixN chain protein	RHE_PD00296	*ccoNOQcccA*
C. crescentus str. CB15	cytochrome c oxidase, CcoN subunit	CC1401	*ccoNOQPnapH?zntAfixS*
Caulobacter sp. str. K31	cytochrome c oxidase, cbb3-type, subunit I	Caul_2437	*ccoNOQcccAnapHfixHzntAfixS*

Table 3.9: Alphaproteobacterial species determined to contain *ccoN* genes. Genes coding for hypothetical proteins are denoted '*hp*'. Unnamed genes are denoted '*?*'.

Species	Gene product annotation	Locus ID	Operon
G. diazotrophicus str. PAI 5	Nif-specific regulatory protein	GDI_0429	*nifA*
M. extorquens str. CM4	Fis family transcriptional regulator	Mchl_1311	*nifA*
M. nodulans str. ORS 2060	Fis Family transcriptional regulator NifA	Mnod_4004	*nifA*
R. etli str. CFN 42	transcriptional regulator NifA protein	RHE_PD00228	*nifA*
R. leguminosarum bv. *trifolii* str. WSM2304	Fis family transcriptional regulator	Rleg2_5044	*nifA*
R. palustris str. BisA53	transcriptional regulator NifA	RPE_4543	*nifA*
R. palustris str. BisB18	transcriptional regulator NifA	RPC_4475	*nifA*
Zymomonas mobilis str. ZM4 ATCC 31821	transcriptional regulator NifA	ZMO1816	*nifA*

Table 3.10: Alphaproteobacterial species determined to contain *nifA* genes.

dataset consisted of 24 sequences. The results of experiments using the NtrX datasets are presented in Section 6.1.

Species	Regulator						
	nar	*nif*	*nir*	*nor*	*nos*	*cyd*	*cco*
A. cryptum str. JF-5	+					+	
A. multivorum str. AIU301	+					+	
B. clarridgeiae str. 73							
B. quintana str. Toulouse							
B. abortus str. S19	+		+	+	+	+	+
B. melitensis str. M28	+		+				
C. crescentus str. CB15						+	+
Caulobacter sp. str. K31	+					+	+
E. canis str. Jake							
E. ruminantium str. Welgevonden							
G. diazotrophicus str. PAI 5		+				+	
G. oxydans str. 621H						+	
M. extorquens str. CM4		+				+	
M. nodulans str. ORS 2060	+	+					+
O. tsutsugamushi str. Boryong							
O. tsutsugamushi str. Ikeda							
R. etli str. CFN 42		+	+	+		+	+
R. leguminosarum bv. *trifolii* str. WSM2304		+				+	+
R. palustris str. BisA53		+	+	+	+	+	+
R. palustris str. BisB18		+			+	+	+
R. rickettsii str. Iowa							
R. typhi str. Wilmington							
Z. mobilis str. ATCC 10988							
Z. mobilis str ZM4 ATCC 31821		+					

Table 3.11: Occurrences of NtrX-controlled regulons in selected Alphaproteobacterial species.

3.6 Additional datasets

3.6.1 CRP

The well-known cAMP receptor protein (CRP) dataset has been used in several studies of motif discovery algorithms (for example, those by Bi [23], Lawrence, *et al.* [98] and Stormo and Hartzell [162]). Briefly, CRP is a prokaryotic transcription factor that is important in the regulation of genes involved in energy metabolism. The CRP dataset consists of 18 sequences, each of which is 105nt in length. The dataset contains 24 CRP binding sites determined either by footprinting experiments or sequence similarity to confirmed binding sites; each sequence in the dataset contains one or two sites. Each binding site is 22nt in width. The mean motif conservation is calculated to be 0.48 bits/col; however, it is noted that this value does not account for the motif structure. The CRP transcription factor is known to bind as a dimer, with consensus sequence `N3-TGTGA-N6-TCACA-N3`. The known binding site motif is shown in Figure 3.6. Mean conservation is calculated here to be 0.91 bits/col in the 5' conserved region and 0.74 bits/col in the 3' conserved region.

Figure 3.6: CRP motif sequence logo constructed from the 24 binding sites contained in the CRP dataset.

3.6.2 MalI/SoxR

In order to test the ability of MITSU to discover multiple motifs within a single dataset, a small dataset containing the *E. coli* MalI and SoxR motifs was created. The MalI/SoxR dataset is the union of the individual datasets created in Section 3.2, consisting of two sequences containing occurrences of the MalI motif and two sequences containing occurrences of the SoxR motif. As noted in Table 3.1, the MalI motif is 12nt in width and the SoxR motif is 19nt in width. Both motifs are perfectly conserved.

Chapter 4

Improving deterministic motif discovery algorithms

This chapter examines the theoretical issues and derivations of the deterministic EM algorithm in its application to motif discovery, starting with a basic explanation of how EM is used to solve the motif discovery problem. This is followed by a definition of the EM expressions used in these algorithms, including a discussion of how different sequence models affect these expressions (Sections 4.1 and 4.2). Existing implementations of deterministic motif discovery algorithms have several areas which require improvement. Section 4.3 addresses the fact that implementations of the EM algorithm for motif discovery often assume that all input sequences (that is, a number of promoter regions upstream of coregulated genes) are of equal length. This is solved by generalising the EM expressions, allowing input sequences to have unequal lengths at the expense of slightly more complex expressions. Having explored how the alternative statistical assumptions about the distribution of motif occurrences impact on the EM calculations, Section 4.4 turns to a practical issue in motif discovery that cannot be addressed by modifying the statistical model: that of choosing the most likely motif width. A novel heuristic (named MCOIN) for determining motif width is presented; experimental results show that MCOIN improves on the current most popular method (E-value of the resulting multiple alignment) as a predictor of motif width.

The general EM algorithm is explained in Section 2.2.2. Briefly, some initial values for the model parameters (θ) are estimated, then two steps are repeatedly carried out. In the expectation step, or E-step, the current parameter values are used to evaluate the (posterior) probability of the latent data ($\mathbf{Z}$) given the observed data ($\mathbf{X}$). This probability is then used in the maximisation step, or M-step, to reestimate the parame-

ters. The E-step and M-step are repeated until the parameter values converge, or until a fixed number of iterations have been carried out.

The EM algorithm is used in the context of motif discovery as follows: The initial values for the motif model (that is, the initial PWM parameters) are estimated. In the OOPS and ZOOPS models, this estimation is often carried out by choosing a motif start point at random for each input sequence and then counting the numbers of each nucleotide at each motif position, creating a consensus model from these start points. Alternatively, the initial model may be created by maximum likelihood (effectively a consensus over all W-mers in the dataset). For each width-W subsequence in the dataset, the E-step of the algorithm calculates the probability of that subsequence being an occurrence of the motif, based on the current motif model parameters. This procedure can be viewed as estimating the position of occurrences of the motif within the input dataset. The M-step reestimates the model parameters by maximising the expected value of the log likelihood function. That is, the model parameters are adjusted in order to maximise the log likelihood given the current estimates for the motif positions within the input dataset. These two steps are repeated iteratively; the algorithm is deemed to have converged when there is very little (or no) change in the motif model parameters between subsequent iterations, or equivalently, if there is no change in the estimated motif positions between subsequent iterations.

As noted in Section 2.2.2, the EM Q function is the expected value of the complete data (that is, $\{\mathbf{X},\mathbf{Z}\}$) log likelihood function. Both steps of the EM algorithm depend on the Q function: the E-step of the algorithm requires calculating the parameters of the Q function; Q is then maximised in the M-step. The ability to define the Q function is therefore of prime importance in the EM algorithm. The Q function generally becomes more complex with increasing model complexity. This will be illustrated in the context of motif discovery by the Q functions for the OOPS and ZOOPS models.

The original OOPS and ZOOPS model expressions were defined by Bailey and Elkan [6, 10]. Here, they are updated, generalised and presented in a notation that remains consistent when extended for use with the stochastic EM-based algorithm for motif discovery developed in Chapter 5.

4.1 Expectation-Maximisation expressions for the OOPS model

The OOPS sequence model assumes that there is exactly one motif occurrence in each input sequence. This means that for each sequence, W positions will be part of the motif occurrence and all other positions may be treated as background. The conditional probability of sequence X_i given the hidden variables is defined as the product of probabilities over the W positions within the motif (θ_m: recall that these values are the PWM parameters) and the remaining background positions (θ_0) within that sequence. Assuming that the motif occurs at position j in sequence X_i, the conditional probability of sequence X_i given the hidden variables is defined as:

$$p(X_i|Z_{i,j}=1,\theta) \triangleq \prod_{l\in\Delta_{i,j}}\prod_{k\in\mathcal{L}}\theta_{0,k}^{I(X_{i,l}=k)}\prod_{m=1}^{W}\prod_{k\in\mathcal{L}}\theta_{m,k}^{I(X_{i,j+m-1}=k)}. \tag{4.1}$$

where the indicator variables select the relevant model parameters according to the observed nucleotide at the position in question and $\Delta_{i,j}$ denotes the background positions (that is, all positions not contained within the motif occurrence). It is assumed for simplicity that all input sequences are of equal length; the number of possible motif positions within each sequence is represented as M. It is also assumed that the prior distribution of motif start sites within a sequence is uniform, that is:

$$p(Z_{i,j}=1|\theta) = p(Z_{i,j}=1) \triangleq \frac{1}{M}. \tag{4.2}$$

The complete data joint probability can be written:

$$\begin{aligned} p(X,Z|\theta) &= \prod_{i=1}^{N} p(X_i,Z_i|\theta) \\ &= \prod_{i=1}^{N} p(X_i|Z_i,\theta)\, p(Z_i|\theta) \\ &= \prod_{i=1}^{N}\left[\prod_{j=1}^{M} p(X_i|Z_{i,j}=1,\theta)^{Z_{i,j}}\,\frac{1}{M}\right], \end{aligned} \tag{4.3}$$

where (4.3) takes advantage of the fact that all $Z_{i,j}$ in a sequence will be 0 apart from one; that is, only one position in a sequence will be a motif start point. Since taking logs transforms a product into a sum, the log likelihood function for the complete data is therefore:

$$\ln p(X,Z|\theta) = \sum_{i=1}^{N}\left[\ln\left(\frac{1}{M}\right) + \sum_{j=1}^{M} Z_{i,j}\ln p(X_i|Z_{i,j}=1,\theta)\right] \tag{4.4}$$

and the Q function is the expected value of the log likelihood function, with respect to the conditional distribution of Z given X under the current estimate of parameters $\theta^{(t)}$:

$$\begin{aligned} Q(\theta|\theta^{(t)}) &= \mathbb{E}_{Z|X,\theta^{(t)}}\left[\ln p(X,Z|\theta)\right] \\ &= \mathbb{E}_{Z|X,\theta^{(t)}}\left[\sum_{i=1}^{N}\left\{\ln\left(\frac{1}{M}\right)+\sum_{j=1}^{M} Z_{i,j}\ln p(X_i|Z_{i,j}=1,\theta)\right\}\right] \\ &= \sum_{i=1}^{N}\left\{\ln\left(\frac{1}{M}\right)+\sum_{j=1}^{M}\mathbb{E}_{Z|X,\theta^{(t)}}\left[Z_{i,j}\right]\ln p(X_i|Z_{i,j}=1,\theta)\right\}. \end{aligned} \tag{4.5}$$

$Z_{i,j}^{(t)}$ is defined as the expected probability of a motif start point at position j in sequence i:

$$\begin{aligned} Z_{i,j}^{(t)} &\triangleq \mathbb{E}_{Z|X,\theta^{(t)}}\left[Z_{i,j}\right] \\ &= 1\cdot p(Z_{i,j}=1|X_i,\theta^{(t)}) + 0\cdot p(Z_{i,j}=0|X_i,\theta^{(t)}) \\ &= p(Z_{i,j}=1|X_i,\theta^{(t)}). \end{aligned} \tag{4.6}$$

Substituting definition (4.6) into (4.5) gives:

$$Q(\theta|\theta^{(t)}) = \sum_{i=1}^{N}\left\{\ln\left(\frac{1}{M}\right)+\sum_{j=1}^{M} Z_{i,j}^{(t)}\ln p(X_i|Z_{i,j}=1,\theta)\right\}, \tag{4.7}$$

the expression for the EM Q function in the OOPS model. Bailey and Elkan continue by summing out the prior term and rearranging to give:

$$Q(\theta|\theta^{(t)}) = \left[\sum_{i=1}^{N}\sum_{j=1}^{M} Z_{i,j}^{(t)}\ln p(X_i|Z_{i,j}=1,\theta)\right] + N\ln\frac{1}{M}. \tag{4.8}$$

The expression for the EM Q function given by Keles, *et al.* [90] is the same as that in Equation 4.8 (with the omission of the $N\ln\frac{1}{M}$ term) and is described as being 'up to a constant'; as will be noted, this term is indeed invariant with respect to θ and can be ignored for the purposes of EM.

4.1.1 E-step

The E-step of the EM algorithm requires the evaluation of the probability of the latent data $p(Z|X,\theta^{(t)})$. That is, $p(Z_{i,j}=1|X_i,\theta^{(t)}) \equiv Z_{i,j}^{(t)}$ must be evaluated for each position in each input sequence. This can be carried out by making use of (4.1); using Bayes' theorem, $Z_{i,j}^{(t)}$ is defined as:

$$Z_{i,j}^{(t)} = \frac{p(X_i|Z_{i,j}=1,\theta^{(t)})}{\sum_{j=1}^{M} p(X_i|Z_{i,j}=1,\theta^{(t)})}, \tag{4.9}$$

for all $i \in \{1,\ldots,N\}$ and $j \in \{1,\ldots,M\}$.

4.1.2 M-step

The M-step of the EM algorithm requires the maximisation of the Q function (4.7) in order to determine new parameter values. The prior term ($\sum_{i=1}^{N} \ln \frac{1}{M}$) is invariant with respect to θ and so can be ignored. Therefore, an analytical solution maximising:

$$Q(\theta|\theta^{(t)}) = \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{i,j}^{(t)} \ln p(X_i|Z_{i,j}=1,\theta) \tag{4.10}$$

is required. Substituting (4.1) into (4.10) gives:

$$\begin{aligned}
Q(\theta|\theta^{(t)}) &= \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{i,j}^{(t)} \ln \left\{ \prod_{l\in\Delta_{i,j}} \prod_{k\in\mathcal{L}} \theta_{0,k}^{I(X_{i,l}=k)} \prod_{m=1}^{W} \prod_{k\in\mathcal{L}} \theta_{m,k}^{I(X_{i,j+m-1}=k)} \right\} \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{i,j}^{(t)} \left\{ \sum_{l\in\Delta_{i,j}} \sum_{k\in\mathcal{L}} I(X_{i,l}=k) \ln \theta_{0,k} \right. \\
&\qquad \left. + \sum_{m=1}^{W} \sum_{k\in\mathcal{L}} I(X_{i,j+m-1}=k) \ln \theta_{m,k} \right\} \qquad (*) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{i,j}^{(t)} \sum_{l\in\Delta_{i,j}} \sum_{k\in\mathcal{L}} I(X_{i,l}=k) \ln \theta_{0,k} \\
&\qquad + \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{i,j}^{(t)} \sum_{m=1}^{W} \sum_{k\in\mathcal{L}} I(X_{i,j+m-1}=k) \ln \theta_{m,k}
\end{aligned} \tag{4.11}$$

Note that (*) is equivalent to the expression for the Q function given by Keles, *et al.* To simplify the expansion (4.11), the expected counts for each nucleotide $k \in \mathcal{L}$ at each position in the motif ($m = 1,\ldots,W$) and the background ($m = 0$) can be defined:

$$N_{m,k} \triangleq \begin{cases} \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{i,j}^{(t)} \sum_{l\in\Delta_{i,j}} I(X_{i,l}=k), & m=0, \\ \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{i,j}^{(t)} I(X_{i,j+m-1}=k), & m\neq 0. \end{cases} \tag{4.12}$$

Substituting (4.12) into (4.11) gives:

$$\begin{aligned}
Q(\theta|\theta^{(t)}) &= \sum_{k\in\mathcal{L}} N_{0,k} \ln \theta_{0,k} + \sum_{m=1}^{W}\sum_{k\in\mathcal{L}} N_{m,k} \ln \theta_{m,k} \\
&= \sum_{m=0}^{W}\sum_{k\in\mathcal{L}} N_{m,k} \ln \theta_{m,k}.
\end{aligned} \tag{4.13}$$

Maximisation of (4.13) with respect to θ is achieved by maximising separately for each $m \in 0,\ldots,W$ and taking advantage of Gibbs' inequality, where it can be shown that for two (discrete) probability distributions f and g:

$$\underset{g}{\operatorname{argmax}}\ f \ln g = f. \tag{4.14}$$

θ_m is already a (multinomial) probability distribution and N_m can be changed into this form by normalising over k. The parameter updates of the M-step are therefore the normalised ratio of expected nucleotide counts:

$$\theta_{m,k}^{(t+1)} = \frac{N_{m,k}}{\sum_{k \in \mathcal{L}} N_{m,k}}, \tag{4.15}$$

for $m \in \{0, \ldots, W\}$ and $k \in \mathcal{L}$.

4.2 Expectation-Maximisation expressions for the ZOOPS model

The ZOOPS model assumes that each input sequence either contains exactly one occurrence of the motif, or no occurrences of the motif. The ZOOPS model accounts for this by introducing an additional indicator variable which indicates whether a particular input sequence contains a motif occurrence or not. The new indicator variable Q_i is defined as $Q_i \triangleq \sum_{j=1}^{M} Z_{i,j}$. That is, $Q_i = 1$ if sequence i contains a motif occurrence and 0 otherwise. The OOPS model then becomes a special case of the ZOOPS model where all input sequences contain a motif occurrence. If sequence i contains a motif occurrence, the conditional probability of i given the hidden variables is the same as in the OOPS model:

$$p(X_i | Z_{i,j} = 1, \theta) \triangleq \prod_{l \in \Delta_{i,j}} \prod_{k \in \mathcal{L}} \theta_{0,k}^{I(X_{i,l}=k)} \prod_{m=1}^{W} \prod_{k \in \mathcal{L}} \theta_{m,k}^{I(X_{i,j+m-1}=k)}. \tag{4.16}$$

As in the OOPS model, (4.16) is the product of probabilities over the W positions within the motif and the remaining background positions. The conditional probability for a sequence which does not contain a motif occurrence is also defined as the product of probabilities, this time using background probabilities for all positions within sequence i:

$$p(X_i | Q_i = 0, \theta) \triangleq \prod_{l=1}^{L_i} \prod_{k \in \mathcal{L}} \theta_{0,k}^{I(X_{i,l}=k)}, \tag{4.17}$$

where L_i is the length of input sequence i. As in the OOPS model, a uniform distribution of start sites within a sequence is assumed. If the prior probability of a sequence containing a motif occurrence is defined as γ, it follows from the assumption of equal input sequence length that the prior probability of any position being a motif start point is:

$$\lambda \triangleq p(Z_{i,j} = 1 | \theta) = \frac{\gamma}{M}. \tag{4.18}$$

For simplicity the model parameters are now collected and denoted as $\phi = (\theta, \gamma)$. It is noted that the model parameters now include the prior probability of a sequence containing a motif occurrence, in addition to the motif and background models from the OOPS model. The complete data joint probability can be written as:

$$\begin{aligned} p(X,Z|\phi) &= \prod_{i=1}^{N} p(X_i,Z_i|\phi) \\ &= \prod_{i=1}^{N} p(X_i|Z_i,\phi)\, p(Z_i|\phi) \\ &= \prod_{i=1}^{N} \left[\left(\prod_{j=1}^{M} p(X_i|Z_{i,j}=1,\theta)^{Z_{i,j}} \right) \times p(X_i|Q_i=0,\theta)^{(1-Q_i)} \right. \\ &\qquad \left. \times \lambda^{Q_i} \times (1-\gamma)^{(1-Q_i)} \right]. \end{aligned} \tag{4.19}$$

As in the OOPS model, (4.19) takes advantage of the fact that all $Z_{i,j}$ in a sequence will be 0 apart from one. The first term in (4.19) is the expression for a sequence containing a motif occurrence (4.16). The second term is the expression for a sequence without a motif occurrence (4.17). Only one of these terms will be used, depending on the value of Q for sequence i. If $Q_i = 0$, then all $Z_{i,j}$ will be 0, cancelling the first term and using the second. If $Q_i = 1$, then $Z_{i,j} = 1$ for some j and the first term is used while the second term is cancelled. This cancelling works similarly for the prior terms. Note that the prior term for positions in a sequence was $1/M$ before as every sequence had a motif occurrence. Now only γ of sequences contain a motif, the prior on positions within a sequence is $\gamma/M = \lambda$ (and the prior term for a sequence not containing a motif is $1-\gamma$, from above). The log likelihood function for the complete data may be written:

$$\begin{aligned} \ln p(X,Z|\phi) = \sum_{i=1}^{N} \left(\sum_{j=1}^{M} Z_{i,j} \ln p(X_i|Z_{i,j}=1,\theta) \right) \\ + \sum_{i=1}^{N} (1-Q_i) \ln p(X_i|Q_i=0,\theta) \\ + \sum_{i=1}^{N} Q_i \ln \lambda + \sum_{i=1}^{N} (1-Q_i) \ln (1-\gamma). \end{aligned} \tag{4.20}$$

Again, $Z_{i,j}^{(t)}$ is defined as the expected probability of a motif start point at position j in sequence i:

$$\begin{aligned} Z_{i,j}^{(t)} &\triangleq \mathbb{E}_{Z|X,\phi^{(t)}} \left[Z_{i,j} \right] \\ &= 1 \cdot p(Z_{i,j}=1|X_i,\phi^{(t)}) + 0 \cdot p(Z_{i,j}=0|X_i,\phi^{(t)}) \\ &= p(Z_{i,j}=1|X_i,\phi^{(t)}), \end{aligned} \tag{4.21}$$

and, as Q_i is dependent on $Z_{i,j}$, $Q_i^{(t)}$ is defined as the expected probability of sequence i containing a motif occurrence (this reduces to a sum of the relevant $Z_{i,j}^{(t)}$ values):

$$\begin{aligned} Q_i^{(t)} &\triangleq \mathbb{E}_{Z|X,\phi^{(t)}}[Q_i] \\ &= \sum_{j=1}^{M} \mathbb{E}_{Z|X,\phi^{(t)}}\left[Z_{i,j}\right] \\ &= \sum_{j=1}^{M} Z_{i,j}^{(t)}. \end{aligned} \tag{4.22}$$

The Q function is the expected value of the log likelihood function (4.20), with respect to the conditional distribution of Z given X under the current estimate of parameters $\theta^{(t)}$:

$$\begin{aligned} Q(\phi|\phi^{(t)}) &= \mathbb{E}_{Z|X,\phi^{(t)}}[\ln p(X,Z|\phi)] \\ &= \mathbb{E}_{Z|X,\phi^{(t)}}\left[\sum_{i=1}^{N}\left(\sum_{j=1}^{M} Z_{i,j}\ln p(X_i|Z_{i,j}=1,\theta)\right)\right. \\ &\qquad + \sum_{i=1}^{N}(1-Q_i)\ln p(X_i|Q_i=0,\theta) \\ &\qquad \left. + \sum_{i=1}^{N} Q_i \ln\lambda + \sum_{i=1}^{N}(1-Q_i)\ln(1-\gamma)\right]. \\ &= \sum_{i=1}^{N}\left(\sum_{j=1}^{M}\mathbb{E}_{Z|X,\phi^{(t)}}[Z_{i,j}]\ln p(X_i|Z_{i,j}=1,\theta)\right) \\ &\qquad + \sum_{i=1}^{N}(1-\mathbb{E}_{Z|X,\phi^{(t)}}[Q_i])\ln p(X_i|Q_i=0,\theta) \\ &\qquad + \sum_{i=1}^{N}\mathbb{E}_{Z|X,\phi^{(t)}}[Q_i]\ln\lambda \\ &\qquad + \sum_{i=1}^{N}(1-\mathbb{E}_{Z|X,\phi^{(t)}}[Q_i])\ln(1-\gamma) \\ &= \sum_{i=1}^{N}\left(\sum_{j=1}^{M} Z_{i,j}^{(t)}\ln p(X_i|Z_{i,j}=1,\theta)\right) \\ &\qquad + \sum_{i=1}^{N}(1-Q_i^{(t)})\ln p(X_i|Q_i=0,\theta) \\ &\qquad + \sum_{i=1}^{N} Q_i^{(t)}\ln\lambda + \sum_{i=1}^{N}(1-Q_i^{(t)})\ln(1-\gamma), \end{aligned} \tag{4.23}$$

where (4.21) and (4.22) have been substituted as required. This is equivalent to the expression given by Bailey and Elkan and by Keles, *et al.*

4.2.1 E-step

As in the OOPS model, the E-step requires the evaluation of the probability of the latent data $p(Z|X,\theta)$, that is, $Z_{i,j}^{(t)}$ for each position. Again, Bayes' theorem is used to define $Z_{i,j}^{(t)}$ in terms of (4.16) and (4.17):

$$Z_{i,j}^{(t)} = \frac{p(X_i|Z_{i,j}=1,\theta^{(t)})\lambda^{(t)}}{p(X_i|Q_i=0,\theta^{(t)})(1-\gamma^{(t)})+\sum_{j=1}^{M} p(X_i|Z_{i,j}=1,\theta^{(t)})\lambda^{(t)}}, \tag{4.24}$$

for all $i \in \{1,\ldots,N\}$ and $j \in \{1,\ldots,M\}$.

4.2.2 M-step

The M-step of the EM algorithm requires the maximisation of the Q function (4.23) in order to find new values for the parameters (ϕ). To simplify the problem, it is noted that $Q(\phi|\phi^{(t)})$ can be split into two terms; the first relying only on θ and the second relying only on γ (recalling that λ can be expressed as a function of γ). Maximisation of $Q(\phi|\phi^{(t)})$ can be achieved by maximising separately over each term. The first term in $Q(\phi|\phi^{(t)})$:

$$\sum_{i=1}^{N}\left(\sum_{j=1}^{M} Z_{i,j}^{(t)} \ln p(X_i|Z_{i,j}=1,\theta)\right) + \sum_{i=1}^{N}(1-Q_i^{(t)}) \ln p(X_i|Q_i=0,\theta) \tag{4.25}$$

requires maximisation over θ. Substituting (4.16) and (4.17) gives:

$$\begin{aligned}
&\sum_{i=1}^{N}\left(\sum_{j=1}^{M} Z_{i,j}^{(t)} \ln\left\{\prod_{l\in\Delta_{i,j}}\prod_{k\in\mathcal{L}} \theta_{0,k}^{I(X_{i,l}=k)} \prod_{m=1}^{W}\prod_{k\in\mathcal{L}} \theta_{m,k}^{I(X_{i,j+m-1}=k)}\right\}\right) \\
&\qquad + \sum_{i=1}^{N}(1-Q_i^{(t)}) \ln\left\{\prod_{l=1}^{L_i}\prod_{k\in\mathcal{L}} \theta_{0,k}^{I(X_{i,l}=k)}\right\} \\
= &\sum_{i=1}^{N}\left(\sum_{j=1}^{M} Z_{i,j}^{(t)}\left\{\sum_{l\in\Delta_{i,j}}\sum_{k\in\mathcal{L}} I(X_{i,l}=k)\ln\theta_{0,k} + \sum_{m=1}^{W}\sum_{k\in\mathcal{L}} I(X_{i,j+m-1}=k)\ln\theta_{m,k}\right\}\right) \\
&\qquad + \sum_{i=1}^{N}(1-Q_i^{(t)})\left\{\sum_{l=1}^{L_i}\sum_{k\in\mathcal{L}} I(X_{i,l}=k)\ln\theta_{0,k}\right\}.
\end{aligned} \tag{4.26}$$

Rearranging and grouping the 'motif' and 'background' terms together yields:

$$\begin{aligned}
&\sum_{i=1}^{N}\left(\sum_{j=1}^{M} Z_{i,j}^{(t)}\sum_{l\in\Delta_{i,j}}\sum_{k\in\mathcal{L}} I(X_{i,l}=k)\ln\theta_{0,k} + (1-Q_i^{(t)})\sum_{l=1}^{L_i}\sum_{k\in\mathcal{L}} I(X_{i,l}=k)\ln\theta_{0,k}\right) \\
&\qquad + \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{i,j}^{(t)}\sum_{m=1}^{W}\sum_{k\in\mathcal{L}} I(X_{i,j+m-1}=k)\ln\theta_{m,k}
\end{aligned} \tag{4.27}$$

As in the OOPS model, the expected counts for each nucleotide $k \in \mathcal{L}$ at each position in the motif ($m = 1,\ldots,W$) and the background ($m = 0$) can be defined in order to simplify the expression:

$$N_{m,k} \triangleq \begin{cases} \sum_{i=1}^{N} \left[\sum_{j=1}^{M} Z_{i,j}^{(t)} \sum_{l \in \Delta_{i,j}} I(X_{i,l} = k) + (1 - Q_i^{(t)}) \sum_{l=1}^{L_i} I(X_{i,l} = k) \right], & m = 0, \\ \sum_{i=1}^{N} \sum_{j=1}^{M} Z_{i,j}^{(t)} \, I(X_{i,j+m-1} = k), & m \neq 0 \end{cases} \tag{4.28}$$

Note that the $m = 0$ case now includes the indicator variable Q_i, unlike the analogous expression in the OOPS model. The $m \neq 0$ case is the same as it was for the OOPS model. Substituting (4.28) into (4.27) gives:

$$\begin{aligned} &\sum_{k \in \mathcal{L}} N_{0,k} \ln \theta_{0,k} + \sum_{m=1}^{W} \sum_{k \in \mathcal{L}} N_{m,k} \ln \theta_{m,k} \\ &= \sum_{m=0}^{W} \sum_{k \in \mathcal{L}} N_{m,k} \ln \theta_{m,k}, \end{aligned} \tag{4.29}$$

exactly as the OOPS model. Maximisation over θ is therefore performed in the same way, using Gibbs' inequality. The parameter updates for θ are again the normalised ratio of expected nucleotide counts:

$$\theta_{m,k}^{(t+1)} = \frac{N_{m,k}}{\sum_{k \in \mathcal{L}} N_{m,k}}, \tag{4.30}$$

for $m \in \{0,\ldots,W\}$ and $k \in \mathcal{L}$, as in the OOPS model. Calculating the parameter update for γ requires maximising the second term in $Q(\phi|\phi^{(t)})$ over γ. The second term is:

$$\sum_{i=1}^{N} Q_i^{(t)} \ln \lambda + \sum_{i=1}^{N} (1 - Q_i^{(t)}) \ln (1 - \gamma). \tag{4.31}$$

Substituting λ using (4.18) and further substituting $S \triangleq \frac{1}{N} \sum_{i=1}^{N} Q_i^{(t)}$ gives:

$$NS \ln \frac{\gamma}{M} + N(1 - S) \ln (1 - \gamma). \tag{4.32}$$

Splitting the first log term gives and rearranging gives:

$$N(S \ln \gamma + (1 - S) \ln (1 - \gamma)) - NS \ln M. \tag{4.33}$$

Note that ($NS \ln M$) is invariant with respect to γ and can be ignored in terms of maximisation. A maximisation for:

$$N(S \ln \gamma + (1 - S) \ln (1 - \gamma)) \tag{4.34}$$

is required. This maximisation may be carried out analytically; doing so yields a maximum at $\gamma = S$. The parameter update for γ is therefore:

$$\gamma^{(t+1)} = S = \frac{1}{N}\sum_{i=1}^{N} Q_i^{(t)}. \tag{4.35}$$

Intuitively, the new value for the fraction of sequences containing a motif occurrence is simply the empirical fraction, based on the updates from the E-step.

4.3 Generalising ZOOPS expressions for Expectation-Maximisation

This section provides a generalisation of the ZOOPS sequence model expressions used in deterministic EM for motif discovery that removes the requirement that input sequences must be of equal length. Definition of these expressions is important, as they will be used as the basis for the stochastic EM algorithm for motif discovery developed in Chapter 5. In particular, removing the constraint of equal input sequence length is vital in successfully implementing the cutting heuristic which allows discovery of multiple occurrences of a motif within a single input sequence (Section 5.3.1), a technique that fulfils the same role as the TCM model in MEME. Removing the assumption that all input sequences are the same length increases flexibility at the expense of some additional mathematics; fortunately, removing this assumption does not fundamentally alter the calculations required in the E- and M-steps for the ZOOPS model.

The expressions for the conditional probability of a sequence with and without a motif occurrence (4.16 and 4.17) remain the same as in the ZOOPS model. Again, γ is defined as the prior probability of a sequence containing a motif occurrence. However, the previous definition for the prior probability of a position being a motif start position (λ) becomes problematic in the general setting developed here. In the ungeneralised ZOOPS model, λ could be used as a mathematical convenience as a prior for all sequences; now assuming that input sequences need not have equal lengths means that the prior term on each sequence will be different and a single prior is inappropriate. The simplest solution is to substitute $L_i - W + 1 = M$, therefore:

$$\lambda = \frac{\gamma}{L_i - W + 1}. \tag{4.36}$$

The dependence of the prior term on the length of the input sequence L_i is now clear. Note that if L_i *is* the same for each input sequence and M is set to $L_i - W + 1$, this

is equivalent to the previous definition. The definition of Q_i is modified in order to account for the possibility of different sequence lengths: $Q_i \triangleq \sum_{j=1}^{L_i-W+1} Z_{i,j}$. As before, $Q_i = 1$ if sequence i contains a motif occurrence and 0 otherwise. Following the new definition of λ (4.36), the generalised expression for the complete data joint probability becomes:

$$\begin{aligned} p(X,Z|\phi) &= \prod_{i=1}^{N} p(X_i,Z_i|\phi) \\ &= \prod_{i=1}^{N} p(X_i|Z_i,\phi)\, p(Z_i|\phi) \\ &= \prod_{i=1}^{N} \left[\left(\prod_{j=1}^{L_i-W+1} p(X_i|Z_{i,j}=1,\theta)^{Z_{i,j}} \right) \times p(X_i|Q_i=0,\theta)^{(1-Q_i)} \right. \\ &\qquad \left. \times \left(\frac{\gamma}{L_i-W+1} \right)^{Q_i} \times (1-\gamma)^{(1-Q_i)} \right]. \end{aligned} \tag{4.37}$$

The log likelihood function for the complete data is therefore:

$$\begin{aligned} \ln p(X,Z|\phi) = \sum_{i=1}^{N} & \left(\sum_{j=1}^{L_i-W+1} Z_{i,j} \ln p(X_i|Z_{i,j}=1,\theta) \right) \\ &+ \sum_{i=1}^{N} (1-Q_i) \ln p(X_i|Q_i=0,\theta) \\ &+ \sum_{i=1}^{N} Q_i \ln \left(\frac{\gamma}{L_i-W+1} \right) \\ &+ \sum_{i=1}^{N} (1-Q_i) \ln (1-\gamma). \end{aligned} \tag{4.38}$$

While the definition of $Z_{i,j}^{(t)}$ (4.21) remains the same as before, the definition of $Q_i^{(t)}$ is updated:

$$\begin{aligned} Q_i^{(t)} &\triangleq \mathbb{E}_{Z|X,\phi^{(t)}}[Q_i] \\ &= \sum_{j=1}^{L_i-W+1} \mathbb{E}_{Z|X,\phi^{(t)}}\left[Z_{i,j}\right] \\ &= \sum_{j=1}^{L_i-W+1} Z_{i,j}^{(t)}. \end{aligned} \tag{4.39}$$

Finally, the Q function is generalised, using the updated definitions above:

$$\begin{aligned}
Q(\phi|\phi^{(t)}) &= \mathbb{E}_{Z|X,\phi^{(t)}}\left[\ln p(X,Z|\phi)\right] \\
&= \mathbb{E}_{Z|X,\phi^{(t)}}\left[\sum_{i=1}^{N}\left(\sum_{j=1}^{L_i-W+1} Z_{i,j}\ln p(X_i|Z_{i,j}=1,\theta)\right)\right. \\
&\qquad + \sum_{i=1}^{N}(1-Q_i)\ln p(X_i|Q_i=0,\theta) \\
&\qquad \left. + \sum_{i=1}^{N} Q_i \ln\left(\frac{\gamma}{L_i-W+1}\right) + \sum_{i=1}^{N}(1-Q_i)\ln(1-\gamma)\right]. \\
&= \sum_{i=1}^{N}\left(\sum_{j=1}^{L_i-W+1}\mathbb{E}_{Z|X,\phi^{(t)}}[Z_{i,j}]\ln p(X_i|Z_{i,j}=1,\theta)\right) \\
&\qquad + \sum_{i=1}^{N}(1-\mathbb{E}_{Z|X,\phi^{(t)}}[Q_i])\ln p(X_i|Q_i=0,\theta) \\
&\qquad + \sum_{i=1}^{N}\mathbb{E}_{Z|X,\phi^{(t)}}[Q_i]\ln\left(\frac{\gamma}{L_i-W+1}\right) \\
&\qquad + \sum_{i=1}^{N}(1-\mathbb{E}_{Z|X,\phi^{(t)}}[Q_i])\ln(1-\gamma) \\
&= \sum_{i=1}^{N}\left(\sum_{j=1}^{L_i-W+1} Z_{i,j}^{(t)}\ln p(X_i|Z_{i,j}=1,\theta)\right) \\
&\qquad + \sum_{i=1}^{N}(1-Q_i^{(t)})\ln p(X_i|Q_i=0,\theta) \\
&\qquad + \sum_{i=1}^{N} Q_i^{(t)}\ln\left(\frac{\gamma}{L_i-W+1}\right) + \sum_{i=1}^{N}(1-Q_i^{(t)})\ln(1-\gamma). \qquad (4.40)
\end{aligned}$$

4.3.1 Generalised E-step

The new definition of λ is used in the generalisation of the E-step. The probability of the latent data $p(Z|X,\theta)$ is evaluated for each position:

$$Z_{i,j}^{(t)} = \frac{p(X_i|Z_{i,j}=1,\theta^{(t)})\frac{\gamma}{L_i-W+1}}{p(X_i|Q_i=0,\theta^{(t)})(1-\gamma^{(t)}) + \sum_{j=1}^{L_i-W+1} p(X_i|Z_{i,j}=1,\theta^{(t)})\frac{\gamma}{L_i-W+1}}, \qquad (4.41)$$

where (4.36) is substituted into (4.24) and (4.16) and (4.17) are used as required.

4.3.2 Generalised M-step

The M-step now requires maximising the generalised Q function (4.40) over θ and γ (note that as a result of the substitution (4.36), there are now no terms including λ).

Selecting the terms in $Q(\phi|\phi^{(t)})$ relying on θ, substituting (4.16) and (4.17) gives:

$$\sum_{i=1}^{N}\left(\sum_{j=1}^{L_i-W+1} Z_{i,j}^{(t)}\left\{\sum_{l\in\Delta_{i,j}}\sum_{k\in\mathcal{L}} I(X_{i,l}=k)\ln\theta_{0,k}+\sum_{m=1}^{W}\sum_{k\in\mathcal{L}} I(X_{i,j+m-1}=k)\ln\theta_{m,k}\right\}\right)$$
$$+\sum_{i=1}^{N}(1-Q_i^{(t)})\left\{\sum_{l=1}^{L_i}\sum_{k\in\mathcal{L}} I(X_{i,l}=k)\ln\theta_{0,k}\right\}. \tag{4.42}$$

where M is also replaced with L_i-W+1. The expected counts for each nucleotide $k\in\mathcal{L}$ at each position are defined as before (again, replacing m with L_i-W+1 and making use of the indicator variable Q_i):

$$N_{m,k}\triangleq\begin{cases}\sum_{i=1}^{N}\left[\sum_{j=1}^{L_i-W+1} Z_{i,j}^{(t)}\sum_{l\in\Delta_{i,j}} I(X_{i,l}=k)+(1-Q_i^{(t)})\sum_{l=1}^{L_i} I(X_{i,l}=k)\right], & m=0,\\ \sum_{i=1}^{N}\sum_{j=1}^{L_i-W+1} Z_{i,j}^{(t)}\, I(X_{i,j+m-1}=k), & m\neq 0.\end{cases} \tag{4.43}$$

Substituting (4.43) into (4.42) gives the expression from (4.29) and so the parameter updates for θ are therefore:

$$\theta_{m,k}^{(t+1)}=\frac{N_{m,k}}{\sum_{k\in\mathcal{L}} N_{m,k}}, \tag{4.44}$$

for $m\in\{0,\ldots,W\}$ and $k\in\mathcal{L}$, as before. The second term in $Q(\phi|\phi^{(t)})$ is now maximised over γ. From above, the generalised expression is:

$$\sum_{i=1}^{N} Q_i^{(t)}\ln\left(\frac{\gamma}{L_i-W+1}\right)+\sum_{i=1}^{N}(1-Q_i^{(t)})\ln(1-\gamma). \tag{4.45}$$

This is rearranged, substituting $S\triangleq\frac{1}{N}\sum_{i=1}^{N} Q_i^{(t)}$ as before to give:

$$NS\ln\left(\frac{\gamma}{L_i-W+1}\right)+N(1-S)\ln(1-\gamma). \tag{4.46}$$

Splitting the first log term and rearranging gives:

$$N(S\ln\gamma+(1-S)\ln(1-\gamma))-NS\ln(L_i-W+1). \tag{4.47}$$

Note that $(NS\ln(L_i-W+1))$ is invariant with respect to γ and so this term can be ignored for the purposes of maximisation. This leaves the same expression as before (4.34); performing the maximisation gives the same parameter updates (4.35).

4.4 MCOIN: a novel heuristic for determining TFBS motif width

As shown in the preceding sections, increasing the flexibility of the statistical model (in moving from the OOPS model to the ZOOPS model, then further generalising the

ZOOPS model) by altering the EM calculations allows greater practicality for applying motif discovery algorithms. However, a number of practical problems which cannot be solved by altering the EM calculations remain. One of the most important of these is automatically determining the width of a motif. In a 2010 study, two decades on from the first motif discovery algorithms, Li, *et al.* noted that "determining the actual motif length for a given set of motif-containing sequences remains an unsolved problem" [102]. In this section, a novel heuristic for automatically determining the width of a motif in PWM-based motif discovery algorithms, based on **M**otif **CO**ntainment and **IN**formation content (MCOIN) is validated. Based on tests with previously characterised prokaryotic TFBS motifs, MCOIN is shown to outperform the E-value of the resulting multiple alignment as a predictor of motif width, using mean absolute error. MCOIN is also shown to improve the overall correctness of results, based on receiver operating characteristic (ROC) analysis. Finally, it is shown that the performance of MCOIN will improve as the performance of the core motif discovery algorithm improves.

Automatically determining the width of a novel TFBS motif is a desirable property for motif discovery algorithms since the true motif width is generally not known *a priori*. An ideal algorithm would be executed over a range of reasonable candidate widths and return the most likely result based on some criterion. This is an important but challenging computational problem, as the likelihood function maximised by motif discovery algorithms cannot be used directly to compare models with different motif widths [98]. The difficulty partially stems from the fact that the maximum value of the joint likelihood of the model given the data and the latent data is bound to increase with increasing motif width as a consequence of the increasing number of free parameters [11, 90, 98]. The complexity of the problem is increased when additional constraints on the parameters (for example, the palindrome constraint in the popular MEME algorithm) are employed, as the maximum likelihood value of models with parameter constraints will be lower than unconstrained models of the same motif width. To some degree, this problem corresponds to the more general problem of model selection in statistics. A number of general model selection criteria which incorporate adjustments for model dimensionality (for example, the Akaike information criterion (AIC) [3] and the Bayesian information criterion (BIC) [146]) have been used in other areas with success. However, these criteria have generally not performed well at determining motif width in known datasets [98].

The complexity of the computational problem is further increased by the diver-

sity of TFBS motifs (Figure 3.1 illustrates this diversity in *E. coli* motifs). Clearly, biologists are interested in the true motif width; however, while some motifs provide statistically strong signals, the majority of motifs are more subtle and in the worst cases may be statistically indistinguishable from random artefacts in a given set of DNA sequences [9]. This subtlety means that the most statistically significant motif width need not match the biologically known true motif width. As an example, the true width of the FruR motif in *E. coli* is known to be 18nt. However, the sequence logo and known FruR binding sites (Figure 4.1) show that the outermost motif positions are very poorly conserved, providing little information above that of background 'noise'. Furthermore, motif discovery algorithms cannot guarantee to return the best possible result at each candidate width. Such algorithms often display a phenomenon known as 'shifting' (Figure 4.1), where a motif is only partially recovered, along with some additional non-motif 'background' positions [98]. This is in part due to the above fact that, from a statistical viewpoint, the true boundaries of a motif are often unclear. Although strategies to deal with this phenomenon have been devised, none can provide a guarantee that shifting is completely eliminated. This means that, even if the true motif width were known in advance, a motif discovery algorithm is not guaranteed to discover this motif perfectly. A heuristic which is robust in practice is therefore required. Such a heuristic should be able to cope with both cases where a statistically strong motif signal is present and where the motif signal is more subtle.

Attempts at a heuristic to automatically determine motif width in a deterministic (Expectation-Maximization, or EM-based) algorithm have included functions based on the maximum likelihood ratio test (LRT) [12], methods based on V-fold cross-validation [90] and the Bayesian information criterion (BIC) [21]. However, in practice, estimators based on the E-value of the resulting multiple alignment are used instead [9]. The E-value (described below) of the multiple alignment of predicted motif occurrences is an approximate p-value for testing the hypothesis that the predicted motif occurrences were generated from the predicted model against the null hypothesis that the predicted occurrences were generated by the background model. Typically, E-values are calculated for models at each candidate width and the model with the minimum E-value chosen.

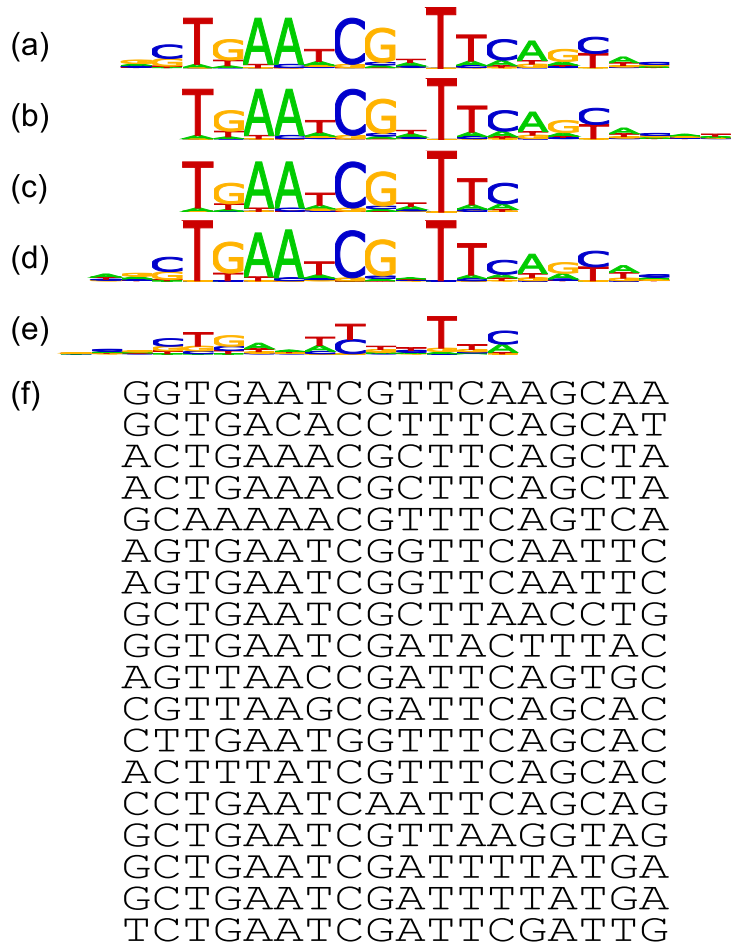

Figure 4.1: *E. coli* FruR motif sequence logos and occurrences: (a-e) Known and inferred *E. coli* FruR motif sequence logos. (a) The known *E. coli* FruR motif. The central part of the motif has a number of well-conserved positions; however, the outermost positions are very poorly conserved and may be incorrectly statistically regarded as background. A heuristic for determining the most likely width is required to be robust in statistically unclear situations such as this. (b) A motif discovery algorithm may become locked in a non-optimal local maximum of the likelihood function which corresponds to a shifted version of the true motif. (c) The most statistically significant model in a set of candidate models may only represent a portion of the true motif. (d) From the candidate set of computationally discovered models, MCOIN chooses the model at $W^* + 1$, which corresponds well with the true motif. (e) The E-values estimator chooses the model at $W^* - 3$, which corresponds less well with the true motif. (f) Known occurrences of the *E. coli* FruR motif. Note that logos (d) and (e) correspond to the discovered model at the chosen motif width and not to the known FruR occurrences (f). This explains why, for example, column 13 of model (e) contains a contribution from the nucleotide G despite the corresponding position in the known occurrences (11) always being nucleotide T.

4.4.1 Approach

The MCOIN heuristic is based on two orthogonal concepts: motif containment and mean information content per column. If it is assumed that the motif discovery algorithm discovers the true motif within the dataset as well as possible at every candidate width $\{W_{min}, \ldots, W_{max}\}$, then the algorithm discovers the true motif *exactly* at the *true* width W^*. It follows that, at candidate widths smaller than the true width (that is, $\{W_{min}, \ldots, W^* - 1\}$), only a portion of the true motif is discovered while at candidate widths larger than the true width (that is, $\{W^* + 1, \ldots, W_{max}\}$), the full motif is discovered, along with a number of background positions. Clearly, these models must be similar and are describing the same underlying motif. If it is known that the models for widths $W - 1$ and W are describing the same motif and also assumed that model selection criteria (for example, BIC) will favour the shorter model due to it having fewer free parameters, then the model with width $W - 1$ can be removed from the set of candidate models as the width-W model also describes the same motif.

Retaining the assumption that the motif discovery algorithm discovers the true motif as well as possible at every candidate width, it follows that the model at the true width W^* will also be removed as a result of it being contained within the model at width $W^* + 1$. The result of discarding models based only on containment would be to discard all but the longest model. Clearly, it would preferable to discard models at widths W_{min} to $W^* - 1$ in favour of the model at width W^*, but this model not to be discarded in favour of longer models. Calculating the mean information content per column (IC/col) for each model allows a method of stopping containment at widths greater than W^*. If, for example, the IC/col of the model at width W^* is B bits, the model at width $W^* + 1$ will have these same columns plus an additional background column, which will have a very low information content (if each nucleotide in the background model is equiprobable, the information content of this column will be 0 bits); the low information content of this additional background column will make the IC/col of the model at $W^* + 1$ less than B bits. The model selection process can therefore be modified, discarding a shorter model in favour of a longer model only if the shorter model is contained within the longer model *and* the IC/col of the longer model is similar to that of the shorter model.

At a high level, this is implemented as follows: the PWM of the shortest model (W_{min}) is tested against each longer model ($W_{min} + 1, \ldots, W_{max}$), calculating the Jensen-Shannon distance per column (JSD/col) for each comparison. The Jensen-Shannon

distance [59] is a modification of the Jensen-Shannon divergence [105] and is used as a measure of similarity; intuitively, the lower the JSD/col, the more similar the PWMs are. The IC/col ratio of the longer model to the shorter model is then calculated. If this is significantly lower than 1, it is assumed that the additional column in the longer model is not information-rich and the longer model is longer than the true motif width. If the shorter model is 'contained' within the longer model (that is, the minimum JSD/col is smaller than some similarity threshold t_{sim}, where $0 \leq t_{sim} \leq 1$) *and* the models have similar information (that is, the IC/col ratio of the longer model to the shorter model is greater than some information threshold t_{info}), the shorter model is removed from the set of candidate models. The process is repeated for model widths $W_{min}+1$ to $W_{max}-1$ (the longest model is always kept in the set of candidate models). The remaining model with the lowest BIC score is chosen as the best estimate of motif width.

4.4.2 Method

Predicting motif occurrences

The first step is to predict the positions of motif occurrences in the input sequences X. Following motif discovery at a particular width W, a model $\phi = \{\theta, \lambda\}$ is predicted, where $\theta = \{\theta_0, \theta_1\}$ represents the background (θ_0) and motif (θ_1) models and λ represents the prior probability that a given position within the input sequences is a motif occurrence[1]. Using the predicted model, a log-odds scoring matrix LO and threshold t may be calculated:

$$LO_{m,k} = \ln\left(\frac{\theta_{m,k}}{\theta_{0,k}}\right), \quad \text{for} m \in \{1,\ldots,W\}, k \in \mathcal{L} \tag{4.48}$$

$$t = \ln\left(\frac{1-\lambda}{\lambda}\right) \tag{4.49}$$

Together, LO and t form a Bayes-optimal classifier; each width-W subsequence x_i is scored (using Equation 4.50) and deemed to be a motif occurrence if $s(x_i) > t$ [12].

$$s(x_i) = \sum_{m=1}^{W} \sum_{k \in \mathcal{L}} LO_{m,k}\, I(k, x_{i,m}), \tag{4.50}$$

where $I(k, x_{i,m})$ is an indicator function which is 1 if and only if the nucleotide at $x_{i,m}$ is k and 0 otherwise and $x_{i,j}$ is the nucleotide in the jth position of sample x_i. Let x_{pred}

[1] This assumes the two-component mixture (TCM) model, which allows any number of non-overlapping motif occurrences.

be the set of non-overlapping predicted motif occurrences and n_{pred} be the number of predicted motif occurrences $|x_{pred}|$.

Calculating the BIC for candidate models

The motif discovery algorithm is run over a number of reasonable candidate widths and a model $\phi = \{\theta, \lambda\}$ is returned for each width. It is assumed that the unknown true motif width W^* is within the range of tested candidate widths, that is, $W_{min} \leq W^* \leq W_{max}$.

For each width $W \in \{W_{min}, \ldots, W^*, \ldots, W_{max}\}$, $\phi^{(W)}$ is used to create a set of *predicted* sites x_{pred}, as described above. For each width, the log likelihood of a particular model $\phi^{(W)}$ given the set of predicted sites can be calculated:

$$\ln L(\phi^{(W)}|x_{pred}) = \sum_{i=1}^{n_{pred}} \ln\left[p(x_i|\theta_1)\lambda + p(x_i|\theta_0)(1-\lambda)\right], \tag{4.51}$$

where the distributions for the motif and the background model (following [12]) are defined as:

$$p(x_i|\theta_1) = \prod_{m=1}^{W} \prod_{k \in \mathcal{L}} {\theta_{m,k}}^{I(k,x_{i,m})} \tag{4.52}$$

and

$$p(x_i|\theta_0) = \prod_{m=1}^{W} \prod_{k \in \mathcal{L}} {\theta_{0,k}}^{I(k,x_{i,m})}. \tag{4.53}$$

Following [146], the BIC for each model is calculated using:

$$-2\ln L(\theta, \lambda|x_{pred}) + P \cdot \ln(n_{pred}), \tag{4.54}$$

where P is the number of free parameters in the model, equal to $3(W+1)$. There now exists a set of models $\left\{\phi^{(W_{min})}, \ldots, \phi^{(W^*)}, \ldots, \phi^{(W_{max})}\right\}$; each model with its own BIC score, based on its log likelihood (calculated using its set of predicted sites) and the number of model parameters. MCOIN is now applied, as described in the next section.

MCOIN heuristic

The approach used by MCOIN is partly based on the similarity between two motif models of different widths. A measure of this similarity between two motif models is therefore required. As noted in Section 2.1, each column of the PWM can be interpreted as an independent multinomial distribution with four categories, each category giving the probability of a certain nucleotide appearing at a certain motif position. It follows that a measure of motif similarity can be obtained by comparing each column

in a given motif model θ_A with the corresponding column in another motif model θ_B; that is, measuring the similarity between the two multinomial distributions.

The Kullback-Leibler (KL) divergence[2] is a commonly used method for comparing two probability distributions, as it has some useful information theoretic properties [115] and gives a single figure for the similarity between the two distributions. For two probability distributions p and q, the KL divergence of q from p is defined in the discrete case as:

$$D_{\mathrm{KL}}(p||q) = \sum_i p(i)\log_2\frac{p(i)}{q(i)}. \tag{4.55}$$

However, there are some issues with using the KL divergence in this context. Perhaps most importantly, the KL divergence has no upper bound: the KL divergence $D_{\mathrm{KL}}(p||q) = 0$ when $p = q$ and tends towards infinity as p becomes the complement of q. The KL divergence is asymmetric (that is, in general, $D_{\mathrm{KL}}(p||q) \neq D_{\mathrm{KL}}(q||p)$) and does not satisfy the *triangle inequality* (that is, $D_{\mathrm{KL}}(x||z) \leq D_{\mathrm{KL}}(x||y) + D_{\mathrm{KL}}(y||z)$); for these reasons, it is known as a 'divergence' measure rather than a 'distance' measure. The KL divergence can also prove problematic when $p(i)$ or $q(i) = 0$ for some i (the special cases $0\log\frac{0}{x}$ and $x\log\frac{x}{0}$ are defined as 0 and ∞ respectively). This latter property is generally not an issue in the context of motif discovery, as the incorporation of pseudocounts into motif discovery algorithms eliminates any 0 entries. However, PWMs created from consensus sequences may well have 0 entries, depending on the data used to create them.

In cases such as this, the Jensen-Shannon (JS) divergence[3] [105], based on the KL divergence, is more appropriate as it is bounded in [0:1] (assuming the base 2 logarithm is used in the KL divergence) and can deal with 0 entries (both special cases reduce to $0\log\frac{0}{x}$, which is defined as 0, as before). Given two probability distributions p and q, the JS divergence of q from p is defined as:

$$D_{\mathrm{JS}}(p||q) = \frac{1}{2}D_{\mathrm{KL}}(p||m) + \frac{1}{2}D_{\mathrm{KL}}(q||m), \tag{4.56}$$

where the probability distribution m (known as the 'mixture' distribution) is defined as $m = \frac{1}{2}(p+q)$. The JS divergence $D_{\mathrm{JS}}(p||q) = 0$ when $p = q$ and tends towards 1 as p becomes the complement of q. Unlike the KL divergence, the JS divergence is symmetric by definition. While the JS divergence as defined in Equation 4.56 does not satisfy the triangle equality, the square root of the JS divergence does [59]. Taking the square root of the JS divergence therefore provides a distance function, or metric, in

[2]The KL divergence is often known as the *relative entropy*.

[3]The JS divergence is sometimes known as the *information radius*.

the mathematical sense[4]. This is known as the Jensen-Shannon distance. Based on the mathematical properties outlined above, the Jensen-Shannon distance is used as the measure of motif model similarity within MCOIN.

The approach used by MCOIN is also based on the information content of a motif model. The expression for the information content of a given motif position introduced by Schneider and Stephens [145] (page xvii) is extended to define the mean information content per column of a given motif model θ_1 as:

$$IC/col(\theta_1) = \frac{1}{W}\sum_{m=1}^{W}\sum_{k\in\mathcal{L}}\theta_{m,k}\log_2\left(\frac{\theta_{m,k}}{\theta_{0,k}}\right). \tag{4.57}$$

MCOIN relies on two threshold parameters, t_{sim} and t_{info}. The value of t_{sim} may be chosen to be anywhere between 0 and 1. Choosing a good value for t_{sim} is important. If this value is too small, smaller models are required to match longer models more exactly before being discarded. Therefore, fewer models are discarded and MCOIN tends to choose models of shorter widths, leading to an underestimation of the true motif width. In contrast, if the value of t_{sim} is too large, shorter models may be discarded in favour of longer models when they are dissimilar, leading to an overestimation of true motif width. The optimal value of t_{sim} was calculated using tests on the realistic synthetic data collection described in Chapter 3; root mean squared error was minimised at $t_{sim} = 0.32$. Tests using the previously characterised *E. coli* data described in Chapter 3 validated this parameter value: root mean squared error was minimised when $0.30 \leq t_{sim} \leq 0.32$. A value of $t_{sim} = 0.32$ is used here; this is reasonable as the value of t_{sim} should be kept low in order to ensure that two models are reasonably similar before discarding the shorter in favour of the longer. Tests which removed the motif discovery phase of the algorithm showed that the mean information content per column ratio alone was sufficient to choose the true motif width. That is, the value of t_{sim} had no effect. From this it can be concluded that, as motif predictions become stronger, the exact value of t_{sim} becomes less important. At current motif discovery algorithm performance levels, a value of 0.32 gives successful results with the data used in this study. However, it may be possible to change this value data-adaptively.

The second threshold parameter, t_{info}, is calculated based on a perfectly conserved motif model having a mean information content per column of 2 bits. The 'best case' background column is defined here as having an information content of 1 bit (equivalent to a PWM column such as $(0.5, 0.5, 0.0, 0.0)^T$, where any two nucleotides are equiprobable). It is then possible to calculate the 'best case' IC/col ratio between

[4]The square root of the JS divergence is also proportional to the Fisher information metric.

Assume that the width-12 model $\theta_1^{(12)}$ represents the motif to be discovered. It follows that models $\theta_1^{(i)}$ ($i < 12$) are 'sub-model's of $\theta_1^{(12)}$ and that $\theta_1^{(13)}$ is the same as $\theta_1^{(12)}$, but with 1 additional background column. Models $\theta_1^{(i)}$ ($i \leq 12$) will have an average IC/col of ~ 2 bits; however, the average IC/col of $\theta_1^{(13)}$ will be less. If the additional background column in $\theta_1^{(13)}$ is defined as having 1 bit of information as above, the *theoretical* IC/col ratio can be calculated:

$$IC/col(\theta_1^{(13)}) \approx \frac{1}{13}[12 \times 2 + 1 \times 1] = 1.9231 \text{ bits}$$

$$\frac{IC/col(\theta_1^{(13)})}{IC/col(\theta_1^{(12)})} \approx 0.9615$$

now define $t_{info(12||13)} = 0.9615$: if the *actual* IC/col ratio $\frac{IC/col(\theta_1^{(13)})}{IC/col(\theta_1^{(12)})}$ is less than 0.9615, it is assumed that the full motif is of width 12 and that the loss in information content is due to the addition of a background position. In contrast, calculating $\frac{IC/col(\theta_1^{(12)})}{IC/col(\theta_1^{(11)})}$ gives a value ~ 1; $\theta_1^{(11)}$ is therefore discarded in favour of $\theta_1^{(12)}$.

Example 4.1: Calculating the 'best case' IC/col ratio.

two models of any given widths. If the *actual* IC/col ratio is less than the calculated 'best case', the longer model is deemed to have unwanted background positions and the shorter model is not discarded in favour of the longer model. Example 4.1 illustrates how the 'best case' IC/col ratio is calculated. The calculation for the information threshold t_{info} can be generalised as:

$$t_{info(W_1||W_2)} = \frac{2W_1 + (W_2 - W_1)}{2W_2},\ W_2 > W_1. \tag{4.58}$$

This is equivalent to adding the required number of columns $W_2 - W_1$ at 1 bit/col. MCOIN is described in pseudocode in Algorithm 4.1.

E-value of the resulting multiple alignment

The E-value of the multiple alignment of predicted motif occurrences [80] is an approximate p-value for testing the hypothesis that the predicted motif occurrences were generated from the predicted model against the null hypothesis that the predicted occurrences were generated by the background model. The E-value is then an estimate of the expected number of multiple alignments with statistical significance as great or greater than the observed alignment. Briefly, the E-value is calculated by com-

procedure MCOIN

Define the similarity threshold t_{sim}.

for $a = W_{min}$ **to** $W_{max} - 1$ **do**

for $b = a + 1$ **to** W_{max} **do**

Calculate information threshold $t_{info(a||b)} = \frac{2a+(b-a)}{2b}$

for *offset* $= 0$ **to** $b - a$ **do**

Calculate similarity (mean Jensen-Shannon distance per column) using:

$$sim = \frac{1}{a} \sum_{j=1}^{a} \sqrt{D_{\mathrm{JS}}(f_j^{(a)} || f_{j+offset}^{(b)})},$$

Calculate mean information content per column ratio using:

$$inf = \frac{IC/col(\theta_1^{(b)})}{IC/col(\theta_1^{(a)})}.$$

Remove $\phi^{(a)}$ from the set of candidate models IF:

$$sim < t_{sim} \quad \text{AND} \quad inf > t_{info(a||b)}.$$

end

end

end

Return the remaining model with the lowest BIC as the best estimate of motif width.

end MCOIN

Algorithm 4.1: Pseudocode outlining the MCOIN heuristic.

puting the log-likelihood ratio of each column of the resulting multiple alignment of predicted sites and computing the p-value for each based on the background model. The p-value of the product of column p-values is computed as described by Bailey and Gribskov [13], then multiplied by the number of possible ways to select positions for the given number of sites in the set of input sequences to give the E-value. The E-value is calculated for models at each candidate width and minimised to select the best estimate of motif width [9, 21]. Calculating E-values for a number of different motif widths raises the issue of correcting the calculated E-values for multiple tests. However, methods for dealing with multiple tests such as the Bonferroni correction or the Benjamini-Hochberg procedure are not applied in this case, since it is the minimum E-value which is of interest, rather than the actual calculated value.

4.4.3 Results and Discussion

MCOIN was evaluated quantitatively using a mixture of realistic synthetic and previously characterised *E. coli* and diverse prokaryotic data. The datasets were constructed as described in Chapter 3. The results presented in Tables 4.1, 4.3, 4.5 and 4.7 show that, in general, mean site-level sensitivity (*sSn*) and positive predictive value (*sPPV*) decrease with decreasing motif conservation. The decrease in *sSn* is a result of the motif discovery algorithm predicting fewer sites in total. That is, at lower motif conservations, fewer sites score highly enough such that $s(x_i) > t$ (see Equation 4.50). This leads to an increase in the number of false negative results (sites incorrectly classified as 'background') and therefore a decrease in *sSn*. The decrease in *sPPV* is attributable to background sites better matching the weaker motif model; as the model becomes weaker, the difference in scores between true motif occurrences and spurious background sites decreases. This can lead to an increase in the number of false positive results (sites incorrectly classified as motif occurrences) and therefore a decrease in *sPPV*.

Realistic synthetic data: width determination without motif discovery

MCOIN was initially evaluated on realistic synthetic data (as described in Section 3.1) without the motif discovery phase of the algorithm. That is, for each dataset, the heuristic was tested using a set of candidate models which were constructed as if the motif discovery algorithm had discovered the motif in that dataset as well as possible at each candidate width. For each dataset, all candidate widths from $W^* - 4$ to $W^* + 4$ were tested. MCOIN is compared against the E-values estimator and also (following [21]) evaluations using the known width (equivalent to having a set of candidate models consisting only of W^*). Results of these evaluations are summarised in Tables 4.1 and 4.2.

Note from Table 4.2 that the width predicted by MCOIN closely matches the true width in almost all cases; the error in the predicted width increases slightly as mean motif conservation is decreased. The E-values estimator initially matches MCOIN but quickly begins to underestimate motif width, leading to a much larger increase in the error in predicted width. MCOIN shows a clear performance advantage in terms of predicted width at all conservation levels.

Given that the widths predicted by MCOIN generally match the known width, it is unsurprising that the classification-based results (Table 4.1) match those in the case

Conservation	Known width (W^*)			MCOIN ($W^*\pm4$)			E-values ($W^*\pm4$)		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
2.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
1.49	**0.98**	**0.94**	**1.00**	**0.98**	**0.94**	**1.00**	0.97	0.93	**1.00**
1.08	0.80	**0.93**	**1.00**	0.80	**0.93**	**1.00**	**0.82**	0.79	**1.00**
0.76	0.49	**0.89**	**0.99**	0.49	**0.89**	**0.99**	**0.56**	0.71	**0.99**
0.51	**0.23**	**0.79**	**0.99**	**0.23**	**0.79**	**0.99**	**0.23**	0.77	0.98

Table 4.1: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for five collections of realistic synthetic data at varying levels of motif conservation. Best results are printed in bold. In these tests, the motif discovery phase of the algorithm was removed and the set of candidate models constructed as if the motif discovery algorithm had performed as well as possible at each candidate width.

Conservation	MCOIN ($W^*\pm4$)		E-values ($W^*\pm4$)	
(mean bits/col)	MAE	RMSE	MAE	RMSE
2.00	0.00	0.00	0.00	0.00
1.49	**0.00**	**0.00**	0.12	0.50
1.08	**0.00**	**0.06**	1.55	1.84
0.76	**0.01**	**0.09**	1.79	2.04
0.51	**0.07**	**0.39**	3.33	3.60

Table 4.2: Mean absolute error (MAE) and root mean squared error (RMSE) for five collections of realistic synthetic data at varying levels of motif conservation. Best results are printed in bold. In these tests, the motif discovery phase of the algorithm was removed and the set of candidate models constructed as if the motif discovery algorithm had performed as well as possible at each candidate width.

where the width is known. As noted above, *sSn* decreases with decreasing motif conservation. A similar, but less sharp, decrease is seen in *sPPV*. Although the E-values estimator slightly outperforms MCOIN in terms of *sSn* for the data collections with mean motif conservation of 1.08 bits/col and 0.76 bits/col (0.82 compared to 0.80 and 0.56 compared to 0.49, respectively), the corresponding values of *sPPV* are outperformed by MCOIN (0.93 compared to 0.79 and 0.89 compared to 0.71, respectively). Combining these results with the results presented in Table 4.1, this is likely a result of the E-values estimator choosing models at non-optimal widths which predict more sites overall at the expense of more false positive predictions.

Realistic synthetic data: width determination with motif discovery

Subsequent evaluations use models discovered by an EM-based algorithm using the TCM model; the motif discovery phase of the algorithm is run as it would normally. Again, for each estimator, all candidate widths from $W^* - 4$ to $W^* + 4$ are tested. Results of evaluations on each of the five data collections are summarised in Tables 4.3 and 4.4; detailed results for Table 4.3 are provided in Table B.1 in Appendix B.

Note that the results for predictions at the known width are generally lower than when the motif discovery phase of the algorithm was removed. These results illustrate the fact that the core motif discovery algorithm is far from perfect: even when the true motif width is known, performance in terms of *sSn* and *sPPV* may be low. In all data collections, both MCOIN and the E-values estimator are shown to have a performance similar to or better than that at the known width in terms of classification-based measures. As noted by [21], this may be attributed to the fact that predicted sites are only required to overlap the known site by a quarter in order to be counted as a true positive.

As noted above, results for all three classification-based measures generally decrease as mean motif conservation also decreases (Table 4.3). At higher levels of motif conservation, MCOIN is shown to outperform the E-values estimator in terms of *sSn*. In this test, MCOIN generally chooses models which increase *sSn*, at the expense of *sPPV*. That is, MCOIN chooses models which tend to predict more false positive sites. While it would be preferable to have fewer false results (that is, higher values for both *sSn* and *sPPV*) overall, it may be preferable to increase *sSn* at the expense of *sPPV*. For example, when searching for putative binding sites to be verified experimentally, it may be more useful to have more false positives than false negatives. The E-values estimator is shown to achieve a higher *sPPV* in all cases; this matches the findings of [21], where the E-values estimator was shown to achieve a slightly higher *sPPV* than

Conservation	Known width (W^*)			MCOIN ($W^*\pm4$)			E-values ($W^*\pm4$)		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
2.00	0.84	0.25	0.99	**0.93**†‡	0.42†	**1.00**†‡	0.91†	**0.79**$^{\dagger *}$	0.99†
1.49	0.26‡	0.07	0.98	**0.28**†‡	0.15†	**0.99**†‡	0.21	**0.45**$^{\dagger *}$	0.98†
1.08	**0.02***‡	0.01	**0.96**	0.01	0.01†	**0.96**†‡	0.01*	**0.23**$^{\dagger *}$	**0.96**
0.76	0.00	0.00	**0.94***	0.00	0.00	0.93	0.00	**0.12**†	**0.94***
0.51	0.00	0.00	**0.93***	0.00	0.00	**0.93**	0.00	**0.09**†	**0.93***

Table 4.3: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for five collections of realistic synthetic data at varying levels of motif conservation. Best results are printed in bold. In these tests, the motif discovery algorithm was allowed to run as it would normally. Results marked † are statistically significant with regard to the known width, results marked * are statistically significant with regard to the MCOIN heuristic and results marked ‡ are statistically significant with regard to the E-values estimator, all at $p \leq 0.05$ (see main text).

Conservation	MCOIN ($W^*\pm4$)		E-values ($W^*\pm4$)	
(mean bits/col)	MAE	RMSE	MAE	RMSE
2.00	**1.60**	**2.06**	1.80	2.28
1.49	**1.59**	**2.08**	2.46	2.82
1.08	**1.97**	**2.42**	2.16	2.51
0.76	2.38	2.74	**1.84**	**2.22**
0.51	2.38	2.71	**1.95**	**2.32**

Table 4.4: Mean absolute error (MAE) and root mean squared error (RMSE) for five collections of realistic synthetic data at varying levels of motif conservation. Best results are printed in bold. In these tests, the motif discovery algorithm was allowed to run as it would normally.

other estimators on datasets containing human TFBS motifs. At higher levels of motif conservation, MCOIN is also shown to outperform the E-values estimator in terms of AUC.

While MCOIN generally matches the E-values estimator in terms of overall correctness based on AUC values, this does not represent the full picture. It follows from the above that an estimator may appear to perform well even if the chosen width does not match the true width [21]. Errors in the predicted width are presented in Table 4.4. It is noted from these results that the error in width predicted by both estimators generally increases as mean motif conservation is decreased. However, at higher levels of motif conservation, MCOIN outperforms the E-values estimator using both error measures.

Statistical significance tests for the results in this section were performed using a paired one-sided Wilcoxon signed rank test[5]. The majority of results marked as being statistically significant in Table 4.3 are significant with p-values $< 2.20 \times 10^{-16}$. The *sSn* result for the E-values estimator on the group with mean conservation 1.08 bits/col was significantly higher than that of MCOIN ($p = 7.05 \times 10^{-7}$). The AUC results for MCOIN were significantly higher than those of the E-values estimator for the groups with mean conservation 2.00, 1.49 and 1.08 bits/col ($p = 2.26 \times 10^{-7}$, $p = 6.66 \times 10^{-12}$ and $p = 5.09 \times 10^{-9}$, respectively). The AUC result for MCOIN for the group with conservation 1.08 bits/col was significantly higher than that for the known width ($p = 7.22 \times 10^{-6}$). The AUC results for the known width and the E-values estimator on the groups with mean conservation 0.76 and 0.51 bits/col were significantly higher than those for MCOIN ($p = 2.86 \times 10^{-3}$ and $p = 5.55 \times 10^{-4}$, respectively for the known width and $p = 8.22 \times 10^{-7}$ and $p = 1.18 \times 10^{-6}$, respectively for the E-values estimator).

E. coli and prokaryotic ChIP data

MCOIN was then evaluated in the same manner using the previously characterised *E. coli* TFBS sequences described in Section 3.2 and the diverse prokaryotic TFBS sequences as described in Section 3.3. The results of this evaluation are summarised in Tables 4.5-4.8. Detailed results for Tables 4.5 and 4.7 are provided in Tables B.2 and B.3 respectively in Appendix B.

The heterogeneity of the motifs in both data collections may suggest that results

[5] p-values smaller than the machine epsilon in R (2.20×10^{-16}) are presented as '$< 2.20 \times 10^{-16}$'.

Conservation	Known width (W^*)			MCOIN ($W^*\pm4$)			E-values ($W^*\pm4$)		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
'high' (1.36)	**0.81***	0.22‡	**0.96**	0.72	**0.29**‡	**0.96**	0.70	0.17	0.95
'low' (0.78)	0.63	0.41	0.96	**0.69**	**0.51**‡	**0.98**	0.66	0.32	0.97
overall (1.13)	**0.74**	0.30‡	**0.96**	0.71	**0.38**‡	**0.96**‡	0.68	0.23	**0.96**

Table 4.5: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for 20 datasets created using real *E. coli* data. Best mean results are printed in bold. Results marked * are significant with regard to the MCOIN heuristic and results marked ‡ are significant with regard to the E-values estimator, all at $p \leq 0.05$ (see main text).

Conservation	MCOIN ($W^*\pm4$)		E-values ($W^*\pm4$)	
(mean bits/col)	MAE	RMSE	MAE	RMSE
'high' (1.36)	**2.08**	**2.43**	2.92	3.12
'low' (0.78)	**1.75**	**2.06**	3.00	3.20
overall (1.13)	**1.95**	**2.29**	2.95	3.15

Table 4.6: Mean absolute error (MAE) and root mean squared error (RMSE) for 20 datasets created using real *E. coli* data. Best results are printed in bold.

Conservation	Known width (W^*)			MCOIN ($W^*\pm4$)			E-values ($W^*\pm4$)		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
0.99	**0.75**	0.67	**0.99**	**0.75**	**0.68**	**0.99**	0.73	0.67	**0.99**

Table 4.7: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for 9 datasets created using real prokaryotic data determined through ChIP experiments. Best results are printed in bold. The datasets used are summarised in Table 3.2.

Conservation	MCOIN ($W^*\pm4$)		E-values ($W^*\pm4$)	
(mean bits/col)	MAE	RMSE	MAE	RMSE
0.99	**1.44**	**1.86**	2.33	2.73

Table 4.8: Mean absolute error (MAE) and root mean squared error (RMSE) for 9 datasets created using real prokaryotic data determined through ChIP experiments. Best results are printed in bold. The datasets used are summarised in Table 3.2.

for these datasets could be equally varied. However, both MCOIN and the E-values estimator are reasonably robust in terms of predicted sites (Tables 4.5 and 4.7). While the *sSn* results for the low conservation group in the *E. coli* data collection are lower than that for the high conservation group, *sPPV* increases with decreasing motif conservation. This is a result of the smaller set of predicted sites containing fewer false positive results and can be attributed to the small number of datasets tested. When combined, the reduction in the number of false positive predictions and the consistently high AUC values suggest that models are chosen where true motif occurrences are predicted with greater confidence. For both the *E. coli* and prokaryotic ChIP data collections, MCOIN outperforms the E-values estimator in terms of classification-based results. The prokaryotic ChIP data collection shows a slight improvement in *sSn* and *sPPV* values (Table 4.7); this improvement is greater in the *E. coli* data collection (Table 4.5). It is also noted that the classification-based results for MCOIN better match those at the known width than the results of the E-values estimator.

The MCOIN *sPPV* results for all three groups of *E. coli* results (Table 4.5) are shown to be significantly higher than those of the E-values estimator ($p = 1.82 \times 10^{-2}$, $p = 1.12 \times 10^{-2}$ and $p = 4.66 \times 10^{-4}$ for the 'high conservation', 'low conservation' and 'overall' groups, respectively). The MCOIN AUC result for the 'overall' group is also significantly higher than that of the E-values estimator ($p = 2.92 \times 10^{-2}$). The *sSn* result for the 'high conservation' group at the known width is significantly higher than the corresponding MCOIN result ($p = 2.96 \times 10^{-2}$). Finally, the *sPPV* results for the 'high conservation' and 'overall' groups at the known width are significantly higher compared to those of MCOIN ($p = 2.05 \times 10^{-2}$ and $p = 3.93 \times 10^{-3}$, respectively).

Tables 4.6 and 4.8 present the mean error in motif width based on both data collections. MCOIN is shown to outperform the E-values estimator for both data collections. Although the mean error in motif width for models predicted by MCOIN appears to decrease with decreasing motif conservation in the *E. coli* data collection, this is explained by the small number of datasets tested. The small number of datasets tested also accounts for the fact that the error in motif widths predicted by the E-values estimator is relatively high for both real data collections, given the results previously obtained on realistic synthetic data.

As noted above, performance in terms of AUC may be improved by choosing a better motif model at a non-optimal width. The *E. coli* RcsB motif provides an example of this (Figure 4.2 illustrates some of the observations made here). At the true width ($W^* = 14$nt), the motif is discovered relatively poorly (*sSn* = 0.27, *sPPV* = 0.21, AUC

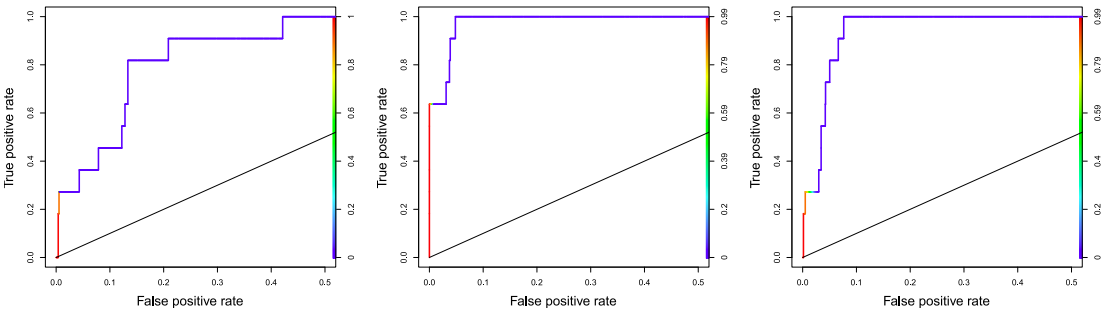

Figure 4.2: *E. coli*: RcsB motif ROC curves: ROC curves (plotted for $0 \leq sFPR \leq 0.5$) for the most likely *E.coli* RcsB motif, as chosen using the known width (left), MCOIN (centre) and E-values based estimator (right). The curve colour illustrates the threshold of $p(Z_{i,j} = 1|X_{i,j}, \theta)$, from 1.0 (red) to 0.0 (blue). Although MCOIN and the E-values estimator both underestimate the known motif width, site-level predictions are improved as the true motif is relatively weakly discovered at the true width. Performance in terms of AUC may be increased by choosing stronger and/or unshifted motif models at non-optimal widths. MCOIN displays improvement over the known motif width and the E-values estimator in all three classification performance measures.

= 0.88). Both MCOIN and the E-values based estimator improve AUC by choosing models at shorter widths. The E-values estimator chooses the model at $W^* - 2$ (sSn = 0.27, $sPPV$ = 0.09, AUC = 0.97) and MCOIN chooses the model at $W^* - 4$ (sSn = 0.64, $sPPV$ = 0.39, AUC = 0.99). MCOIN displays improvement in all measures; it may be concluded that, although the chosen width is not the true motif width, the model chosen by MCOIN is a better model overall. Similar results are noted in the CaiF, FruR and PurR motifs in the *E. coli* data collection and the *B. subtilis* Spo0A motif in the prokaryotic ChIP data collection. As noted above, the model at the optimal width need not be the closest match to the biologically known motif. The results presented in Figure 4.2 also show that the model chosen by MCOIN gives more predictions at higher values of $p(Z_{i,j} = 1|X_{i,j}, \theta)$, compared to the model chosen by the E-values estimator and the model at the true width. Similar results are noted for some other *E. coli* motifs, although this cannot be guaranteed for all motifs.

Comparing Tables 4.5 and 4.7 to Table 4.3, MCOIN is shown to give excellent classification-based results (particularly on the prokaryotic ChIP datasets) given the overall mean motif conservation and the results on realistic synthetic data. This is due to the conservation of individual positions within each motif: while the conservation of positions in each synthetic motif is uniform and independent, this pattern of conser-

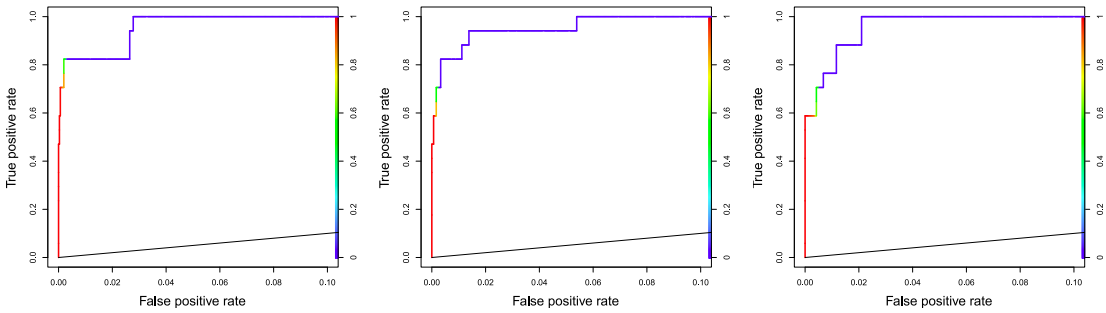

Figure 4.3: *E. coli*: GntR motif ROC curves: ROC curves (plotted for $0 \leq sFPR \leq 0.1$) for the most likely *E.coli* GntR motif, as chosen using the known width (left), MCOIN (centre) and E-values based estimator (right). The curve colour illustrates the threshold of $p(Z_{i,j} = 1|X_{i,j}, \theta)$, from 1.0 (red) to 0.0 (blue). All three estimators predict the GntR motif much better than expected, considering the low conservation of the motif and the results of the experiments using realistic synthetic data.

vation is not mirrored in real TFBS motifs. Analysis of the previously characterised motifs used in this study indicates that motifs with low mean conservation may have several positions which are very well or even perfectly conserved. This matches well with previous studies [56], which noted that the conservation of a given motif position is correlated with the conservation of surrounding motif positions, producing clusters of well-conserved positions, which may aid TFBS motif discovery algorithms. This phenomenon is clear in a number of *E. coli* motifs, particularly GntR (Figure 4.3), which has a mean conservation of 0.74 bits/col; the synthetic data results suggest relatively low values of *sSn* and *sPPV* for this motif. However, the GntR motif has a cluster of reasonably well-conserved positions, with a maximum conservation of 1.61 bits/col and is discovered well at the known width (*sSn* = 0.82, *sPPV* = 0.70, AUC = 0.99), with similar results for both the MCOIN and E-values based estimators (*sSn* = 0.71, *sPPV* = 0.71, AUC = 0.99 and *sSn* = 0.71, *sPPV* = 0.48, AUC = 1.00, respectively).

4.4.4 Conclusions

Determining the width of a TFBS motif is an important and challenging problem with direct relevance to computational motif discovery. MCOIN is a novel heuristic for determining the width of a motif, based on motif containment and information content. Results of tests on two data collections of previously characterised prokaryotic motifs show that MCOIN outperforms the E-value of the resulting multiple alignment (currently the most widely used estimator) as a predictor of motif width, using mean

absolute error and root mean squared error. MCOIN is also shown to choose models which improve the overall correctness of predicted motif sites, based on site-level sensitivity, positive predictive value and the area under the ROC curve.

MCOIN also has a clear advantage over methods based on cross-validation with limited numbers of folds, as all available data is used for motif discovery, improving discovery results. Further, the results of experiments which removed the motif discovery phase of the algorithm show that, as the performance of this phase improves, the performance of MCOIN as a predictor of motif width also improves: as the discovered model becomes stronger and better models the true motif, the error in the width estimated by MCOIN will decrease.

Chapter 5

MITSU: a novel stochastic and entropy-based Expectation-Maximisation algorithm for motif discovery

This chapter presents a novel algorithm for motif discovery based on stochastic EM (sEM). This algorithm is used to evaluate the primary hypothesis, that transcription factor binding site discovery using stochastic EM improves on deterministic EM-based approaches, in terms of previously established metrics. The chapter begins with a high-level explanation of how sEM can be used to solve the motif discovery problem. The derivations in Section 5.1 provide a confirmation of the equivalence of sequence models in previous studies. This leads to the first major contribution of this chapter: a set of generalised expressions which implement the ZOOPS sequence model in the context of sEM (Section 5.2). Section 5.3 presents the second major contribution of this chapter: a novel stochastic EM-based algorithm for motif discovery named MITSU. Section 5.4 evaluates MITSU quantitatively on realistic synthetic data and previously characterised prokaryotic data and discusses the results in relation to the primary hypothesis of the thesis.

The general stochastic EM algorithm is introduced in Section 2.2.3. As with the deterministic EM algorithm, initial values for the model parameters are estimated, then two steps repeatedly carried out. The sEM algorithm replaces the expectation step of deterministic EM with a sampling step, known as the S-step. In this step, the probability of the latent data given the observed data and the current parameter estimates

is computed and a pseudosample simulated. The update step, or U-step, of the algorithm (equivalent to the M-step in deterministic EM) updates the model parameters; however in sEM this update is based on the pseudo-complete sample (that is, the observed data and the pseudosample). The S-step and the U-step are repeated until the parameter values converge. Detecting convergence of the algorithm and choosing an appropriate stopping rule is more difficult for stochastic EM than deterministic EM; how this is achieved in MITSU is described in Section 5.3.4. The sEM algorithm is used in the context of motif discovery (using the OOPS sequence model) as follows: The initial values for the motif model are estimated (as before, either by consensus model or maximum likelihood). For each input sequence, the S-step of the algorithm calculates the probability of each width-W subsequence being an occurrence of the motif, based on the current model parameters. One subsequence is then sampled based on these probabilities. This procedure can be viewed as estimating the position of the motif within that input sequence. Note that since, in the context of motif discovery, sEM effectively makes a hard assignment of each width-W subsequence to one of the mixture components, this step designates the sampled position as a motif occurrence and the remaining positions as background positions. Once each input sequence has been sampled, the U-step updates the motif model parameters by taking a consensus of the sampled positions. These two steps are repeated iteratively until convergence.

5.1 Equivalence of OOPS sequence model expressions

The following derivation confirms that, despite notational differences, the expressions for deterministic EM-based motif discovery are the same in the work of Bailey and Elkan [10] and Bi [23]. Bi extends his deterministic EM expressions for the OOPS model to stochastic EM; therefore, a line may be traced from Bailey and Elkan's original deterministic EM expressions to Bi's stochastic EM expressions, which form the basis for the algorithm developed in this thesis.

Bi uses a different notation to that used by Bailey and Elkan. The differences between the two methods are noted here. In Bi's notation, $[A,a]$ refers to the (latent) alignment (equivalent to $[Z,z]$ in the current notation) and $[S,s]$ refers to the (observed) sequence data (equivalent to $[X,x]$ in the current notation). Bi represents the model θ as Θ and also uses the notation $a_i = l$ as an indicator variable, equivalent to $A_{i,l} = 1$.

Latent data ($Z_{i,j}^{(t)}$)

Bailey and Elkan define the expectation of the latent data (at iteration t) for the OOPS model as:

$$Z_{i,j}^{(t)} \triangleq p(Z_{i,j}=1|X_i,\theta^{(t)}) = \frac{p(X_i|Z_{i,j}=1,\theta^{(t)})}{\sum_{l=1}^{M} p(X_i|Z_{i,l}=1,\theta^{(t)})}, \tag{5.1}$$

for all $i \in \{1,\ldots,N\}$ and $j \in \{1,\ldots,M\}$. As before, M represents the number of possible width-W subsequences in an input sequence. Bi defines the expectation of the latent data as:

$$p(a_i=l|S_i,\Theta^{(t)}) = \frac{p(S_i|a_i=l,\Theta^{(t)}))}{\sum_{j=1}^{L_i-W+1} p(S_i|a_i=j,\Theta^{(t)}))}. \tag{5.2}$$

Simply interchanging Bi's notation with Bailey and Elkan's notation shows that Equations (5.1) and (5.2) are equivalent.

Conditional probability for a sequence

Bailey and Elkan define the conditional sequence probability for sequences containing a motif (all sequences in the OOPS model) as:

$$p(X_i|Z_{i,j}=1,\theta) \triangleq \prod_{l\in\Delta_{i,j}} \prod_{k\in\mathcal{L}} \theta_{0,k}^{I(X_{i,l}=k)} \prod_{m=1}^{W} \prod_{k\in\mathcal{L}} \theta_{m,k}^{I(X_{i,j+m-1}=k)} \tag{5.3}$$

This can be viewed as the product of two terms: the first term calculates probabilities for each position outside the motif ($\Delta_{i,j}$ is the set of positions in X_i which lie outside the motif when the motif starts at position j), while the second term calculates the probability for the motif sequence.

Bi defines the probability as:

$$p(S_i|a_i=l,\Theta) = \prod_{y\in A_{i,l}^c} \prod_{k\in K} \theta_{0k}^{I(S_{i,y}=k)} \prod_{m=1}^{W} \prod_{k\in K} \theta_{mk}^{I(S_{i,l+m-1}=k)}. \tag{5.4}$$

Again, interchanging Bi's notation with Bailey and Elkan's notation (note that K is the set of nucleotide letters $\{A,C,G,T\}$ and $A_{i,l}^c$ denotes the complement of the motif positions, that is, the background positions) shows that these expressions are equivalent.

Joint (log) likelihood function

Further confirmation of the equivalence is given by the joint (log) likelihood function for the OOPS sequence model. This is defined by Bailey and Elkan as:

$$\ln p(X,Z|\theta) = \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{i,j} \ln p(X_i|Z_{i,j}=1,\theta) + N \ln \frac{1}{M}, \tag{5.5}$$

This can be shown to be equivalent to the Q function as defined by Bi:

$$Q(\Theta;\Theta^{(t)}) = \sum_{i=1}^{N}\sum_{l=1}^{L_i-W+1} p(a_i=l|S_i,\Theta^{(t)}) \ln p(S_i|a_i=l,\Theta) p_0(a_i|\Theta) \tag{5.6}$$

Firstly note that the site prior $p_0(a_i|\Theta)$ is defined as uniform over all width-W subsequences in input sequence i; that is,

$$p_0(a_i|\Theta) = p_0(a_i) = \frac{1}{L_i - W + 1}. \tag{5.7}$$

Rewriting with the current notation and substituting Equation (5.1), this becomes:

$$\sum_{i=1}^{N}\sum_{j=1}^{M} Z_{i,j}^{(t)} \ln p(X_i|Z_{i,j}=1,\theta) \frac{1}{M}. \tag{5.8}$$

Splitting the log term and rearranging gives:

$$\sum_{i=1}^{N}\sum_{j=1}^{M} Z_{i,j}^{(t)} \ln p(X_i|Z_{i,j}=1,\theta) + Z_{i,j}^{(t)} \ln \frac{1}{M}. \tag{5.9}$$

Since $\ln \frac{1}{M}$ does not rely on Z, this may be summed out: note that $\sum_j Z_{i,j} = 1$, therefore $\sum_{i,j} Z_{i,j} = N$. Carrying out this final substitution derives Bailey and Elkan's expression (Equation 5.5).

A density for Monte Carlo sampling

Finally, it can be shown that the density used by Bi can be derived from the above expressions. Therefore, a line can be traced from the original EM definitions due to Bailey and Elkan through to the sEM expressions used by Bi. This is important, for two reasons. Firstly, it confirms the correctness of the expressions used in SEAM. Secondly, the method for deriving the OOPS model sEM expressions will be used in this thesis as a template for deriving sEM expressions that implement the ZOOPS sequence model. The idea underlying SEAM [23] and MCEMDA [24] is to replace the computation and maximisation of $Q(\theta|\theta^{(t)})$ by the much simpler computation of

$p(Z_{i,j} = 1|X_i, \theta^{(t)})$, drawing a number[1] of samples $Z^{(t)}$ (S-step), followed by an update to θ based on the pseudo-complete samples $(X, Z^{(t)})$ (U-step).

A suitable density to represent an input sequence X_i is required. Bi substitutes Equation (5.4) into Equation (5.2), which results in the expression:

$$p(a_i = l|S_i, \Theta^{(t)}) = \frac{\prod_{y \in A^c_{i,l}} \prod_{k \in K} \theta_{0k}^{I(S_{i,y}=k)} \prod_{m=1}^{W} \prod_{k \in K} \theta_{mk}^{I(S_{i,l+m-1}=k)}}{\sum_{j=1}^{L_i-W+1} \left\{ \prod_{y \in A^c_{i,l}} \prod_{k \in K} \theta_{0k}^{I(S_{i,y}=k)} \prod_{m=1}^{W} \prod_{k \in K} \theta_{mk}^{I(S_{i,j+m-1}=k)} \right\}}. \tag{5.10}$$

To ease computation, Equation (5.10) may be divided through by the common (background) term $\prod_{y \in A^c_{i,l}} \prod_{k \in K} \theta_{0k}^{I(S_{i,y}=k)}$ to get the target density for the OOPS model:

$$p(a_i = l|S_i, \Theta^{(t)}) = \frac{\prod_{m=1}^{W} \prod_{k \in K} \left(\frac{\theta_{mk}^{(t)}}{\theta_{0k}^{(t)}} \right)^{I(S_{i,l+m-1}=k)}}{\sum_{j=1}^{L_i-W+1} \left\{ \prod_{m=1}^{W} \prod_{k \in K} \left(\frac{\theta_{mk}^{(t)}}{\theta_{0k}^{(t)}} \right)^{I(S_{i,j+m-1}=k)} \right\}}. \tag{5.11}$$

Bi also suggests that the accuracy of the computation may be increased by taking logs. Doing this and converting to the current notation yields:

$$p(Z_{i,j} = 1|X_i, \theta^{(t)}) = \exp \left[\sum_{m=1}^{W} \sum_{k \in \mathcal{L}} \left\{ I(X_{i,j+m-1} = k) \ln \left(\frac{\theta_{m,k}^{(t)}}{\theta_{0,k}^{(t)}} \right) \right\} \right] \Big/ \Phi(i), \tag{5.12}$$

where $\Phi(i)$ is a normalising factor such that $\sum_{j=1}^{M} p(Z_{i,j} = 1|X_i, \theta^{(t)}) = 1$.

5.2 Defining expressions for stochastic Expectation-Maximisation

This section presents the first novel result: a set of generalised expressions which implement the ZOOPS ('zero or one occurrences per sequence') sequence model in the context of stochastic EM for motif discovery. This builds upon the work of Bi; the OOPS ('one occurrence per sequence') model presented in [23] is generalised here and rewritten in a consistent notation. The OOPS model is then extended naturally to the ZOOPS model, based on Bailey and Elkan's original definitions and using a similar approach to that adopted by Bi.

[1] In SEAM, one sample (equivalent to a full alignment) is drawn, following the sEM algorithm. In MCEMDA, based on the MCEM algorithm, multiple samples are drawn.

5.2.1 OOPS model stochastic EM expressions

The SEAM algorithm replaces the computation and maximisation of $Q(\theta|\theta^{(t)})$ with the computation of $p(Z_{i,j} = 1|X_i, \theta^{(t)}) \equiv Z_{i,j}^{(t)}$, drawing a sample from each input sequence (S-step), followed by an update to θ based on the pseudo-complete samples (U-step).

S-step

A suitable density is required in order to sample from each input sequence X_i. As shown in the previous section, this density is defined:

$$p(Z_{i,j} = 1|X_i, \theta^{(t)}) = \exp\left[\sum_{m=1}^{W}\sum_{k\in\mathcal{L}}\left\{I(X_{i,j+m-1} = k)\ln\left(\frac{\theta_{m,k}^{(t)}}{\theta_{0,k}^{(t)}}\right)\right\}\right] \Big/ \Phi(i) \quad (5.13)$$

where $\Phi(i)$ is a normalising factor such that $\sum_{j=1}^{L_i-W+1} p(Z_{i,j} = 1|X_i, \theta^{(t)}) = 1$. To actually perform the sampling, the empirical cumulative density function $C(Z_{i,j} = 1|X_i, \theta^{(t)})$ is constructed. A random number $r \sim \text{Unif}[0,1]$ is drawn and a start site j_i chosen such that

$$C(Z_{i,(j_i-1)} = 1|X_i, \theta^{(t)}) < r \leq C(Z_{i,j_i} = 1|X_i, \theta^{(t)}). \quad (5.14)$$

Z_{i,j_i} becomes the sample for the sequence X_i. The procedure is repeated for all input sequences $i \in \{1,\ldots,N\}$.

U-step

As in the deterministic EM algorithm, the second step of the stochastic EM algorithm updates the parameters of the model. The U-step requires the construction of a proposed model θ' based on the samples from the S-step; in sEM, the parameters of the proposed model are the normalised ratio of nucleotide counts at each position in the motif. This is in contrast to the parameter updates in deterministic EM, which are the normalised ratio of *expected* nucleotide counts at each position in the motif. The parameters of the proposed model are therefore:

$$\theta'_{m,k} = \frac{\sum_{i=1}^{N} I(X_{i,j_i+m-1} = k) + \beta_k}{\sum_{i=1}^{N}\sum_{k\in\mathcal{L}} I(X_{i,j_i+m-1} = k) + \beta}, \quad (5.15)$$

for $m \in \{1,\ldots,W\}$ and $k \in \mathcal{L}$. Bi does not reestimate the background model, but this could be reestimated if desired. $\beta = \sum_{k\in\mathcal{L}} \beta_k$ is a vector of pseudocounts (also known as a *Laplace estimator*), equivalent to a Dirichlet prior distribution. The principle

reason for including pseudocounts in the parameter updates is to ensure that the value of each parameter never becomes zero. If the value of a parameter becomes zero at a given iteration, it will remain at zero for subsequent iterations. This becomes problematic when scoring subsequences using the motif model, as a probability of zero at one position in the motif would cancel the probabilities at other positions no matter how well the other positions matched the motif model. The use of pseudocounts avoids this situation. The choice of values for β is reasonably important, as overly large values would make the motif model more general by contributing more to the parameter values. Following Bailey and Elkan [10], the values of β_k are kept low ($\beta_k = 0.001$) in the algorithm developed in this thesis.

The Metropolis algorithm is used to decide whether θ' is kept or not. In SEAM, this is implemented by calculating the values of the entropy function $G(\cdot)$ of the current model ($\theta^{(t)}$) and the proposed model (θ'). The entropy function is described in Section 5.3.2. The change in $G(\cdot)$ is defined as:

$$\Delta G = G(\theta^{(t)}) - G(\theta') \tag{5.16}$$

and the Metropolis ratio is defined as:

$$\alpha_M(\theta', \theta^{(t)}) = \min\{1, \exp(-\Delta G)\}. \tag{5.17}$$

A random number $u \sim \text{Unif}[0,1]$ is drawn and the model updated to the proposed model only if u is less than or equal to the Metropolis ratio, that is:

$$\theta_{m,k}^{(t+1)} = \begin{cases} \theta'_{m,k}, & \text{if } u \leq \alpha_M(\theta', \theta^{(t)}), \\ \theta_{m,k}^{(t)}, & \text{otherwise} \end{cases} \tag{5.18}$$

for $m \in \{1, \ldots, W\}$ and $k \in \mathcal{L}$.

5.2.2 ZOOPS model stochastic EM expressions

This section presents a novel set of generalised expressions which implement the ZOOPS sequence model in the context of sEM. The method used to derive an expression for the sampling step is the same as in the OOPS model in Section 5.2.1; the relevant ZOOPS expressions as derived by Bailey and Elkan are used as required. Special consideration is given to the parameter update step, as the additional parameters in the ZOOPS model (namely the prior probability of a sequence containing a motif occurrence, γ) also require updating at each EM iteration.

Latent data ($Z_{i,j}^{(t)}$)

Bailey and Elkan's definition of the expectation of the latent data in the ZOOPS model is generalised here as:

$$Z_{i,j}^{(t)} = p(Z_{i,j} = 1|X_i, \theta^{(t)}) = \frac{f_j}{f_0 + \sum_{k=1}^{L_i-W+1} f_k}, \text{ where} \tag{5.19}$$

$$f_0 = p(X_i|Q_i = 0, \theta^{(t)})(1 - \gamma^{(t)}), \text{ and} \tag{5.20}$$

$$f_j = p(X_i|Z_{i,j} = 1, \theta^{(t)})(\frac{\gamma^{(t)}}{L_i - W + 1}), \quad 1 \leq j \leq L_i - W + 1. \tag{5.21}$$

Equation (5.21) makes use of the substitution in Equation (4.36), used in the generalisation of the ZOOPS model for deterministic EM. The definition of $Z_{i,j}^{(t)}$ in the ZOOPS model contains two terms: one representing sequences with a motif occurrence and the other representing sequences without a motif occurrence. Here, the variable Q_i is defined in the general form as: $Q_i = \sum_{j=1}^{L_i-W+1} Z_{i,j}$. That is, $Q_i = 1$ if X_i contains a motif occurrence and $Q_i = 0$ otherwise. $\gamma^{(t)}$ is the expected value of the prior probability of a sequence containing a motif occurrence at iteration t. In the U-step, this will be shown to be the empirical fraction of sequences containing a motif occurrence, based on the expected values of Q_i calculated in the S-step, as in the deterministic EM algorithm.

Conditional probability for a sequence

The conditional probability for a sequence containing a motif $p(X_i|Z_{i,j} = 1, \theta^{(t)})$ remains the same as in the OOPS model (Equation 5.3). The conditional sequence probability for a sequence without a motif occurrence is defined as:

$$p(X_i|Q_i = 0, \theta) = \prod_{l=1}^{L_i} \prod_{k \in \mathcal{L}} \theta_{0,k}^{I(X_{i,l}=k)}. \tag{5.22}$$

That is, every position in the sequence is regarded as being a background position.

S-step

As in the OOPS model, the S-step for the ZOOPS model samples from each input sequence X_i; substituting the conditional likelihoods for sequences with and without motif occurrences (Equations 5.3 and 5.22, respectively) into the expression for the

expectation of the latent data in the ZOOPS model (Equation 5.19) yields:

$$Z_{i,j}^{(t)} = \frac{\left(\prod_{l\in\Delta_{i,j}}\prod_{k\in\mathcal{L}}\theta_{0,k}^{I(X_{i,l}=k)}\ \prod_{m=1}^{W}\prod_{k\in\mathcal{L}}\theta_{m,k}^{I(X_{i,j+m-1}=k)}\right)\left(\frac{\gamma}{L_i-W+1}\right)}{\left(\begin{array}{c}\left(\prod_{l=1}^{L_i}\prod_{k\in\mathcal{L}}\theta_{0,k}^{I(X_{i,l}=k)}\right)(1-\gamma^{(t)}) \quad + \\ \sum_{j=1}^{L_i-W+1}\left\{\left(\prod_{l\in\Delta_{i,j}}\prod_{k\in\mathcal{L}}\theta_{0,k}^{I(X_{i,l}=k)}\ \prod_{m=1}^{W}\prod_{k\in\mathcal{L}}\theta_{m,k}^{I(X_{i,j+m-1}=k)}\right)\left(\frac{\gamma}{L_i-W+1}\right)\right\}\end{array}\right)} \tag{5.23}$$

Multiplying through by $(L_i - W + 1)$ yields:

$$Z_{i,j}^{(t)} = \frac{\left(\prod_{l\in\Delta_{i,j}}\prod_{k\in\mathcal{L}}\theta_{0,k}^{I(X_{i,l}=k)}\ \prod_{m=1}^{W}\prod_{k\in\mathcal{L}}\theta_{m,k}^{I(X_{i,j+m-1}=k)}\right)\gamma}{\left(\begin{array}{c}(L_i-W+1)\left(\prod_{l=1}^{L_i}\prod_{k\in\mathcal{L}}\theta_{0,k}^{I(X_{i,l}=k)}\right)(1-\gamma^{(t)}) \quad + \\ \sum_{j=1}^{L_i-W+1}\left\{\left(\prod_{l\in\Delta_{i,j}}\prod_{k\in\mathcal{L}}\theta_{0,k}^{I(X_{i,l}=k)}\ \prod_{m=1}^{W}\prod_{k\in\mathcal{L}}\theta_{m,k}^{I(X_{i,j+m-1}=k)}\right)\gamma\right\}\end{array}\right)} \tag{5.24}$$

and dividing through by $p(X_i|\text{background})$ as before yields:

$$Z_{i,j}^{(t)} = \frac{\prod_{m=1}^{W}\prod_{k\in\mathcal{L}}\left(\frac{\theta_{m,k}}{\theta_{0,k}}\right)^{I(X_{i,j+m-1}=k)}\gamma}{(L_i-W+1)(1-\gamma^{(t)})+\sum_{l=1}^{L_i-W+1}\left\{\prod_{m=1}^{W}\prod_{k\in\mathcal{L}}\left(\frac{\theta_{m,k}}{\theta_{0,k}}\right)^{I(X_{i,l+m-1}=k)}\gamma\right\}}. \tag{5.25}$$

This density was presented by Bi [23], but no derivation of the density was provided, nor were expressions given for the S- and U-steps of the sEM algorithm.

Note that the equivalent expression for the OOPS model (Equation 5.13) summed to 1 over each sequence X_i by definition: each sequence contained a motif occurrence. However, this is no longer the case; in the ZOOPS model, it is expected that $p(Z_{i,j} = 1|X_i, \theta^{(t)}))$ will tend to (near) 0 for all positions in sequences without a motif occurrence. To sample from each sequence, $Z_{i,j}^{(t)} \equiv p(Z_{i,j} = 1|X_i, \theta^{(t)}))$ is calculated for each position using Equation (5.25). These values are then normalised such that $\sum_{j=1}^{L_i-W+1} Z_{i,j}^{(t)} = 1$ (denote the normalised $Z_{i,j}^{(t)}$ values as $\bar{Z}_{i,j}^{(t)}$). For sequences without a motif occurrence (that is, with all $Z_{i,j}^{(t)}$ values near 0), this will effectively mean that the values of $\bar{Z}_{i,j}^{(t)}$ approximate the uniform prior distribution; although this means that a sample is chosen from that sequence with near-uniform probability, it will be shown in the U-step that this is of little consequence, as the samples from such sequences will have little weight in forming a proposed model. As with the OOPS model, sampling is performed by constructing the empirical cumulative distribution of the normalised values $C(\bar{Z}_{i,j}^{(t)})$, drawing $r \sim \text{Unif}[0,1]$ and choosing a start site j_i such that:

$$C(\bar{Z}_{i,(j_i-1)}^{(t)}) < r \leq C(\bar{Z}_{i,j_i}^{(t)}). \tag{5.26}$$

Again, $\bar{Z}_{i,j_i}$ becomes the sample for sequence X_i and the procedure is repeated for all input sequences $i \in \{1,\ldots,N\}$.

U-step

Again, the U-step requires the construction of a proposed model $\phi' = (\theta', \gamma')$ based on the samples from the S-step. However, new update expressions must be defined for the ZOOPS model. As not all sequences may contain a motif, it is reasonable to put less weighting on sequences that do not contain a motif when constructing the proposed model. The latent variables Q_i are useful here: recall that they take the value 1 if a sequence contains a motif occurrence and 0 otherwise. If the true values of Q_i were known, it would be possible to construct the proposed model from sequences where $Q_i = 1$, or perhaps only sample from these sequences. However, the value of Q_i is never calculated; only the expected value $Q_i^{(t)}$ is calculated. So there is no way of knowing for certain in advance which sequences contain a motif occurrence. However, the samples for each sequence drawn in the S-step can be weighted by the expectation that the sequence contains a motif occurrence. As the algorithm reaches convergence, it is expected that $Q_i \to 1$ for sequences containing a motif and $Q_i \to 0$ for sequences without a motif occurrence. By weighting the samples using the $Q_i^{(t)}$ values, samples from sequences without a motif occurrence will contribute less to the proposed model as the algorithm converges. The parameters for the proposed motif model are defined as:

$$\theta'_{m,k} = \frac{\sum_{i=1}^{N} I(X_{i,j_i+m-1} = k)\, Q_i^{(t)} + \beta_k}{\sum_{i=1}^{N} \sum_{k \in \mathcal{L}} I(X_{i,j_i+m-1} = k)\, Q_i^{(t)} + \beta}, \tag{5.27}$$

for $m \in \{1,\ldots,W\}$ and $k \in \mathcal{L}$. The parameter updates are similar to that for the OOPS model, but now take into account the fact that not all sequences may contain a motif occurrence. Again, the background model is not reestimated, but could be reestimated if desired. As before, $\beta = \sum_{k \in \mathcal{L}} \beta_k$ is a vector of pseudocounts (Laplace estimator), equivalent to a Dirichlet prior distribution.

A reestimate is also required for the other parameter in the model, γ. As in the deterministic EM derivations, the proposed value for the fraction of sequences containing a motif occurrence is just that, based on the values of $Q_i^{(t)}$ calculated in the S-step:

$$\gamma' = \frac{1}{N} \sum_{i=1}^{N} Q_i^{(t)}. \tag{5.28}$$

The update of γ from the expected values of the latent variables Q_i is therefore performed deterministically, unlike the update of θ, which is based on the sampled values

of the latent variables $Z_{i,j}$. A deterministic update is used here for a number of reasons. As the expected values of Q_i are calculated based on the current expected values of $Z_{i,j}$, the two sets of latent variables cannot be sampled independently; doing so would lead to an inconsistency between Z and Q. It may be possible to implement a more complex sampling scheme which accounts for this dependence, ensuring that the relationship between Z and Q (Equation 4.39) remains consistent. However, this is beyond the scope of this project. Additionally, at present, the Bayes-optimal classifier used in MITSU (defined in Equations 5.35-5.37) relies on the value of γ. The implications of updating γ based on a number of samples from Q on motif prediction are therefore also unclear.

As with the OOPS model, the Metropolis algorithm is used to decide whether to keep ϕ' or not. Again, this is implemented by calculating the values of the entropy function $G(\cdot)$ for the current and proposed models. The entropy function is described in Section 5.3.2. However, it is noted here that the entropy function for the ZOOPS model is dependent on both θ and γ; the model parameters are therefore collected as $\phi = (\theta, \gamma)$ and the value of the entropy function for the model ϕ is therefore denoted as $G(\phi)$. The change in $G(\cdot)$ is defined as:

$$\Delta G = G(\phi^{(t)}) - G(\phi') \tag{5.29}$$

and the Metropolis ratio is defined as:

$$\alpha_M(\phi', \phi^{(t)}) = \min\{1, \exp(-\Delta G)\}. \tag{5.30}$$

A random number $u \sim \text{Unif}[0,1]$ is drawn and the parameters updated to the proposed parameters only if u is less than or equal to the Metropolis ratio, that is:

$$\theta_{m,k}^{(t+1)} = \begin{cases} \theta'_{m,k}, & \text{if } u \leq \alpha_M(\phi', \phi^{(t)}), \\ \theta_{m,k}^{(t)}, & \text{otherwise,} \end{cases} \tag{5.31}$$

for $m \in \{1, \ldots, W\}$ and $k \in \mathcal{L}$ and

$$\gamma^{(t+1)} = \begin{cases} \gamma', & \text{if } u \leq \alpha_M(\phi', \phi^{(t)}), \\ \gamma^{(t)}, & \text{otherwise.} \end{cases} \tag{5.32}$$

5.2.3 A comparison of sequence sampling methods

As noted above, SEAM replaces the computation and maximisation of the expected complete-data log likelihood function (Q-function) with the much simpler estimation

of the posterior distribution $Z_{i,j}^{(t)} = p(Z_{i,j} = 1|X_i, \theta^{(t)})$ for each input sequence X_i, simulating a pseudo-sample from this distribution and updating the model θ based on the pseudo-complete samples [23]. The expression for the distribution has been noted previously (Equation 5.10). In a given EM iteration t, one sample is drawn from each density. This is done by evaluating $Z_{i,j}^{(t)}$ for all $j \in \{1, \ldots, L_i - W + 1\}$, summing to calculate the normalising factor $\Phi(i)$, then dividing each $Z_{i,j}^{(t)}$ by $\Phi(i)$. The values of $Z_{i,j}^{(t)}$ are sorted (largest first improves the efficiency of the sampling) and the empirical cumulative distribution function constructed. A uniform random number $u \sim \text{Unif}[0,1]$ is drawn and the empirical cumulative distribution function scanned until the value for a given position exceeds u. This position is chosen as the sample for sequence X_i. This method is equivalent to the weighted 'roulette wheel selection' (sometimes known as 'fitness proportionate selection') method in genetic algorithms. Having sampled each input sequence, a proposed model θ' is constructed from the samples and the current model $\theta^{(t)}$ updated to the proposed model if the Metropolis ratio is satisfied [23]. It is noted that, in using this method, the probability that a given position j in input sequence i is a motif occurrence ($Z_{i,j}^{(t)}$) must be evaluated for every position in i at every EM iteration in order to calculate the density. This may be inefficient, especially at later EM iterations, when the majority of $Z_{i,j}^{(t)}$ values are expected to be near zero. This motivates the consideration of alternative sampling strategies which could sample from an input sequence without having to evaluate $Z_{i,j}^{(t)}$ at each position.

One potential solution is to use a Markov Chain Monte Carlo (MCMC) strategy to sample from each input sequence. The simplest MCMC strategy (Metropolis algorithm with independence sampler) uses a uniform proposal distribution to sample from the target distribution $p(Z_{i,j} = 1|X_i, \theta^{(t)})$; use of the uniform proposal distribution greatly simplifies the calculation of the acceptance probability in the Metropolis algorithm. Analysis of the posterior distribution for a given input sequence $p(Z_{i,j} = 1|X_i, \theta^{(t)})$ shows that there is a large variation in the values taken by this distribution. Further, the value of $p(Z_{i,j} = 1|X_i, \theta^{(t)})$ at a given position j is not an indicator of the value at neighbouring positions $j-1$ or $j+1$. Despite these issues, it can be shown that the Metropolis algorithm with independence sampler does indeed converge to $p(Z_{i,j} = 1|X_i, \theta^{(t)})$ when taking large numbers of samples. However, this method is only an improvement on the roulette wheel selection method if the computational cost of drawing the required number of samples is substantially smaller than the cost of evaluating $Z_{i,j}^{(t)}$ at each position. As the number of samples is reduced, the lack of structure within the values of $p(Z_{i,j} = 1|X_i, \theta^{(t)})$ with regard to j means that the convergence of the

independence sampler towards $p(Z_{i,j} = 1|X_i, \theta^{(t)})$ will become considerably poorer.

Further, the lack of structure between values at adjacent positions means that it is impossible to choose an alternative standard proposal distribution from which to draw samples; unimodal distributions such as the Gaussian and Cauchy distributions which are regularly used in MCMC will not fit the target distribution well. As sEM converges, it is reasonable to expect that the motif model will become stronger and (at least in the OOPS sequence model) match one position in the input sequence much better than any other position, tending towards a single point of high probability. Although the distribution is expected to become unimodal as the algorithm becomes closer to convergence, it still cannot be modelled well by a standard proposal distribution as all probability mass will be at one position.

It is noted that more complex sampling strategies (for instance, with samples at a given EM iteration informed by the samples from previous iterations) may work well in this context. However, such strategies are beyond the scope of this project and, following the SEAM algorithm [23], a roulette wheel selection method is used to sample from the input sequences in the algorithm presented in the following section.

5.3 MITSU: a novel stochastic and entropy-based Expectation-Maximisation algorithm for motif discovery

This section builds on the results of the previous sections in order to set out the second major result of this chapter: the complete MITSU (**M**otif discovery by **IT**erative **S**ampling and **U**pdating)[2] algorithm for motif discovery. Section 5.3.1 begins by outlining the cut heuristic which is used in MITSU in order to remove the ZOOPS model constraint on motif distribution. This is followed by the definition of a new entropy function which is consistent with the improved sequence model used by MITSU. Classification in this sequence model is discussed (5.3.3), before Section 5.3.4 discusses the problem of determining convergence in sEM and describes the solution implemented by MITSU. Section 5.3.5 concludes by describing the MITSU algorithm in pseudocode.

[2]In Japanese, the word *mitsu* is the root word for *mitsudo*, meaning 'density'. Since stochastic EM samples from a probability density at each iteration, this name seems doubly appropriate.

5.3.1 Removing the ZOOPS constraint

Despite overcoming the limitations of the OOPS sequence model, the ZOOPS sequence model still enforces some constraints on the distribution of motif occurrences; it is assumed that each input sequence contains at most one occurrence of a motif. However, there are many biological examples of promoter regions which contain multiple copies of the same transcription factor binding site [21]. This is the primary motivation for the two-component mixture (TCM) model introduced by Bailey and Elkan, which allows an arbitrary number of non-overlapping motif occurrences in each input sequence [10]. Bembom, *et al.* have also noted that the OOPS and ZOOPS models perform significantly worse than TCM if their assumptions on the distribution of motif occurrences do not hold [21]. The difficulty in guaranteeing that these assumptions hold strengthens the motivation for implementing a sequence model which is unconstrained with regard to the distribution of motif occurrences.

The likelihood function for the TCM model is more computationally complex than those for the OOPS and ZOOPS models. As a result, exact methods based on the TCM model have been avoided in favour of more tractable approximations [21]. The TCM model presented by Bailey and Elkan uses a derived dataset consisting of all overlapping subsequences of width W from the original dataset [10]. Some proportion of these subsequences are motif occurrences; the remainder are background. While the subsequences in this derived dataset are necessarily overlapping, the likelihood function is based on a sample of independent sequences [21]. Bembom, *et al.* note that the effect of this violation of the independence assumption is unclear. An additional smoothing step is required in order to counter the faulty independence assumption and reduce the degree to which two overlapping subsequences can both be assigned to the motif component of the model.

Keles, *et al.* suggest an alternative cutting heuristic which involves deriving a different dataset from the original, then applying motif discovery to the derived dataset, using the ZOOPS model [90]. The main advantages of this method are that no additional steps are required to deal with the assumption of independence and the approximation to the likelihood function is improved. This method is improved by Bembom, *et al.* [21] and is implemented in MITSU. Briefly, Bembom, *et al.*'s method cuts the original dataset into a small number of overlapping subsequences of a given length U, such that each subsequence contains the first $(W-1)$ positions of the next subsequence. The ZOOPS model is then applied to this derived dataset. The previous

studies implementing this heuristic have shown that the method is fairly robust with respect to the choice of cut length U but have suggested that this parameter may be optimised using cross-validation [90, 21]. The cut heuristic is implemented here as an inner loop within the motif discovery algorithm (Section 5.3.5). The ZOOPS model is applied to derived datasets with varying values of U and the parameter settings that yield the highest value of $G(\cdot)$ are returned as the best motif model. It will be shown in Section 5.4.3 that the cut heuristic in combination with the ZOOPS model successfully allows discovery of multiple copies of the same motif within a single input sequence, in the context of motif discovery using sEM.

5.3.2 Defining an entropy function

Motivation

As noted above, the entropy function $G(\theta)$ is used in SEAM to monitor convergence and the strength of a particular motif model θ. Although introduced by Bi as an 'energy' function [23], $G(\theta)$ is actually an entropy-based function, proportional to the sum of entropies for the motif and background models. $G(\theta)$ is defined:

$$G(\theta) = N\left(\sum_{k\in\mathcal{L}} \theta_{0,k} \ln \theta_{0,k} + \sum_{m=1}^{W}\sum_{k\in\mathcal{L}} \theta_{m,k} \ln \theta_{m,k}\right). \tag{5.33}$$

Bi notes that $G(\theta)$ is related to the relative entropy discussed by Stormo, which has been shown to be proportional to the DNA-protein binding strength [160].

One consequence of implementing and optimising the entropy function $G(\theta)$ rather than a likelihood-based function as in MEME (the Q function for the OOPS model is given in Equation 4.8, page 84) is the difference in distributions which optimise these functions. Algorithms which optimise a likelihood-based function (for example, MEME) tend to discover motif models with positions (that is, PWM columns) which have a significantly different distribution from the background model. In contrast, the entropy-based function implemented in SEAM is maximised in the case where the distribution of the probability mass at a given position is significantly more concentrated than the background; that is, positions where the probabilities are more diffuse will be penalised by the entropy function.

Analysis of the *E. coli* dataset described in Section 3.2 illustrates why maximising an entropy-based function may be useful for motif discovery. Figure 5.1 plots the highest probability at each position for each motif in the *E. coli* dataset against the infor-

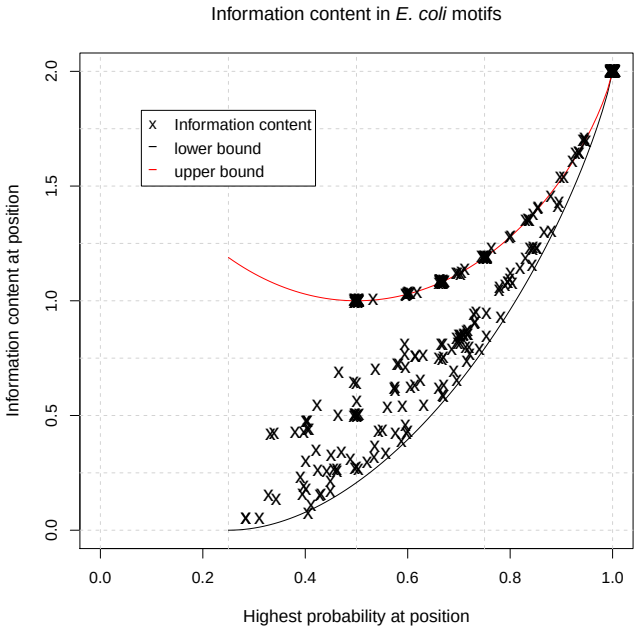

Figure 5.1: Information content of *E. coli* motif positions, plotted against the probability of the dominant nucleotide at that position. Upper and lower bounds of information content are plotted for comparison. Maximising entropy by optimising $G(\theta)$ may improve motif discovery by tending towards models with a more concentrated probability mass at each position.

mation content (relative entropy) of that position. Upper and lower bounds for the information content are plotted for comparison; given a highest probability of p at a particular position, these bounds are reached with motif columns of $[p,(1-p),0,0]^T$ and $\left[p,\frac{(1-p)}{3},\frac{(1-p)}{3},\frac{(1-p)}{3}\right]^T$, respectively. While some positions are close to the bounds, there are a large number of positions with information content between these bounds; this gives scope for a trade-off between increasing the highest probability at a position and increasing the information content. The definitions of information content and relative entropy (page xvii) make clear that these quantities rely on all nucleotides at a given position. Therefore, it is possible to increase the entropy at a given position while the highest probability at that position remains constant, by altering the probabilities of the three remaining nucleotides. It follows that maximising $G(\theta)$ may result in stronger motif models (that is, with increased information content) as a consequence of having a more concentrated probability distribution at each position. Optimising the entropy-based function rather than a likelihood-based function therefore allows a different search, which may be useful for motif discovery.

The entropy function $G(\theta)$ as defined by Bi is used in the context of the OOPS sequence model in SEAM. However, the properties of the OOPS entropy function mean that the function becomes problematic when used with the ZOOPS sequence model and cut heuristic, which are used in combination in order to implement discovery of multiple motifs within a single input sequence in MITSU. The main problem stems from the fact that $G(\theta)$ is scaled by the number of input sequences N (Equation 5.33). N is assumed to be constant in the SEAM algorithm; however, it follows that the values of $G(\theta)$ cannot be fairly compared between datasets with differing values of N. Employing the cutting heuristic means that the value of N may double, or triple, depending on the cut length (U). A method of fairly comparing values of $G(\theta)$ is required. Of further interest are the properties of the entropy function, particularly how it varies with changing motif conservation and varying values of γ. Recall that γ is the proportion of input sequences containing a motif occurrence; it is therefore related to N and the number of motif occurrences within the dataset.

Figure 5.2 shows how $G(\theta)$ varies with motif conservation. Here, the lower bound of $G(\theta)$ is plotted; that is, motif conservation is defined as the probability of the dominant nucleotide at each motif position, with the probability of the remaining nucleotides being split equally. For example, a motif conservation of c is equivalent to a motif model with columns $[c, \frac{(1-c)}{3}, \frac{(1-c)}{3}, \frac{(1-c)}{3}]^T$. $G(\theta)$ has a negative value, increasing exponentially with increasing motif conservation[3]. $G(\theta)$ was designed for use with the OOPS sequence model and therefore includes no factor to account for the fact that only some proportion of the input sequences may contain a motif occurrence (this is assumed to be all input sequences in the OOPS model); the values of $G(\theta)$ plotted in Figure 5.2 would not vary as the proportion of sequences containing a motif occurrence varied. Using the OOPS entropy function $G(\theta)$ in situations where the OOPS assumptions do not hold (for example, in the ZOOPS setting) can become problematic. Example 5.1 shows how this may be a problem.

In the case of MITSU, the ZOOPS model is used in combination with a cutting heuristic, in order to discover multiple motif occurrences within a single input sequence. However, as noted above, $G(\theta)$ is independent of γ, the proportion of sequences containing a motif occurrence. This becomes even more problematic when the dataset is cut in order to discover multiple motifs in an input sequence, as illustrated in Example 5.2.

[3]It is noted that Bi's entropy function drops the negative sign from the conventional definition of entropy; maximising $G(\theta)$ therefore corresponds to minimising a conventional entropy. Following Bi, the ZOOPS entropy function (to be discussed shortly) is defined similarly.

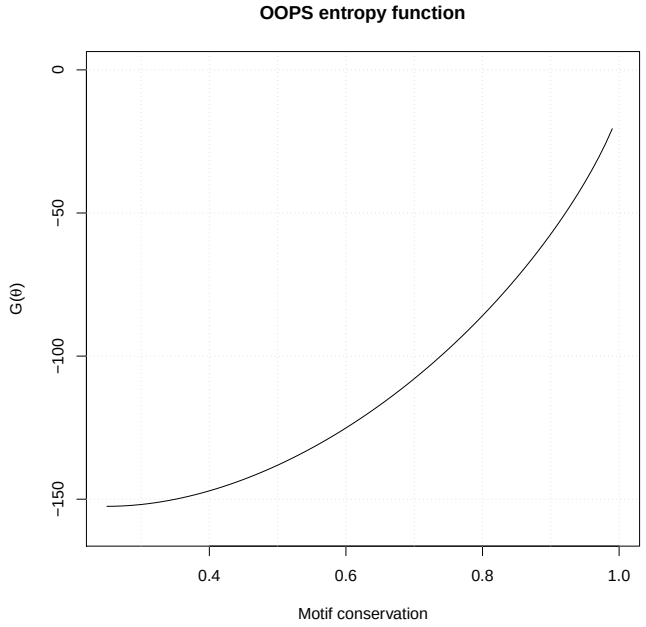

Figure 5.2: A plot of the lower bound of the original (OOPS) motif entropy function (Equation 5.33), assuming $N = 10$ and $w = 10$. While $G(\theta)$ increases with increasing motif conservation, it is independent of the proportion of input sequences containing a motif occurrence (γ).

Entropy function definition

Together, the issues discussed above motivate the definition of a new entropy function which can be used in sEM-based motif discovery with the ZOOPS sequence model. This new function should remain consistent when used in conjunction with the cutting heuristic implemented in MITSU. Here, a modification to the original entropy function is presented, such that:

$$G(\phi) = \frac{1}{\gamma N}\left(\sum_{k\in\mathcal{L}} \theta_{0,k} \ln \theta_{0,k} + \sum_{m=1}^{W}\sum_{k\in\mathcal{L}} \theta_{m,k} \ln \theta_{m,k}\right), \tag{5.34}$$

where there is now a direct reliance on the proportion of sequences containing a motif occurrence (γ) and, as before, the model parameters are now collected and denoted as $\phi = (\theta, \gamma)$. It can be shown that the new (ZOOPS) entropy function $G(\phi)$ satisfies the requirements; the properties of the $G(\phi)$ are discussed in the following section.

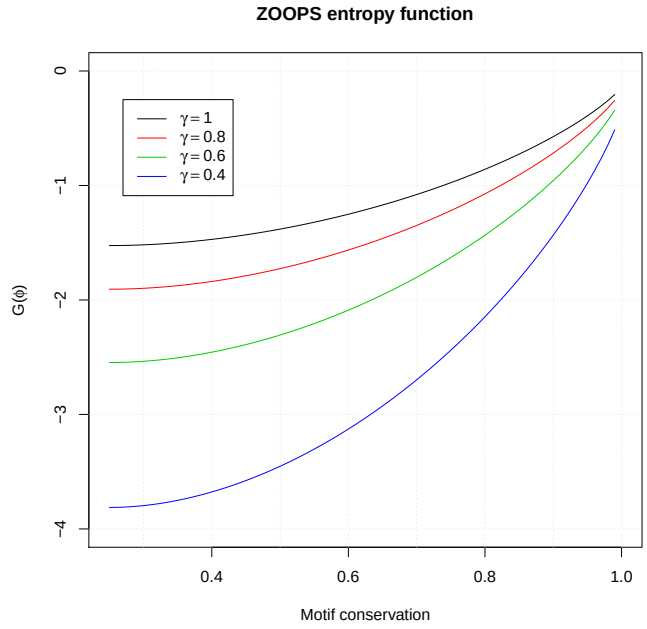

Figure 5.3: A plot of the lower bound of the ZOOPS motif entropy function (Equation 5.34), for varying proportions of input sequences containing a motif occurrence (γ). As in Figure 5.2, it is assumed that $N = 10$ and $w = 10$. It is observed that $G(\phi)$ is dependent on both motif conservation and γ.

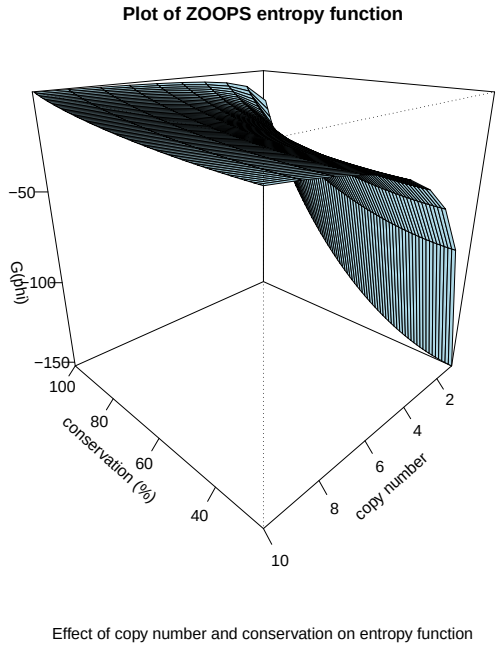

Figure 5.4: 3-dimensional surface plot of the lower bound of the ZOOPS motif entropy function (Equation 5.34).

A dataset of size N contains two motifs of equal width W and equal conservation C in differing proportions: motif A occurs in all sequences, while motif B occurs in half the sequences. It would be preferable to discover motif A first due to its occurrence in all sequences; however, this cannot be guaranteed when using the original entropy function (Equation 5.33), as both motifs have equal values of $G(\theta)$.

Example 5.1: The OOPS entropy function can become problematic in the ZOOPS setting.

A dataset X of size $N = 10$ contains a single, perfectly conserved motif. The motif occurs twice in the first sequence and once in the remaining sequences. The sEM algorithm is run on X and discovers one occurrence of the motif in each input sequence (that is, the second occurrence in the first sequence is missed). X is then 'cut' to give a derived dataset X' with $2N = 20$ sequences, each half the length (ignoring the cut overlap for now). In X', 11 sequences contain one motif occurrence. The sEM algorithm is run again and discovers all occurrences. The constructed model in each case has the same value of $G(\theta)$; however, the second case is preferable to the first, as all occurrences of the motif are discovered.

Example 5.2: The OOPS entropy function cannot handle cases where the cutting heuristic is used to discover multiple motif occurrences in a sequence.

Properties of $G(\phi)$

The dependence of the ZOOPS entropy function $G(\phi)$ on motif conservation and the proportion of sequences containing a motif occurrence (γ) is shown in Figure 5.3. Again, the lower bound is plotted, with motif conservation defined as above. $G(\phi)$ is shown to increase with increasing motif conservation; however, for any given motif conservation level, $G(\phi)$ now decreases as the proportion of sequence containing a motif occurrence decreases. Figure 5.4 shows the 3-dimensional surface plot for $G(\phi)$, plotting the value of $G(\phi)$ against both motif conservation and copy number (equal to γN). The 3-dimensional plot confirms that the $G(\phi)$ is maximised with a perfectly conserved motif occurring in every input sequence. The inclusion of the γN factor in the definition of $G(\phi)$ means that the function is unaffected in cases where datasets are derived by the cut heuristic (γ and N cancel each other in such cases). It can be shown that the following three useful properties hold:

If two motifs are equally conserved, the motif with the higher number of occurrences will have a higher value of $G(\phi)$. This follows as a direct result of the γN factor in the definition of $G(\phi)$. Consider a dataset consisting of 10 sequences, each containing a perfectly conserved motif A and 5 of the sequences also containing a secondary motif B. The proportion of sequences containing these motifs are clearly $\gamma_A = 1.0$ and $\gamma_B = 0.5$. This difference in the number of motif occurrences reduces the theoretical maximum value of $G(\phi)$ for motif B; it follows that the higher value of $G(\phi)$ for motif A means that it is more likely that the motif discovery algorithm will converge to a model representing the motif with the higher number of occurrences. It should be noted that this property holds (on a much smaller scale) even without the γN factor in the entropy function. The reason for this is due to how the sEM expressions for the ZOOPS model update the motif model at each iteration. The expected probability that a sequence contains a motif occurrence $Q_i^{(t)}$ is used to weight the samples from each input sequence. In sequences which do not contain a motif occurrence, $Q_i^{(t)}$ tends towards 0, but in practice remains very slightly above 0. This has the effect of adding some noise to the motif model, hence producing a slightly lower value of $G(\phi)$.

All else being equal, a higher proportion of sequences containing a motif occurrence will yield a higher value of $G(\phi)$. Again, this follows directly from the definition of $G(\phi)$ (Equation 5.34). If the number of input sequences N and the motif conservation (through θ) are held constant, increasing γ also increases $G(\phi)$.

Given two motifs of equal prevalence and unequal motif conservation, the motif discovery algorithm will tend to discover the motif with the higher value of $G(\phi)$ (equivalently, the higher motif conservation). To confirm that the sEM algorithm used in MITSU tends to discover better conserved motifs, a test dataset was constructed. Ten input sequences were created, each containing a perfectly conserved motif A. A secondary motif B was added to each sequence, but with mutations such that 1 letter in each occurrence was changed *and* 1 letter in each position was changed (a conservation of 90%). Calculating the expected values of $G(\phi)$ for motifs A and B using Equation 5.34 confirms that the value of $G(\phi)$ for motif B is lower than that for motif A. Running MITSU with 1,000 random seeds returned motif A perfectly in around 8% of runs. Motif B was returned perfectly in around 1% of runs (a similar percentage of runs converge to slightly weaker versions of motif C). This result should perhaps not be surprising, as the algorithm is designed to avoid becoming trapped in

local optima.

The ZOOPS entropy function $G(\phi)$ and the cutting heuristic

Example 5.3 demonstrates how the cutting heuristic affects $G(\phi)$. Where the ZOOPS assumption holds (that is, there is at most one motif occurrence per input sequence), the γN factor in definition of $G(\phi)$ ensures that all cuts of the dataset should yield the same value of $G(\phi)$. Where the ZOOPS assumption does not hold (that is, there is at least one sequence with more than one motif occurrence), the γN factor ensures that the cut with the highest expected copy number is returned as the best result.

Conclusion

The ZOOPS entropy function defined in Equation 5.34 is adopted in MITSU. However, it is noted that other alternative entropy functions may be possible; since the sEM accept/reject mechanism is based on a difference of entropies, substituting other functions based on the model entropy should have little effect on this mechanism.

In Figure 5.5, dataset A consists of 10 input sequences, each containing a motif occurrence. Each sequence is cut in half to construct dataset B; half of these sequences now contain motif occurrences, while the rest do not. Therefore, $N_A = 10, N_B = 20, \gamma_A = 1.0, \gamma_B = 0.5$. Assume that in each case the motif is discovered perfectly. The γN factor in the definition of $G(\phi)$ ensures that, in theory, the motif model has the same value of $G(\phi)$ in both datasets.

Dataset C contains a perfectly conserved motif similar to A, but contains an additional motif occurrence in the first input sequence. This additional occurrence is not discovered by the algorithm. Again, each sequence is cut in half to construct dataset D. This time, the algorithm discovers every occurrence; therefore, $N_C = 10, N_D = 20, \gamma_C = 1.0, \gamma_D = 0.55$. The γN factor in the definition of $G(\phi)$ ensures that a higher expected copy number gives a higher value of $G(\phi)$; therefore, dataset D is returned as the best result.

Example 5.3: Demonstrating properties of the ZOOPS entropy function on cut datasets.

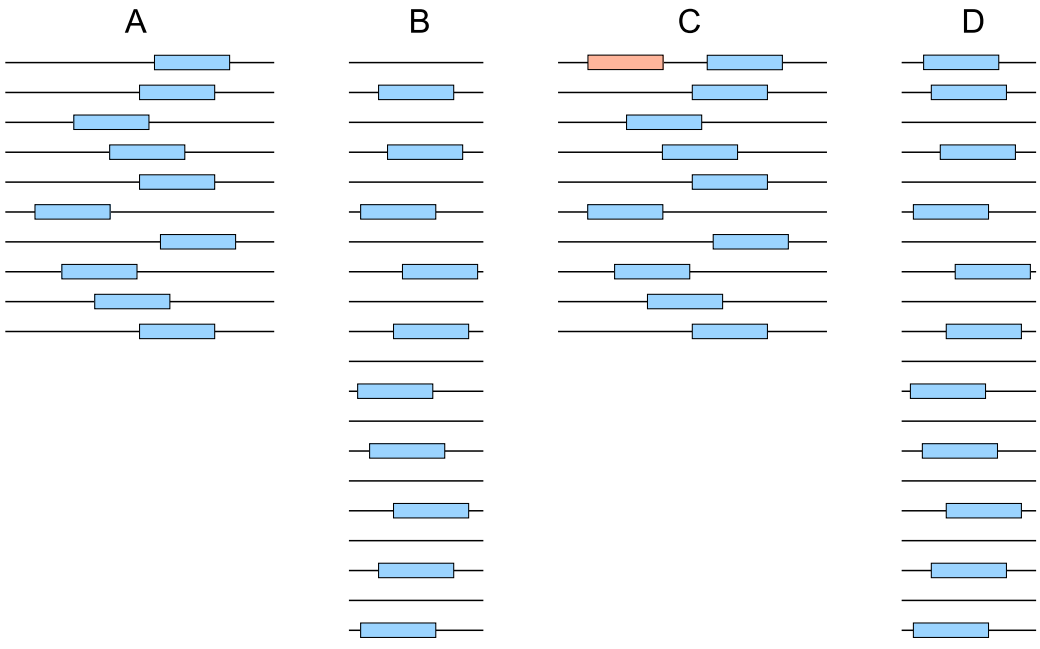

Figure 5.5: Cut datasets demonstrating properties of the ZOOPS entropy function (refer to Example 5.3).

5.3.3 Classification in the stochastic EM ZOOPS model

MITSU's use of the ZOOPS sequence model in the context of sEM raises a further issue, that of predicting the positions of motif occurrences given the output motif model. When using the OOPS sequence model, the positions in each sequence corresponding to the samples drawn in the last iteration of the algorithm are deemed to be the occurrences of the motif. This is the method of prediction used in SEAM. However, since the ZOOPS model used by MITSU allows sequences to contain no motif occurrences, some further work is required in order to determine the position of predicted motif occurrences. Bailey and Elkan state that a Bayes-optimal classifier may be formed, classifying a subsequence $X_{i,j}$ as a motif occurrence if:

$$\frac{p(X_{i,j}|\text{motif model})}{p(X_{i,j}|\text{background model})} > \frac{p(\text{background})}{p(\text{motif})}, \tag{5.35}$$

where the terms on the right hand side are prior probabilities. In their original definitions, these prior terms are defined in terms of λ; however, following the substitution made in generalising the EM expressions (Equation 4.36, page 91), these prior terms are defined here as:

$$p(\text{background}) = 1 - \frac{\gamma}{L_i - W + 1} \tag{5.36}$$

and

$$p(\text{motif}) = \frac{\gamma}{L_i - W + 1}, \tag{5.37}$$

for a given sequence X_i. Motif classification is implemented in MITSU as follows. For each sequence in the derived dataset, each position is scored using the log-odds matrix formed from the motif and background models. The subsequence with the largest log-odds score is classified as an occurrence of the motif, with the additional constraint that Equation 5.35 must also hold. This results in at most one occurrence per sequence within the derived dataset, with zero occurrences, where the constraint does not hold.

This method improves on the method used in SEAM in that by discovering a motif model first and then performing classification of positions, poor final sample positions are allowed to be discarded in favour of positions which better fit the discovered model. That is, the sEM sampling procedure may select a non-matching position in a given sequence by chance (recall that at each iteration of the algorithm, there is a non-zero probability of accepting a sample which reduces the likelihood); if the algorithm were stopped at this point and the final samples regarded as motif occurrences, an incorrect occurrence would be predicted, despite having predicted a good motif model. By predicting motif occurrences using the predicted motif model rather than using the

positions sampled in the final iteration of the algorithm, this scenario is avoided. Of course, it may be expected that the majority of the time, the final samples will indeed be the motif occurrences; however, this may depend on how strong the motif model is.

Predicting more than one occurrence per sequence

MITSU uses the Bayes-optimal classifier to predict motif occurrences in the derived dataset, using the ZOOPS sequence model. The cutting heuristic outlined above means that, although each derived sequence may contain at most one motif occurrence, the sequences in the original dataset may contain more than one predicted motif occurrence. Once the predictions of motif occurrences in the derived sequences have been made, the final step is to map these predictions back to their positions in the original dataset. This is relatively trivial and involves reversing the cutting procedure which was used to create the derived dataset, ensuring that the overlaps between cuts are dealt with appropriately.

5.3.4 Stochastic EM convergence and stopping rules

In this section, it is shown that sEM is known to converge in general. Convergence of sEM in the context of MITSU is shown through example in Section 5.4. The difficulty of defining a suitable stopping rule for sEM is also discussed, as well as how stopping is implemented in MITSU.

Stochastic EM convergence

A Markov chain requires two properties to ensure that the chain converges to the desired distribution: the desired distribution must be an *invariant distribution* of the chain. The chain must also be *ergodic* [115].[4]

A distribution is invariant (or stationary) with respect to a Markov chain if every step in the chain leaves that distribution unchanged. A given Markov chain may have more than one invariant distribution (for example, if the transition probabilities are given by the identity transformation, then any distribution will be invariant). MacKay also notes that an invariant distribution is an eigenvector of the transition probability matrix with eigenvalue 1 [115]. Diebolt and Ip [49] state that for sEM, alternately imputing pseudo-complete data (S-step) and performing a subsequent maximisation (U-step) generates a Markov chain that converges to an invariant distribution under the

[4]Many chains also satisfy *detailed balance*, but this is not a requirement.

mild condition that all parameters are positive. The use of pseudocounts when updating the motif model parameters ensures that this is the case; indeed, this is always the case when using Dirichlet prior distributions [85].

An ergodic Markov chain is one that converges to a required invariant distribution regardless of the choice of initial distribution. If a Markov chain is ergodic, there will be only one invariant distribution. This distribution is then known as the *equilibrium distribution*. Celeux and Diebolt [38] prove the ergodicity of the Markov chain generated by sEM in a mixture context (this is the context in which sEM is used here for motif discovery). Briefly, this reduces to showing that the sequence of samples is a finite-state homogeneous irreducible and aperiodic Markov chain. This guarantees the (weak) convergence to the unique stationary distribution of the ergodic Markov chain generated by sEM.[5]

Although (simple) lower bounds on the time required for convergence of MCMC methods can be calculated, putting an exact figure on this is a difficult problem in general, and most theoretical results for upper bounds are of little practical use [115]. However, in practice, MITSU was found to take approximately 5 times longer to converge than deterministic EM, based on tests with the CRP dataset described in Section 3.6.1.

Stopping stochastic EM

Diebolt and Robert [51] note that, although satisfactory convergence results for sEM have been published [38, 49], designing a stopping rule for sEM is challenging: a simple deterministic stopping rule as implemented in the EM algorithm may be triggered by what is a chance fluctuation stemming from the S-step of the algorithm. A number of different approaches for stopping sEM have been suggested. Early approaches simply ran sEM for a large number of iterations in place of a stopping rule, which can be inefficient, especially if sEM converges in a relatively small number of iterations [38]. Recent stopping rules suggest monitoring the gradient of the likelihood function [74] or differences in the Q function [33]; however, the method used most often is the implementation of a deterministic stopping rule for a number of successive iterations to reduce the chances of a premature stop [28, 29]. This method has been criticised for placing too much emphasis on the parameter estimates with little regard

[5] In addition to these results, Diebolt and Robert prove convergence of the Data Augmentation algorithm [50, 51], which can be viewed as a Bayesian version of sEM (Celeux *et al.* note that sEM is a special case of the Data Augmentation algorithm, where some suitable non-informative prior is used [37]).

to the estimation of information [33]. However, in the context of motif discovery, it is the motif model parameter estimates which are of interest therefore this is the method implemented in MITSU. After each sEM iteration, the Euclidean distance between the current and previous motif models is calculated. If this distance is below a given threshold for three successive iterations, the algorithm is deemed to have converged. Booth and Hobert suggest reducing the stringency of the threshold for stochastic variants of EM in comparison with deterministic EM as a result of the added Monte Carlo error [28]. Following this suggestion, the threshold for MITSU is chosen to be 10^{-3} (the threshold used in MEME is 10^{-6}). It is noted that a threshold of 10^{-3} corresponds to an average change of 0.00014 in each motif model parameter when $W = 12$; this change is deemed to be sufficiently small to diagnose convergence.

5.3.5 MITSU pseudocode

Pseudocode describing MITSU is presented in Algorithm 5.1. The sEM algorithm at the heart of MITSU is run for several initial values of γ for each of n random seeds. If $W_{min} \neq W_{max}$, the algorithm is run for each possible motif width and the most likely width estimated using the MCOIN heuristic. The complete algorithm may be run multiple times to discover multiple different motifs (which may be of different width and have a different distribution of motif occurrences within the dataset); the method for discovering multiple different motifs will be discussed in Section 6.2.

```
procedure MITSU algorithm
 create Markov background model
 for M motifs do
  for W = W_min to W_max do
   for cut length in {set of cut lengths} do
    for n random seeds do
     for γ = 1/√N to 1 by ×2 do
      run sEM on cut dataset using ZOOPS model at width W:
      until convergence do
       S-step (Equation 5.25)
       U-step (Equations 5.31-5.32)
      end
     end
    end
    return the best motif model over n random seeds & varying γ
   end
   return the best motif model over all cut lengths
  end
  estimate most likely width Ŵ using MCOIN (Algorithm 4.1)
  return motif model and list of predicted sites for Ŵ
  probabilistically erase motif occurrences from dataset (Equations 6.1-6.3)
 end
end MITSU algorithm
```

Algorithm 5.1: MITSU pseudocode

5.4 Validating MITSU

This section details the experimental validation of the algorithm developed in this thesis. The results of this validation show that MITSU can discover unknown motifs in a dataset consisting of the upstream sequences of coregulated genes. The results also confirm the hypothesis in that transcription factor binding site discovery using stochastic EM is shown to improve on deterministic EM in terms of previously established metrics. Further results demonstrate that the stochastic EM algorithm allows MITSU to escape insignificant local maxima of the likelihood function which can trap deterministic algorithms and that MITSU can successfully discover multiple copies of

a motif within a single input sequence.

5.4.1 Stochastic EM outperforms deterministic EM

MITSU was evaluated quantitatively using a mixture of realistic synthetic and previously characterised real data. Datasets were constructed as described in Chapter 3. Briefly, five large data collections each consisting of 1,000 datasets were constructed using synthetic motifs of varying conservation and realistic *E. coli* background sequence extracted from the EcoGene database [144]. A sixth data collection consisting of 20 datasets was constructed using known *E. coli* TFBS sequences extracted from RegulonDB [65]. Finally, a data collection consisting of nine datasets was constructed using known TFBS motif sequences from diverse prokaryotic species. These motif sequences were discovered by ChIP methods. Background sequences for these datasets were constructed using synthetic data, altering the probability of choosing each nucleotide to reflect the species GC-content as required. Tables 5.1, 5.2 and 5.3 summarise the results of the tests on these data collections. MITSU is compared here against the results of the SEAM algorithm [23] and a deterministic EM-based motif discovery algorithm. This deterministic EM-based algorithm is a reimplementation of the original MEME algorithm [10], used here in order to compare deterministic EM against stochastic EM more fairly, without the improvements gained through the use of additional heuristics. The deterministic EM results were reported in Section 4.4.3, Tables 4.3 (page 108), 4.5 (page 110) and 4.7 (page 110) and are repeated here for convenience. Detailed results for Tables 5.1, 5.2 and 5.3 are provided in Tables B.4, B.5 and B.6 respectively in Appendix B. AUC results are not available for SEAM due to the method of prediction used (the final sample in each input sequence is regarded to be the predicted position for that sequence, rather than scoring each position using a log-odds matrix constructed from the motif model, as in MEME and MITSU).

Realistic synthetic data

Tests on realistic synthetic data (Table 5.1) show that mean site-level sensitivity (*sSn*) and positive predictive value (*sPPV*) decrease with decreasing motif conservation for all three tested algorithms. This behaviour was noted in deterministic EM in Section 4.4.3; the decrease in *sSn* is due to fewer sites being predicted overall. The decrease in *sPPV* is due to the background sites better matching the motif sites as conservation decreases, leading to an increase in the number of false positive results.

Conservation	Deterministic EM			SEAM			MITSU		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
2.00	0.84‡	0.25	**0.99**‡	**1.00**†‡	**1.00**†‡	-	0.70	0.74†	0.97
1.49	0.26	0.07	0.98	**0.93**†‡	0.93†	-	0.90†	**0.97**$^{\dagger *}$	**1.00**†
1.08	0.02	0.01	0.96	0.49†	0.49†	-	**0.68**$^{\dagger *}$	**0.78**$^{\dagger *}$	**0.99**†
0.76	0.00	0.00	**0.94**	0.09†	0.09†	-	**0.17**$^{\dagger *}$	**0.20**$^{\dagger *}$	**0.94**†
0.51	0.00	0.00	**0.93**‡	0.06†	0.06†	-	**0.07**$^{\dagger *}$	**0.08**$^{\dagger *}$	**0.93**

Table 5.1: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for five collections of realistic synthetic data with varying levels of motif conservation. Best mean results are printed in bold. In these tests, motif discovery was carried out only at the known motif width. Results marked † are statistically significant with regard to deterministic EM, results marked * are statistically significant with regard to SEAM and results marked ‡ are statistically significant with regard to MITSU, all at $p \leq 0.05$ (see main text).

In the majority of tests, MITSU outperforms both the deterministic EM algorithm and SEAM, particularly with regard to *sSn* and *sPPV*. The increased performance at lower levels of motif conservation is particularly notable. The success of MITSU is due to making fewer, but more accurate, predictions. The predictions made are generally more cautious; positions which might previously have been false positive predictions are now more likely to be classified as true negative predictions. This significant reduction in the number of false positive predictions explains the large increase in the *sPPV* values.

It is also notable that the *sSn* and *sPPV* results for the sEM-based algorithms are better balanced, in that the *sSn* and *sPPV* results are more closely matched. The results for the deterministic EM algorithm, particularly at high levels of motif conservation, tend towards increasing *sSn* at the expense of *sPPV*; that is, fewer false negative predictions were made at the expense of having a greater number of false positive predictions. While in the case of SEAM the improved balance between *sSn* and *sPPV* is a result of using the OOPS sequence model (*sSn* and *sPPV* results will always be equal using this model), MITSU is genuinely less biased towards *sSn*, producing fewer false predictions in general.

As in Section 4.4, statistical significance tests for the results in this section were

Conservation	Deterministic EM			SEAM			MITSU		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
'high' (1.36)	**0.81**‡	0.22	0.96	0.67‡	0.67†	-	0.54	**0.75**†	**0.98**
'low' (0.78)	0.63	0.41	0.96	**0.65**	0.65	-	0.57	**0.71**†	**0.97**
overall (1.13)	**0.74**‡	0.30	0.96	0.66‡	0.66†	-	0.55	**0.73**†	**0.98**†

Table 5.2: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for 20 datasets created using previously characterised *E. coli* TFBS sequences. Best mean results are printed in bold. In these tests, motif discovery was carried out only at the experimentally determined motif width. Results marked † are statistically significant with regard to deterministic EM and results marked ‡ are statistically significant with regard to MITSU, both at $p \leq 0.05$ (see main text).

Conservation	Deterministic EM			SEAM			MITSU		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
0.99	0.75	0.67	0.99	0.86	0.86†	-	**0.88**†	**0.92**$^{\dagger *}$	**1.00**

Table 5.3: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for 9 datasets created using real prokaryotic data determined through ChIP experiments. Best mean results are printed in bold. In these tests, motif discovery was carried out only at the experimentally determined motif width. Results marked † are statistically significant with regard to deterministic EM and results marked * are statistically significant with regard to SEAM, both at $p \leq 0.05$. (see main text).

performed using a paired one-sided Wilcoxon signed rank test[6]. In general, the sEM-based algorithms were shown to give significantly better results when compared to deterministic EM (Table 5.1). All *sSn* and *sPPV* results for SEAM were significantly greater than those for deterministic EM ($p < 2.20 \times 10^{-16}$ in all cases). *sSn* and *sPPV* results for MITSU were also significantly greater than those for deterministic EM in all cases ($p < 2.20 \times 10^{-16}$) except for *sSn* in the group with mean conservation 2.00 bits/col. MITSU was also significantly better than deterministic EM in terms of AUC at intermediate levels of conservation ($p < 2.20 \times 10^{-16}$); deterministic EM was shown to be significantly better than MITSU for the groups with mean conservations of 2.00 bits/col and 0.51 bits/col ($p = 1.67 \times 10^{-10}$ and $p = 1.47 \times 10^{-3}$, respectively). For

[6]Again, *p*-values smaller than the machine epsilon in R (2.20×10^{-16}) are presented as '$< 2.20 \times 10^{-16}$'.

the results where MITSU outperformed SEAM in terms of mean values, these results were also significantly better ($p < 2.20 \times 10^{-16}$, except *sSn* for the group with mean conservation 0.51 bits/col: $p = 4.25 \times 10^{-6}$).

E. coli and prokaryotic ChIP data

Tables 5.2 and 5.3 present the results of tests on previously characterised *E. coli* TFBS sequences and TFBS sequences from diverse prokaryotes determined by ChIP experiments, respectively. The general trend remains the same: both *sSn* and *sPPV* decrease with decreasing motif conservation. As noted in Section 4.4.3, deterministic EM-based motif discovery is shown to achieve better classification results on previously characterised *E. coli* data than could be expected given realistic synthetic data of a similar conservation. Again, this improvement in performance is due to the differences in motif structure. Whereas the conservation of the synthetic motifs used here is independent of position, real TFBS motifs with low mean conservation often have clusters of well-conserved positions (this phenomenon was shown in *E. coli* motifs in Section 3.2); based on this observation, it is likely that differences in the distribution of high and low conservation across true motifs in comparison with synthetic motifs explain the improvement in performance on real data. A similar trend is seen in the results of SEAM and MITSU, particularly at lower levels of motif conservation.

As with the realistic synthetic data, MITSU is shown to increase *sPPV* by making fewer, more accurate, predictions (Table 5.2). The *sSn* values are decreased to lower than the corresponding values from deterministic EM and (to a lesser extent) SEAM. This is a side-effect of predicting fewer sites overall: 'borderline' predictions which may have been classified as true positive results previously are now classed as false negative results due to the more cautious predictor. However, as with the realistic synthetic data results, the *sSn* and *sPPV* values for MITSU are now less skewed towards improving *sSn* at the expense of decreasing *sPPV*. Although MITSU uses a Bayes-optimal classifier for site prediction, the results of the *E. coli* tests here suggest that a better balance between *sSn* and *sPPV* may be achieved with a different predictor. However, the complexity of the computational problem and the wide structural variety of TFBS motifs may mean that it is not possible to improve on all performance measures in all cases.

MITSU is observed to be particularly effective in cases where the deterministic EM-based algorithm returned poor results, for example, in the *E. coli* FruR, RcsB and TorR motifs. Figure 5.6 displays ROC curves for the *E. coli* TorR motif as discov-

ered by both deterministic EM and MITSU. This motif was very poorly discovered by the deterministic EM algorithm ($sSn = 0.10$, $sPPV = 0.03$, AUC $= 0.83$); however, MITSU increases performance over all measures ($sSn = 0.30$, $sPPV = 0.50$, AUC $= 0.98$). As noted above, fewer sites are predicted overall, reducing the number of false positive results and therefore increasing *sPPV*. The improvement in *sSn* is a result of an improved motif model which better fits the known occurrences: sequence logos representing the motifs discovered by both algorithms are shown in Figure 5.7. The sequence logos clearly show that the model discovered by MITSU is stronger than that discovered by deterministic EM. As discussed in Section 5.3.2, the discovery of a stronger motif model may be partially due to MITSU optimising an entropy-based objective function rather than a likelihood-based function, as in deterministic EM. Further analysis reveals that deterministic EM discovers an AT-rich background model. The motif model discovered by deterministic EM does contain a number of positions which are significantly different from this background model (for example, probabilities for both the G and T nucleotides in the final four positions are significantly higher than those in the background model); however, the information content at these positions is still relatively low. In contrast, optimisation of the entropy-based function within MITSU results in the discovery of motif positions where the probability mass is more concentrated, leading to fewer false positive predictions. In the case of the TorR motif, the fact that deterministic EM performed so poorly suggests that the dataset may have a significant number of suboptimal local maxima in the likelihood function; the stochastic nature of MITSU may therefore play a more important role in its success on this particular dataset.

The deterministic EM *sSn* results for the *E. coli* 'high conservation' and 'overall' groups (Table 5.2) are shown to be significantly higher than those of MITSU ($p = 5.31 \times 10^{-3}$ and $p = 6.03 \times 10^{-3}$, respectively). Similarly, the SEAM *sSn* results for the 'high conservation' and 'overall' groups are significantly higher than those of MITSU ($p = 4.49 \times 10^{-2}$ and $p = 2.95 \times 10^{-2}$, respectively). The SEAM *sPPV* results for the 'high conservation' and 'overall' groups are significantly higher than those of deterministic EM ($p = 1.22 \times 10^{-3}$ and $p = 1.03 \times 10^{-3}$, respectively). The MITSU *sPPV* results for all three groups are significantly higher than those of deterministic EM ($p = 4.88 \times 10^{-4}$, $p = 1.17 \times 10^{-2}$ and $p = 1.09 \times 10^{-4}$ for the 'high conservation', 'low conservation' and 'overall' groups, respectively). Finally, the MITSU AUC result for the 'overall' group is significantly higher than that for deterministic EM ($p = 3.54 \times 10^{-2}$).

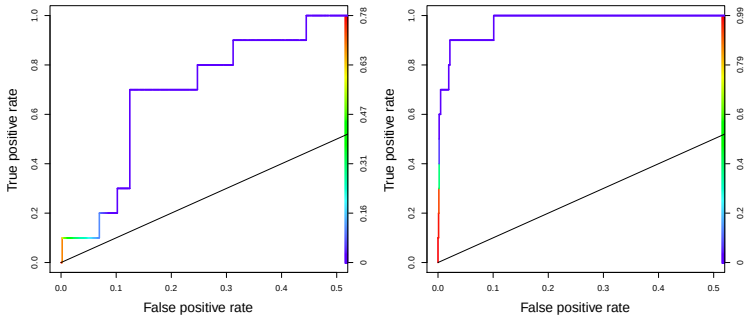

Figure 5.6: ROC curves (plotted for $0 \leq sFPR \leq 0.5$) for the *E. coli* TorR motif discovered by the deterministic EM algorithm (left) and MITSU (right). Curve colour illustrates the threshold of $p(Z_{i,j} = 1|X_{i,j}, \theta)$, from highest (red) to lowest (blue).

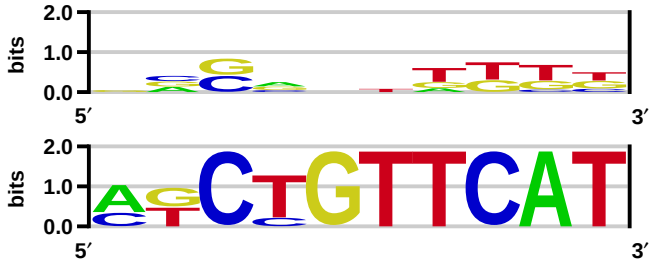

Figure 5.7: Sequence logos representing the *E. coli* TorR motif as discovered by the deterministic EM algorithm (top) and MITSU (bottom).

Table 5.3 shows that for the diverse prokaryotic motifs, MITSU outperforms deterministic EM and SEAM in terms of all three performance measures. As with the realistic synthetic data, the increase in *sPPV* is most dramatic. This result may be of particular interest to biologists, as it means that fewer false positive results are predicted: sites which are predicted by MITSU are therefore more likely to be true transcription factor binding site occurrences. As with the *E. coli* motifs above, performance is significantly increased for motifs which were relatively poorly discovered by deterministic EM, for example, the *E. coli* CRP and RutR motifs and the *B. subtilis* Spo0A motif.

The SEAM *sPPV* result on the diverse prokaryotic data (Table 5.3) is shown to be significantly higher than that of deterministic EM ($p = 7.53 \times 10^{-3}$). The MITSU *sSn* result is also shown to be significantly higher than that of deterministic EM ($p = 3.98 \times 10^{-2}$). The MITSU *sPPV* result is significantly higher than those of both deterministic EM and SEAM ($p = 9.09 \times 10^{-3}$ and $p = 2.95 \times 10^{-2}$, respectively).

In order to validate the MCOIN heuristic developed in Section 4.4 in the context of

sEM (particularly MITSU), further tests were carried out in which the MCOIN heuristic was used to determine the most likely motif width from a range of plausible widths. Consistent with the previous MCOIN tests, all widths $\pm$4nt of the experimentally determined motif width were tested. When the true motif width is unknown the performance of MITSU is decreased slightly; the overall results on the *E. coli* datasets and the diverse prokaryotic dataset when using the MCOIN heuristic ($sSn = 0.43$, $sPPV = 0.68$, AUC $= 0.97$ and $sSn = 0.85$, $sPPV = 0.88$, AUC $= 1.00$, respectively) show that MITSU continues to outperform both deterministic EM and SEAM in terms of *sPPV* and AUC, but decreases in sensitivity compared to the previous results (Tables 5.2 and 5.3). The E-value of the resulting multiple alignment was also tested in the context of sEM. As in Section 4.4.3, MCOIN is shown to outperform the E-values estimator in terms of mean absolute error (MAE) for the *E. coli* dataset (2.90 vs. 3.45). However, the mean average error on the diverse prokaryotic dataset was slightly higher for MCOIN, compared to the E-values estimator (2.11 vs. 2.00).

5.4.2 Stochastic EM escapes local maxima

One major motivation for the stochastic EM algorithm is the fact that the deterministic EM algorithm cannot be guaranteed to converge to the global maximum of the likelihood function and may instead converge to a saddle point or local maximum of the likelihood function. While sEM also cannot be guaranteed to converge to the global maximum of the likelihood function, it can be demonstrated that the stochastic perturbations of sEM allow sEM-based algorithms to escape local maxima which trap deterministic EM-based algorithms, in a motif discovery context.

A small dataset was constructed, comprising 10 sequences of 200nt in length, each sequence containing a single occurrence of a perfectly conserved motif of width 8nt (`CTAAATGC`). As before, *E. coli* intergenic sequences extracted from EcoGene were used as background positions. Despite the relative simplicity of the dataset, it is expected that there will be a large number of local maxima in the likelihood function, corresponding to patterns which are better conserved than the background but less well conserved than the motif of interest.

Traces of the values of $G(\phi)$ for two runs of both the deterministic EM algorithm and MITSU are shown in Figure 5.8. Both algorithms are initialised with the same parameter values and allowed to run to convergence. Both traces illustrate one of the major differences between deterministic and stochastic EM: while each iteration of deter-

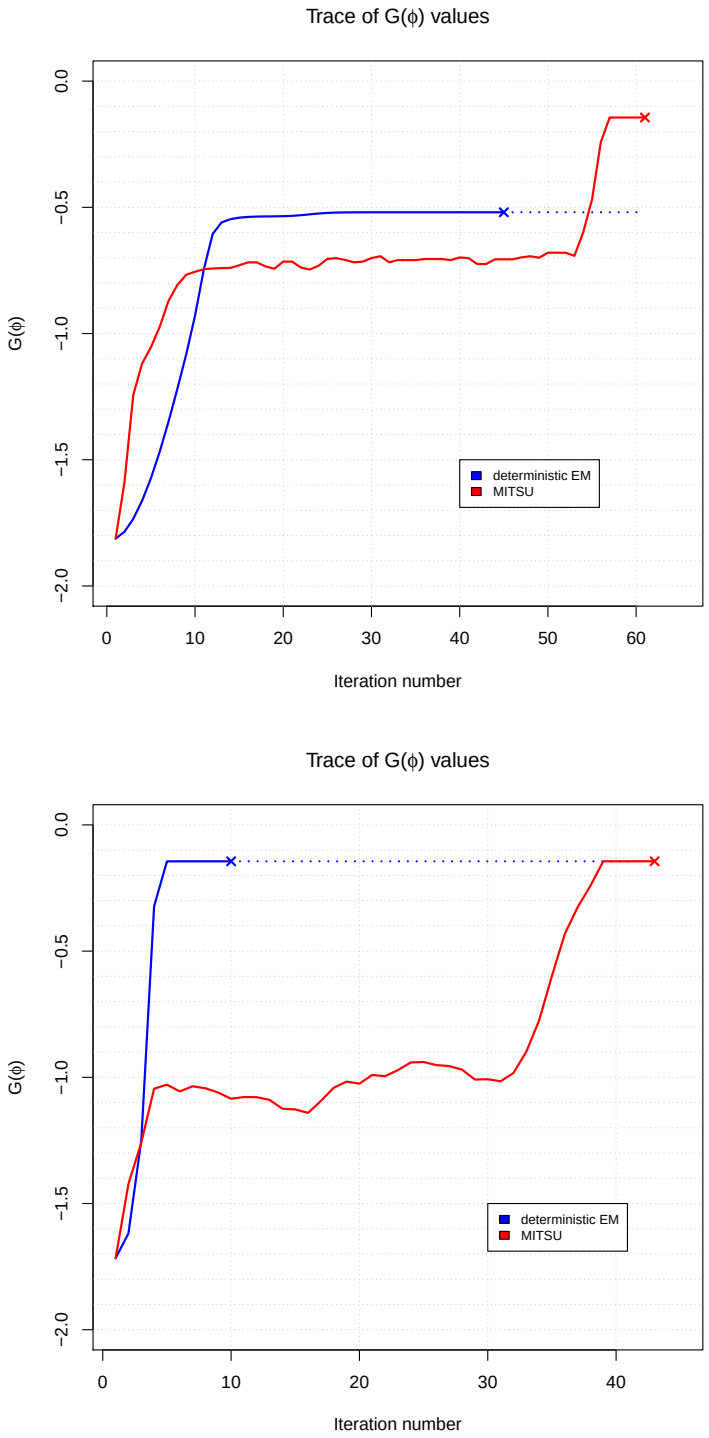

Figure 5.8: $G(\phi)$ traces for two runs of both the deterministic EM algorithm (blue) and MITSU (red) on a synthetic dataset containing a perfectly conserved motif of width 8nt. Algorithm convergence is marked with '$\times$' in both cases. The sampling step of the sEM algorithm allows MITSU to escape local maxima of the likelihood function, which can trap deterministic EM (top).

ministic EM is guaranteed not to decrease the likelihood, at each iteration of stochastic EM, there is a non-zero probability of accepting new model parameters which decrease the likelihood. It is this behaviour which allows stochastic EM to escape local maxima of the likelihood function. The top trace illustrates a case where deterministic EM converges to a local maximum at around $G(\phi) = -0.52$. In contrast, although stochastic EM spends around 40 iterations around $G(\phi) = -0.70$, a small jump which decreases the value of $G(\phi)$ at iteration 53 is followed by several iterations which dramatically increase the value of $G(\phi)$. Using MITSU's stopping rule, sEM converges at $G(\phi) = -0.14$, corresponding to perfect discovery of the known motif.

The lower trace in Figure 5.8 shows a case where both algorithms converge to $G(\phi) = -0.14$. This trace illustrates that deterministic EM generally converges faster than stochastic EM, which can spend a relatively large number of iterations exploring models with lower values of $G(\phi)$ before converging. However, this slower convergence is a small trade-off in exchange for more accurate motif models and binding site predictions, as shown in the top trace.

5.4.3 MITSU successfully discovers multiple occurrences of a motif in a single sequence

As noted in Section 5.3.1, the cut heuristic in combination with the ZOOPS model allows discovery of multiple motif occurrences within a single input sequence. The CRP dataset described in Section 3.6.1 is used to present a proof of principle. Figure 5.9 (top) shows the CRP motif sequence logo constructed from the 24 binding sites in the dataset; it is noted that the low conservation and gapped nature of the CRP motif increases the challenge of computational discovery.

MITSU is compared against MEME and it is assumed that the true motif width is known; both algorithms are run at this width. MITSU was run with the cut length U equal to half the length of each input sequence. The results of this test show that MITSU predicted 28 binding sites ($sSn = 0.71$, $sPPV = 0.61$, AUC $= 0.99$) and successfully predicted both binding sites in the CE1CG, ARA and LAC sequences. The middle logo in Figure 5.9 represents the motif discovered by MITSU. Based on this result, MITSU compares well with MEME, which predicted 18 binding sites and failed to discover more than one site in a sequence using the TCM model when the total number of sites was not provided ($sSn = 0.71$, $sPPV = 0.94$). 14 of the sites predicted by MEME were also predicted by MITSU. The bottom logo in Figure 5.9 represents the

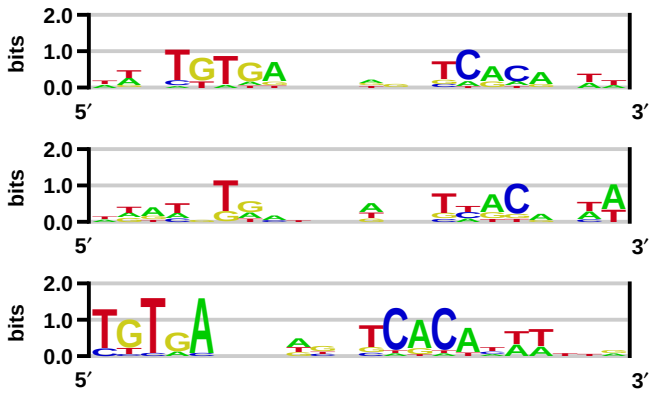

Figure 5.9: CRP motif sequence logos. From top: logo constructed from the 24 binding sites contained in the CRP dataset; logo representing the motif discovered by MITSU; logo representing the motif discovered by MEME when the number of known sites was not provided.

motif discovered by MEME when the number of known sites was not provided. This motif is shifted by 3nt compared to the motif constructed from the known binding sites. When the total number of sites was used as additional information, MEME predicted 24 binding sites and successfully predicted both binding sites in the CE1CG, DEOP2 and MALK sequences ($sSn = sPPV = 0.83$). 16 of the sites predicted by MEME were also predicted by MITSU.

Comparing the sequence logos representing the motifs discovered by MITSU and MEME shown in Figure 5.9, it is noted that the positions in the motif discovered by MITSU are generally underweighted compared to the known motif and that the positions in the motif discovered by MEME are generally overweighted. This would appear to contradict the properties of the entropy-based function optimised by MITSU as discussed previously; that is, for the CRP dataset, MITSU appears to discover a motif model where the probability mass at each position is more diffuse than in the known motif model and the model discovered by MEME. In this case, the difference in weighting appears to be due to the number of sites predicted by each algorithm. Both algorithms return the same number of true positive predictions; the number of false negative predictions is also equal, leading to identical *sSn* results. MITSU predicts more false positive sites than MEME, which leads to an increase in the 'noise' at each position and therefore an underweighting of the positions in the model discovered by MITSU compared to that discovered by MEME. The difference in the number of predicted sites also provides an explanation for the decreased *sPPV* result (0.61

vs 0.94, respectively). While there is room for improvement, the cutting heuristic is shown to allow the successful prediction of multiple motif occurrences within a single input sequence in principle without additional heuristic optimisations to improve performance.

Chapter 6

Further validation and extension of MITSU

This chapter begins by applying the MITSU algorithm developed in Chapter 5 to motif discovery in Alphaproteobacteria. MITSU is validated on previously characterised Alphaproteobacterial motifs before being applied to motif discovery in uncharacterised data. The results of these tests are discussed in detail and suggest several extensions of MITSU. While motif discovery generally aims to discover a single motif from a given dataset, there are many examples of multiple different motifs situated in the same upstream sequence. Section 6.2 presents a method for the discovery of multiple motifs based on a sequential discovery strategy and implements it within MITSU. A higher order Markov model which provides an improved approximation to the background data is described in Section 6.3. The effects of incorporating this higher order model in a deterministic EM-based algorithm are evaluated using *E. coli* and diverse prokaryotic data. For the first time, the effects of incorporating a higher order Markov background model in the context of a stochastic EM-based algorithm (MITSU) are also evaluated. The evaluation of the extensions to MITSU presented in this chapter are used to evaluate the supporting hypothesis, that incorporation of relevant prior biological knowledge through the sEM framework improves motif discovery in terms of previously established metrics.

Finally, a novel information-theoretic measure of motif palindromicity is presented. This measure offers a number of theoretical advantages over current constraint-based approaches; these are discussed in Section 6.4.

6.1 Application to Alphaproteobacteria

In this section, the application of MITSU to discover motifs in the selected Alphaproteobacterial species is described and the results presented. MITSU is first used to discover the previously characterised CtrA motif in *C. crescentus* CB15. MITSU is then used to discover the characterised FnrL motif in *R. sphaeroides* 2.4.1. Following this, MITSU is used to predict a novel motif and consensus sequence for the previously uncharacterised NtrX binding site in the selected Alphaproteobacterial species. Although consensus sequences and lists of regulated genes are available for the characterised motifs, quantitative evaluation is more difficult than in the previous tests (Section 5.4), as the precise location of motif occurrences remains unknown. This difficulty is increased in the case of NtrX, as no consensus sequence has been determined and (as noted in Section 3.5.2) although the regulons controlled by NtrX are known, a list of NtrX-controlled genes has also not been determined.

6.1.1 Characterised Alphaproteobacterial motifs

CtrA

MITSU was applied to the CtrA datasets described in Section 3.5.2, testing 100 random seeds and testing at the experimentally determined motif width (16nt) in all cases. Testing the CtrA-cD dataset, motif occurrences were predicted in all 15 sequences. The predicted motif model is shown in Figure 6.1. The predicted motif is a shifted (by 3nt) version of the canonical gapped consensus sequence (`TTAA-n7-TTAAC`). It is noted that positions 4-11 also match with the ungapped consensus sequence proposed by Laub, *et al.* [96]. The motif discovered by applying MITSU to the CtrA-cE dataset is shown in Figure 6.2. The discovered motif does not match either previously proposed consensus sequence. Although relatively well conserved `AA` dinucleotides are observed at both ends of the motif, the surrounding positions do not suggest that these match the `AA` dinucleotides in either canonical consensus sequence. Testing further random seeds did not produce a model matching the consensus sequence. Testing the CtrA-cDE dataset, MITSU predicted motif occurrences in 29 sequences; the predicted motif model is shown in Figure 6.3. This model matches an unshifted version of the canonical gapped consensus sequence, although the conservation of the 3' half-site is relatively low.

Laub, *et al.* note that in their study, not all intergenic regions bound by CtrA

Figure 6.1: Sequence logo representing the motif model predicted by MITSU, using the *C. crescentus* CtrA-cD dataset.

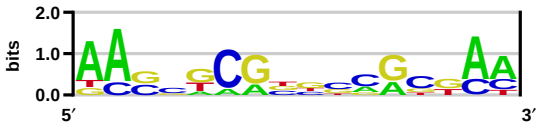

Figure 6.2: Sequence logo representing the motif model predicted by MITSU, using the *C. crescentus* CtrA-cE dataset.

were found to contain a close match to either of the proposed consensus sequences; conversely, not all regions containing a consensus sequence were found to bind to CtrA in an *in vivo* study [96]. It has also been noted that some intergenic regions (such as the promoter for the *fliX* gene) only match one half-site (`TTAA`) but still bind to CtrA [122]. Laub, *et al.* conclude that the factors involved in DNA binding of CtrA are still poorly understood [96]. Although the current tests discover one half of the gapped consensus sequence with relatively high conservation in the CtrA-cD and CtrA-cDE datasets, the second half is significantly less well-conserved. The first possible reason for this lack of conservation may be that some sequences only contain one `TTAA` half-site, as noted above. The structure of the gapped consensus sequence provides a second possible reason. The consensus sequence consists of a direct repeat of `TTAA`, separated by a gap of 7nt. It is possible that if, for a given sequence, the 5' half-site is poorly conserved but the 3' half-site is well conserved, the 3' half-site may be picked up by the algorithm as matching the 5' half-sites in other sequences, introducing error to the 3' half-site of the motif model. Either one, or a combination, of these possibilities may reduce the conservation of the 3' half-site of the motif model.

To test the feasibility of this second explanation, the motif sites predicted by MITSU

Figure 6.3: Sequence logo representing the motif model predicted by MITSU, using the *C. crescentus* CtrA-cDE dataset.

and their surrounding sequences were examined. The aligned sequences are shown in Figure 6.4. If the relatively low conservation of the 3' half-site is due to the 3' half-site of some sequences being predicted as the 5' half-site, the *true* 5' half-site should be visible 7nt upstream of the predicted motif start site. Examining the predicted motif sites shows that there are eight sequences with a potential 5' half site 7nt upstream of the predicted motif start site (sequences with at least two positions matching `TTAA` are counted as potential half-sites). Modifying the alignment such that these sequences are aligned with the rest of the predicted motif sites (that is, shifting the sequences by 11nt) yields the alignment shown in Figure 6.5. The sequence logo constructed from the resulting alignment is shown in Figure 6.6. Comparing with the previous sequence logo (Figure 6.3), it is observed that the 5' half-site of the modified alignment is less well-conserved than that of the original alignment; however, the final two positions in the alignment are now better conserved and match the gapped consensus sequence. It is also noted that the positions in the 'gap' are now less well-conserved, as may be expected. While it is possible that the modified alignment is also suboptimal, it suggests that this explanation of the relatively low conservation of the 3' half-site is feasible. As noted above, the gapped CtrA consensus sequence contains a direct repeat; this result suggests that the computational discovery of direct repeats may be more difficult than that of inverted repeats (or palindromes).

```
                  ttaa       TTAA---n7--TTAAC

>CC_0233   cgccttgtagcgcgacgtgg ttgacgtcctctttac gtgtggtggcgcatcaaggg
>CC_0378   aagcgcctgaaaggccgtgg ttaacggcccgctaac cacgtctctcaacaccggat
>CC_0430   cgggattatcacccttcatg ttggcgatccgtgggc acggagcgctggttggaggg
>CC_0953   tgaggcttc ttaa tgccgga ttaaccctgtttcttc acgacatccctcttttcggc
>CC_1035   ggcgtggcgcgcggttatgc ttaaccacgcgttaag tttggagccccaaaccggac
>CC_1457   gcaagctaggcgaacacgtg ttaacgccaccttgcc accccgaacgaccgcgcttg
>CC_1458   ggaggccttagactttgctg ttaaccatgttttgag gcgctatgcttaactgagtc
>CC_2062   ttgcccggc aaaa cacatcg ttaaccatgcttcgcg catgagtacgggtatagatt
>CC_2063   tactcatgcgcgaagcatgg ttaacgatgtgttttg ccgggcaattggtcgcaggc
>CC_2552   accagcccg gcaa ggtttga ttaaccctccgacccg actctccgccc
>CC_2628   ttcgaaccccgtcgcggggc ttaaccatttttttcga gaccttggggctatccctca
>CC_2868   ttccctaga ttgt atttcgg ttgacgtctcgtctac gatttcgaggttaatgttct
>CC_2949   tggccgcaagcctgtggatt ttagtcttttgttgac caaagaaa
>CC_3286   gcgtcgccgcgctaatgaac ttcatcgtttgttaag tatgagcgcgggaccctagg
>CC_3599   cggcccgattattgcaggcg ttcgcgacccgttaac cagaaccgctcttcaatggc
>CC_0232   ggcgcggccacgattcgtgg ttaacgccccttgatg cgccaccacacgtaaagagg
>CC_0350   tcgcactttt gaa cgcttgc ttaaggctgtcgtggg accgtgcgcgctgttctcgc
>CC_0792   gtcgagagcatctctattca ttgatgttcgtttcac gattaacaaaatagcttcaa
>CC_0793             aactcgtgagt ttaagcgcgcctgcag ggtaaacatatattatcgtt
>CC_1101   gctcgtgca atta accagaa ttcaccacgcgtgagt acaccgccttaaccatcgcg
>CC_1307   tggtaaccaacattcagaaa ttgacgttccgattca cagattaaaaacgcatcgct
>CC_1850   caagtgaacgtcgttcgttc ttaacgcctcgtaaga actctacagcttaggctgca
>CC_1963   gagggccgcgccttttcgt ttcaggccgcgtcgcg cgctccttgcaacctgttaa
>CC_2324   cgcatcctcgacggttaggc ttaatgattgtttgag ccaggaagctgtggattaac
>CC_2640   tcacgttgacgccgggaata ttgacgttgtgaccgc ccccgcagcagcccgtcggc
>CC_2948   attaagtgcagtcggcgaaa ttgatcgtgttttcgg gggtgagcaagtcgaacggc
>CC_3219   cttccgtcg caaa cagcttg ttcgccacgcgcttcg caacaaagacctttccacg
>CC_3295   cgcctggac taac ggttcct ttaaggtttcggtgag gttactggcggcttaaggcg
>CC_3317   ctgccgttccgcgaactgcg ttagggtcttgttaac cgcgcgccgcgaaaatgtga
```

Figure 6.4: Alignment of predicted CtrA motif occurrences. Predicted occurrences are printed in bold. Potential true 5' half-sites are observed in eight sequences (bold).

```
                  ttaa       TTAA---n7--TTAAC

>CC_0233   cgccttgtagcgcgacgtgg**ttgacgtcctctttac**gtgtggtggcgcatcaaggg
>CC_0378   aagcgcctgaaaggccgtgg**ttaacggcccgctaac**cacgtctctcaacaccggat
>CC_0430   cgggattatcacccttcatg**ttggcgatccgtgggc**acggagcgctggttggaggg
>CC_0953              tgaggcttc**ttaa**tgccgga**ttaaccctgtttcttc**acgacatccctcttttcggc
>CC_1035   ggcgtggcgcgcggttatgc**ttaaccacgcgttaag**tttggagccccaaaccggac
>CC_1457   gcaagctaggcgaacacgtg**ttaacgccaccttgcc**accccgaacgaccgcgcttg
>CC_1458   ggaggccttagactttgctg**ttaaccatgttttgag**gcgctatgcttaactgagtc
>CC_2062              ttgcccggc**aaaa**cacatcg**ttaaccatgcttcgcg**catgagtacgggtatagatt
>CC_2063   tactcatgcgcgaagcatgg**ttaacgatgtgttttg**ccgggcaattggtcgcaggc
>CC_2552              accagcccg**gcaa**ggtttga**ttaaccctccgacccg**actctccgccc
>CC_2628   ttcgaaccccgtcgcggggc**ttaaccatttttttcga**gaccttggggctatccctca
>CC_2868              ttccctaga**ttgt**atttcgg**ttgacgtctcgtctac**gatttcgaggttaatgttct
>CC_2949   tggccgcaagcctgtggatt**ttagtcttttgttgac**caaagaaa
>CC_3286   gcgtcgccgcgctaatgaac**ttcatcgtttgttaag**tatgagcgcgggaccctagg
>CC_3599   cggcccgattattgcaggcg**ttcgcgacccgttaac**cagaaccgctcttcaatggc
>CC_0232   ggcgcggccacgattcgtgg**ttaacgccccttgatg**cgccaccacacgtaaagagg
>CC_0350              tcgcactttt**gaa**cgcttgc**ttaaggctgtcgtggg**accgtgcgcgctgttctcgc
>CC_0792   gtcgagagcatctctattca**ttgatgttcgtttcac**gattaacaaaatagcttcaa
>CC_0793            aactcgtgagt**ttaagcgcgcctgcag**ggtaaacatatattatcgtt
>CC_1101              gctcgtgca**atta**accagaa**ttcaccacgcgtgagt**acaccgccttaaccatcgcg
>CC_1307   tggtaaccaacattcagaaa**ttgacgttccgattca**cagattaaaaacgcatcgct
>CC_1850   caagtgaacgtcgttcgttc**ttaacgcctcgtaaga**actctacagcttaggctgca
>CC_1963   gagggccgcgccttttttcgt**ttcaggccgcgtcgcg**cgctccttgcaacctgttaa
>CC_2324   cgcatcctcgacggttaggc**ttaatgattgtttgag**ccaggaagctgtggattaac
>CC_2640   tcacgttgacgccgggaata**ttgacgttgtgaccgc**ccccgcagcagcccgtcggc
>CC_2948   attaagtgcagtcggcgaaa**ttgatcgtgttttcgg**gggtgagcaagtcgaacggc
>CC_3219              cttccgtcg**caaa**cagcttg**ttcgccacgcgcttcg**caacaaagacctttccacg
>CC_3295              cgcctggac**taac**ggttcct**ttaaggtttcggtgag**gttactggcggcttaaggcg
>CC_3317   ctgccgttccgcgaactgcg**ttagggtcttgttaac**cgcgcgccgcgaaaatgtga
```

Figure 6.5: Modified alignment of predicted CtrA motif occurrences. The new alignment includes eight shifted sequences corresponding to identified potential 5' half-sites.

Figure 6.6: Sequence logo constructed using the 29 *C. crescentus* CtrA motif occurrences predicted by MITSU, with 8 sequences shifted by 11nt (see main text)

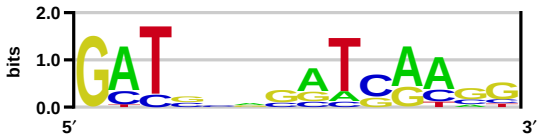

Figure 6.7: Sequence logo representing the motif model predicted by MITSU, using the *R. sphaeroides* FnrL-63 dataset.

FnrL

MITSU was applied to the FnrL-63 dataset described in Section 3.5.2, testing 100 random seeds and testing at the experimentally determined motif width (14nt). Motif occurrences were predicted in 60 sequences; three sequences were predicted not to contain an occurrence of the motif. The predicted motif model is shown in Figure 6.7. The resulting model matches well as a shifted (by 2nt) version of the canonical consensus sequence (`TTGAT-n4-ATCAA`). It is noted that the `GAT` trinucleotide at the 5' end of the motif is very well conserved. If the `TT` positions at the 5' end of the canonical consensus sequence represented positions which were less well conserved than the `GAT` trinucleotide, this may provide an explanation for MITSU returning a shifted version of the motif (testing further random seeds did not improve the model): the well conserved `GAT` trinucleotide may present the algorithm with a local maximum in the likelihood function from which escape is unlikely. As noted above, at each iteration of the algorithm there is a non-zero probability of decreasing $G(\phi)$. Escaping this local maximum should therefore be possible in theory; however, in practice, this may require several successive iterations in which samples decreasing the value of $G(\phi)$ are drawn.

The promoter regions of genes with locus IDs RSP_0105, RSP_3641 and RSP_3643 (with gene product annotations of 'NADH dehydrogenase subunit G', 'putative PfkB family carbohydrate kinase' and 'hypothetical protein', respectively) were predicted by MITSU not to contain a motif occurrence. None of these genes are part of the predicted core FNR regulon well conserved in Alphaproteobacterial species; however, these genes have been determined to correspond to the experimentally determined FnrL regulon in *R. sphaeroides* and therefore would be expected to contain a motif occurrence. In particular, the absence of a motif occurrence upstream of the RSP_0105 gene is unexpected, as motif occurrences were predicted in the sequences upstream of other NADH dehydrogenase subunits (subunits A, B, D, E, H and N). While it remains unclear why motif occurrences should not be predicted in these sequences, it may be

Figure 6.8: Sequence logo representing the motif model predicted by MITSU, using the *R. sphaeroides* FnrL-20 dataset.

reasonable to suggest that potential motif occurrences in these sequences matched the predicted model less well and therefore did not score highly enough in order to pass the classification threshold (Equation 5.35, page 138).

MITSU was then applied to the FnrL-20 dataset described in Section 3.5.2, again testing 100 random seeds and testing at the experimentally determined motif width (14nt). Motif occurrences were predicted in 18 sequences; two sequences were predicted not to contain an occurrence of the motif. The predicted motif model is shown in Figure 6.8. The resulting model matches very well with the canonical consensus sequence.

The upstream sequences of genes with locus IDs RSP_0465 and RSP_0690 (with gene product annotations of 'putative heavy metal translocating P-type ATPase' and 'peptidase U32 family', respectively) were predicted by MITSU not to contain a motif occurrence. Unlike the sequences predicted not to contain a motif occurrence in the FnrL-63 dataset, these genes are part of the predicted core FNR regulon well conserved in Alphaproteobacterial species and therefore would be expected to contain a motif occurrence. Further, the sequences predicted not to contain a motif occurrence in the FnrL-20 dataset were predicted to contain a motif occurrence as part of the FnrL-63 dataset. Together, this lends weight to the above suggestion that these sequences do indeed contain motif occurrences, but the occurrences did not score highly enough to be classified as predicted sites. It is noted that, in general, the scores of motif occurrences in the FnrL-20 dataset are higher than those in the FnrL-63 dataset; however, this should not be surprising, since the number of positions which are better conserved has also increased (that is, the two poorly conserved positions at the 3' end of the FnrL-63 motif are replaced with two highly conserved positions at the 5' end of the FnrL-20 motif).

MITSU performs well on both FnrL datasets in comparison to deterministic EM, which predicts large numbers of false positive results as a result of poorly discovering the FnrL motif. As in the evaluation of MITSU in Section 5.4, the deterministic EM-based algorithm is a reimplementation of the original MEME algorithm [10]. De-

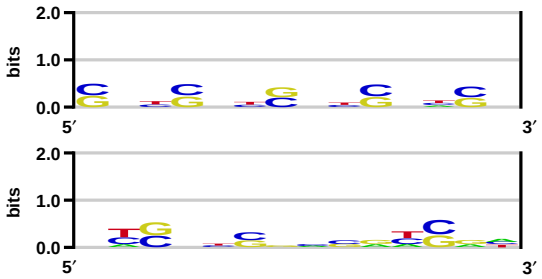

Figure 6.9: Sequence logos representing the motif models predicted by deterministic EM, using the *R. sphaeroides* FnrL-63 (top) and FnrL-20 (bottom) datasets.

terministic EM predicts 245 motif occurrences in the FnrL-63 dataset and 88 motif occurrences in the FnrL-20 dataset. Sequence logos representing the motifs discovered by deterministic EM in both datasets are shown in Figure 6.9. Although the motif discovered in the FnrL-20 dataset can be matched with the canonical consensus sequence (in particular, the `TG` dinucleotide in positions 2 and 3 and the `TC` dinucleotide in positions 11 and 12), the low conservation means that it matches a large number of subsequences within the dataset. As noted in the discussion of the *E. coli* TorR motif in Section 5.4, MITSU's discovery of a stronger motif model may be partially due to the optimisation of an entropy-based function rather than a likelihood-based function as in the deterministic EM-based algorithm. Again, a number of positions in the motif models discovered by deterministic EM are significantly different to those in the background model; however, the information content at these positions (and throughout the models as a whole) is generally very low. In contrast, the positions of the FnrL motif discovered by MITSU have a significantly more concentrated probability mass, which suggests that the entropy function optimised by MITSU has played a role in the discovery of the motifs represented in Figures 6.7 and 6.8. However, as with the *E. coli* TorR motif, the very poor results of the deterministic EM algorithm on these datasets also suggests that there are a large number of local maxima which may trap deterministic EM and therefore that the stochastic nature of MITSU plays an important role in the motif discovery. The size of the FnrL-63 dataset in particular may introduce a significant number of local maxima; datasets such as these demonstrate the power of using a stochastic EM approach which can escape the majority of these local maxima.

A BLAST protein search (performed as described in Section 3.5.2) showed that genes homologous to the *R. sphaeroides* 2.4.1 *fnrL* gene (RSP_0698) were not found in the species in the *Orientia* or *Zymomonas* genera. As noted in Section 3.5.1, the *Ehrlichia*, *Rickettsia* and *Wolbachia* genera (all in the order Rickettsiales) are known

not to possess proteins in the CRP/FNR protein superfamily. Based on the lack of homologous genes in *Orientia* species and the phylogenetic tree created in Section 3.5.1 (Figure 3.5), it may be concluded that the *Orientia* genus is similarly not regulated by FnrL. It is noted that *Orientia* is also in Rickettsiales; further testing may indicate whether or not this conclusion may be extended to the entire Rickettsiales order.

6.1.2 NtrX

MITSU was applied to the NtrX datasets described in Section 3.5.2, again testing 100 random seeds each time. The data was tested at all widths between 12nt and 20nt and the MCOIN heuristic (described in Section 4.4) used to determine the most likely motif width in each case. The results of the tests are summarised in Table 6.1.

Dataset	Consensus sequence	Predicted sites/sequences
Nar-n6	TTGATC-N3-ATCAA	6/6
Cco-n10	TTGAT-N4-ATCAA	9/10
Nif-n8	CCANNCNATATC	8/8
Cyd-n12	SCKSCRNSRSGY	11/12
nitrogen	TGATC-N4-TCAA	22/24
cytochrome oxidase	TTGAT-N4-ATCAA	14/22
NtrX	TGAT-N4-ATCAA	36/46

Table 6.1: Summary of NtrX dataset tests. The determined consensus sequence is given for each tested dataset, along with the number of predicted motif occurrences.

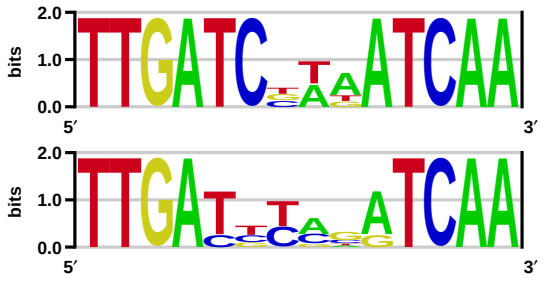

Figure 6.10: Sequence logos representing the motif models predicted by MITSU, using the Nar-n6 (top) and Cco-n10 (bottom) datasets.

Tests were initially performed on the datasets for individual regulons. Tests on the Nar-n6 dataset returned a 14nt motif; the sequence logo for this motif is shown in

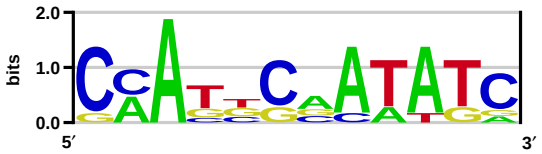

Figure 6.11: Sequence logo representing the motif model predicted by MITSU, using the Nif-n8 dataset.

Figure 6.10 (top). The motif is predicted to occur in all six sequences in the dataset. The consensus sequence is deemed to be TTGATC−N3−ATCAA. Tests on the Cco-n10 dataset return a similar 14nt motif, shown in Figure 6.10 (bottom). The motif is predicted to occur in nine of the ten sequences and the consensus sequence is deemed to be TTGAT−N4−ATCAA. The sequence without a predicted occurrence is that of the *R. etli* str. CFN 42 *fixN* gene, which is shown to have high sequence similarity to the other *ccoN* genes (including the *cco* gene in the same species). However, it is possible that, despite the high sequence similarity, the *fixN* gene (which is found in *R. etli* plasmid F) has a different functionality from the *ccoN* gene (which is found in plasmid D). It is noted that in both cases, the discovered motifs are very similar to the canonical FnrL motif TTGAT−N4−ATCAA. The sixth position in the Nar-n6 motif is the main difference, as this is not conserved in the FnrL motif but is perfectly conserved in this case.

Tests on the Nif-n8 dataset returned the 12nt motif shown in Figure 6.11. The motif is predicted to occur in all input sequences. The consensus sequence is deemed to be CCANNCNATATC. While the motif discovered in this dataset is weaker than those for the Nar-n6 and Cco-n10 datasets, the fact that predicted sites are present in all input sequences suggests that this motif may be worth investigating further.

Tests on the Cyd-n12 dataset returned the 12nt motif shown in Figure 6.12. The motif is predicted to occur in 11 out of 12 sequences. As with the motif discovered in the Nif-n8 dataset, the predicted motif is weaker in this case, although there are two perfectly conserved positions. The consensus sequence SCKSCRNSRSGY gives a weak indication of a possible gCggCg inverted repeat. The sequence without a predicted motif occurrence is that of the *C. crescentus* str. CB15 *cydD* gene. The gene product annotation for this gene (provided in Table A.10) does not suggest why no motif should be predicted in this sequence, although it is possible that a subsequence within this sequence partially matched the motif but did not score highly enough to be predicted. Again, the fact that the motif is predicted in almost all input sequences

Figure 6.12: Sequence logo representing the motif model predicted by MITSU, using the Cyd-n12 dataset.

suggests further investigation is required.

Tests on the other datasets (Nir-n4, Nor-n3 and Nos-n3) did not provide any conclusive results. This is largely attributable to the small size of the datasets and is partially attributable to the sequences themselves. For example, in the case of the Nir-n4 dataset, the 200nt upstream sequences for both *Brucella* species are exactly the same. This results in MITSU converging to a portion of these sequences corresponding to the tested width and ignoring the other two sequences in the dataset. A perfectly conserved motif is therefore predicted, but the algorithm cannot be said to have discovered a motif of significance. Similarly, in the Nos-n3 dataset, there is a 25nt sequence which is perfectly conserved between the two *R. palustris* sequences. MITSU again predicts a perfectly conserved model which corresponds to a portion of this sequence but is not found in the remaining sequence. In both cases, the small size of the dataset means that it is not possible to draw any conclusions from these results.

Following tests on the individual datasets, two combination datasets were created as described in Section 3.5.2. In tests on the 'nitrogen' dataset (the union of the Nar-n6, Nir-n4, Nor-n3, Nos-n3 and Nif-n8 datasets), MCOIN predicts the motif with width 13nt to be most likely (Figure 6.13). The motif is predicted to appear in 22 out of 24 sequences. As with the test on the Nar-n6 dataset (Figure 6.10, top), the motif partially matches the known FnrL motif. Notably, the motif matches sequences where no motif was discovered previously (for example, in the Nir-n4, Nor-n3 and Nos-n3 datasets). There are two possible reasons for this. Firstly, the motif is weaker overall and so has an increased likelihood of matching subsequences within the dataset. The second possible reason lies in the iterative method used by the algorithm to update the motif model. Lawrence, *et al.* have noted (in relation to their Gibbs sampling-based algorithm) that the more accurate the samples in the first step of the algorithm, the more accurately the locations of motif occurrences can be determined in the second step and vice versa [98]. That is, once some correct motif locations have been sampled in the first step, this has the effect of 'gathering' additional motif occurrences, improving the

Figure 6.13: Sequence logo representing the motif model predicted by MITSU, using the nitrogen dataset.

Figure 6.14: Sequence logo representing the motif model predicted by MITSU, using the cytochrome oxidase dataset.

discriminative power of the algorithm. As MITSU also samples motif positions in the S-step of the sEM algorithm, it is likely that a similar process occurs. In this case, the strong motif occurrences in the Nar-n6 dataset may help to gather additional motif occurrences from the other datasets.

Similarly, in tests on the cytochrome oxidase dataset (the union of the Cco-n10 and Cyd-n12 datasets), MITSU predicts the 14nt motif shown in Figure 6.14. The motif is predicted to appear in 14 of the 22 sequences; again, the influence of the Cco-n10 dataset appears to be useful in gathering additional motif occurrences within the Cyd-n12 sequences. As with the Cco-n10 dataset, the consensus sequence is deemed to be `TTGAT-N4-ATCAA`, identical to the FnrL motif.

Finally, MITSU is tested on all NtrX input sequences (the union of the nitrogen and cytochrome oxidase datasets). MITSU predicts the 13nt motif shown in Figure 6.15. Again, this motif is a partial match to the canonical FnrL motif. Motif occurrences are predicted in 36 out of 46 sequences, matching the occurrences predicted when the nitrogen and cytochrome oxidase datasets were tested separately; in this case, there appears to be no 'gathering' of additional motif occurrences as described above. It is notable that the widths predicted by MCOIN for the motifs discovered in the nitrogen dataset and the dataset consisting of all NtrX input sequences are 13nt rather than the 14nt of the canonical FnrL motif. There are two possible explanations for this. Firstly, it is noted that without further experimental testing, it is not clear whether the motifs discovered in the nitrogen and cytochrome oxidase datasets are both instances of the FnrL motif, or an FnrL-like motif. It is possible that the motifs are subtly different (in terms of both the number of positions and their conservations) which may account

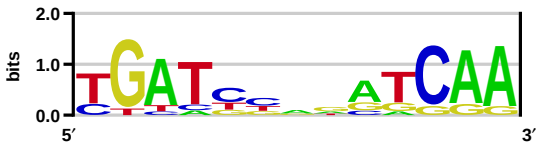

Figure 6.15: Sequence logo representing the motif model predicted by MITSU, using all NtrX input sequences.

for the widths chosen by MCOIN. Secondly, assuming that the motif is indeed the FnrL motif, in the case of the nitrogen dataset, the missing 5' T nucleotide is poorly conserved relative to the rest of the motif; the 13nt motif is predicted as being more significant. This lack of conservation may also explain the 13nt model prediction made when the nitrogen dataset is combined with the cytochrome oxidase dataset as the NtrX dataset.

The most striking result from tests on NtrX-regulated genes is the discovery of motifs very similar to the canonical FnrL motif (Figures 6.10, 6.13, 6.14 and 6.15); in some cases these motifs match the FnrL motif exactly. The role of FnrL in the control of cytochrome cbb_3 oxidase under low-oxygen conditions has previously been noted in *R. sphaeroides* and *R. capsulatus* by Mouncey and Kaplan [126] and Swem and Bauer [163], respectively. However, the role of FnrL in the regulation of denitrification genes remains unclear. The results of tests on the Nar-n6 and nitrogen datasets show a very strong consensus sequence which strongly suggests that FnrL plays a role in controlling the expression of denitrification genes. In the case of the Nar-n6 dataset, the discovered consensus sequence does not exactly match the FnrL consensus sequence as the sixth position in the discovered motif (Figure 6.10, top) is perfectly conserved. However, the strength of this discovered motif strongly suggests that FnrL, or an FnrL-like regulator, also plays a role in the regulation of nitrate reductase genes; given the role of FNR as a transcriptional regulator for many genes involved in anaerobic metabolism, this is a logical conclusion.

Although the FnrL motif was often predicted, the other motifs predicted by MITSU may be good candidates for further investigation. The motif discovered in the Nif-n8 dataset (Figure 6.11), with consensus sequence CCANNCNATATC was slightly weaker in terms of conservation than the FnrL motifs and was not discovered in tests on the nitrogen dataset. However, the fact that motif occurrences were predicted in all sequences in the Nif-n8 dataset suggests that this motif may be worth investigating further. Similarly, the motif discovered in the Cyd-n12 dataset (Figure 6.12) is predicted

in almost all input sequences and may be a good candidate for further investigation.

As noted above, FnrL has previously been shown to play a role in the regulation of cytochrome oxidase; the motifs discovered in the tests in this section confirm that an FnrL (or FnrL-like) motif is present in the promoter regions of cytochrome oxidase genes. However, as noted in Section 3.5.2, Dahouk, *et al.* have determined two cytochrome oxidase regulons (*cyd* and *cco*) to be regulated by NtrX. Together, these suggest that cytochrome oxidase genes are regulated by more than one transcription factor; these transcription factors may be working cooperatively. It follows that the promoter regions of cytochrome oxidase genes contain more than one motif and should contain motifs for (at least) FnrL and NtrX. The situation for the denitrification genes is less clear; however, the discovery of FnrL motifs in the promoter regions of these genes suggests that the situation is similar to that of the cytochrome oxidase genes.

It is generally assumed that motif discovery algorithms tend to discover the most statistically significant motif within a dataset. It follows that if more than one motif is present in a given dataset, motifs which are better conserved (and therefore statistically more significant) will be discovered over other motifs which are less well-conserved. In the tests carried out in this section, the FnrL motif discovered by MITSU is generally very well conserved (for instance, in the motifs shown in Figures 6.10 and 6.14). A less well-conserved NtrX motif would therefore not be discovered. This motivates a strategy for discovering multiple motifs within a single dataset. Section 6.2 discusses strategies for multiple motif discovery and implements a method for multiple motif discovery within MITSU. The results of multiple motif discovery in the context of NtrX are reported in Section 6.2.3.

6.2 Discovering multiple motifs

This section focuses on the discovery of multiple motifs within a dataset and compares a simultaneous discovery strategy (as used in many purely stochastic approaches) to a sequential discovery strategy (as used by the majority of deterministic approaches). The advantages of each strategy are discussed.

6.2.1 Motivation and strategies for multiple motif discovery

The majority of research in motif discovery is focused on discovering the most statistically significant motif within a dataset. However, it is known that multiple transcription

factors often bind in a combinatorial fashion in order to regulate transcription. Therefore gene promoter regions may contain transcription factor binding sites for multiple transcription factors; in some cases there are multiple binding sites for one or more transcription factors [167]. Generally, motif discovery algorithms only discover one motif for a given dataset; it is assumed that this motif is the most statistically significant. However, since promoter regions may contain binding sites for more than one transcription factor, this raises the possibility of datasets containing additional, possibly less statistically significant, motifs. This motivates a strategy for recovering multiple motifs from a single dataset. As noted in Chapter 2, there are two commonly used strategies: simultaneous and sequential discovery.

Lawrence, *et al.* [98] implement a strategy for simultaneous discovery of multiple motifs in their Gibbs sampling-based algorithm, by initialising multiple PWMs at the start of the algorithm, drawing a full set of samples for each PWM in the sampling step of the algorithm, then updating each PWM separately in the update of the algorithm. Care has to be taken to ensure that the samples drawn from each sequence do not overlap each other. It is argued that this provides additional information to the motif discovery process: information regarding the positions of one motif can be used to aid the discovery of further motifs, as motif occurrences must be non-overlapping. However, this also places a limit on the number of motifs which can be discovered, depending on the length of the shortest input sequence.

A strategy for sequential motif discovery was introduced in the context of EM by Bailey and Elkan [11]. In this strategy, the EM algorithm is run to convergence, discovering a motif model; motif occurrences are then predicted based on the discovered model. These motif occurrences are then probabilistically 'erased' by weighting down positions which correspond to previously discovered motif occurrences. The EM algorithm can then be run again to find a different motif; by effectively erasing the occurrences of the first motif from the dataset, the algorithm can discover second and subsequent motifs without interference from the occurrences of the first. Although sequential motif discovery is slower than simultaneous motif discovery, it is more flexible, as there are no limits to the number of motifs which can be discovered in terms of input sequence length.

Since the publication of both strategies, sequential discovery of multiple motifs has become more common than simultaneous discovery. Although simultaneous discovery of multiple motifs lends itself to a sampling-based approach such as the sEM-based approach used in this project, sequential discovery has increased flexibility and

is much more straightforward to implement, particularly when considering other aspects of the motif discovery algorithm (for instance, determining the most likely motif width). Taking these issues into account, a sequential motif discovery strategy is implemented in MITSU in the following section. It is noted that later sampling-based algorithms also take a sequential approach to multiple motif discovery. For example, the AlignACE algorithm [143], which can be viewed as an extension of Lawrence, *et al.*'s Gibbs sampling-based algorithm, implements a 'masking' approach similar to the probabilistic erasing approach described by Bailey and Elkan.

6.2.2 Method

Here, Bailey and Elkan's method for sequential motif discovery in deterministic EM [11] is implemented in the context of sEM.

An 'erasing' factor $V_{i,j}$ is associated with each width-W subsequence within the input sequences, that is, for $i \in \{1 \ldots N\}$ and $j \in \{1 \ldots L_i - W + 1\}$. Equivalently, each latent variable $Z_{i,j}$ has an associated erasing factor. Each erasing factor is initially set to 1, that is, no erasing has occurred. After sEM has converged, the $Z_{i,j}^{(t)}$ variables represent the expected probability that a motif occurrence starts at $X_{i,j}$. The erasing factors for each position are then decreased based on the converged value of $Z_{i,j}^{(t)}$, so that positions with high values of $Z_{i,j}^{(t)}$ are effectively 'erased' by $V_{i,j}$ and not chosen in future passes of the algorithm.

In order to calculate the $V_{i,j}$ values, another variable $U_{i,j}$ is defined. $U_{i,j}$ is the expected probability that $X_{i,j}$ is not part of a previously discovered motif occurrence and is calculated using:

$$U_{i,j}^{(p)} = U_{i,j}^{(p-1)} \cdot (1 - \max_{l=j-W+1,\ldots,j} Z_{i,l}^{(c)}), \tag{6.1}$$

for $i \in \{1 \ldots N\}$ and $j \in \{1 \ldots L_i - W + 1\}$, where $Z_{i,l}^{(c)}$ are the converged values of the latent data (the expected probability of a motif occurrence starting at $X_{i,j}$) after the pth pass of the algorithm, and $U^{(0)} = 1$ for all positions. Some care needs to be taken to ensure that positions at the start of the sequence are handled appropriately (since j cannot be less than 1). $V_{i,j}$ is then calculated as:

$$V_{i,j}^{(p)} = \min_{l=j,\ldots,j+W-1} U_{i,l}^{(p)}, \tag{6.2}$$

again for $i \in \{1 \ldots N\}$ and $j \in \{1 \ldots L_i - W + 1\}$. As above, care needs to be taken to ensure that positions at the end of each input sequence are handled appropriately (since j cannot exceed $L_i - W + 1$).

Following the calculation of the erasing factor $V_{i,j}$, it may be used to implement multiple motif discovery in the context of sEM as follows. $Z_{i,j}^{(t)}$ is calculated as normal in the S-step of the algorithm (Equation 5.25 in the ZOOPS sequence model). The latent data is then multiplied by the erasing factor to give:

$$\hat{Z}_{i,j} = V_{i,j} \cdot Z_{i,j}. \tag{6.3}$$

The sEM algorithm then proceeds as before, but replacing $Z_{i,j}$ with $\hat{Z}_{i,j}$. That is, $\hat{Z}_{i,j}$ is normalised for each input sequence X_i, before samples are drawn using Equation 5.26. The U-step of the algorithm remains the same, updating the model based on the samples drawn in the S-step. (Equations 5.31-5.32 in the ZOOPS model).

The implementation of this is reasonably straightforward in the ZOOPS case but becomes slightly more complex when the cut heuristic (Section 5.3.1) is used in MITSU. In particular, care needs to be taken at positions within the overlap between two cuts of an input sequence. In MITSU, this is managed by subjecting the erasing variables $U_{i,j}$ and $V_{i,j}$ to the same cutting procedure as the input data $X_{i,j}$ (so that each width-W subsequence still has associated Z, U and V values) and updating the erasing variables based on the values of Z mapped back to their positions in the original (uncut) dataset. The probabilistic erasing method is implemented as the outer loops of the motif discovery algorithm (Section 5.3.5).

6.2.3 Results

The MalI/SoxR dataset described in Section 3.6.2 is used as a proof of principle to demonstrate the ability of MITSU to discover multiple motifs within a single dataset. MITSU was run for two passes (that is, $M = 2$); it is assumed that the true motif widths (12nt and 19nt for the MalI and SoxR motifs, respectively) are known. The results of this test show that MITSU discovers the SoxR motif on the first pass of the algorithm, shifted by 1nt. On the second pass of the algorithm, MITSU discovers the MalI motif, shifted by 2nt. In both cases, two motif sites are predicted, matching the known motif sites; that is, *sSn* and *sPPV* are 1.00 for both motifs. It is noted that MITSU discovers the longer SoxR motif first. As both motifs are perfectly conserved, it is expected that SoxR will be more statistically significant than MalI due to its longer width.

Application to NtrX motif discovery

Analysis of the motif discovered in the NtrX datasets showed that the FnrL motif was often found; this motivated the implementation of a strategy for the discovery of mul-

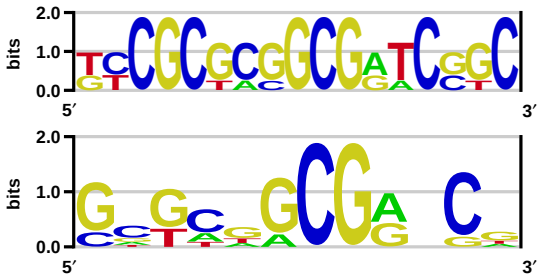

Figure 6.16: Sequence logos representing the motif models predicted by MITSU on the second passes of the Cco-n10 (top) and cytochrome oxidase (bottom) datasets.

tiple motifs. The strategy for multiple motif discovery implemented in MITSU as described above is applied to the NtrX datasets. Where the FnrL (or FnrL-like) motif was discovered on the first pass of MITSU, second or, where appropriate, third passes of MITSU were carried out, erasing previously discovered motif occurrences. The results of these subsequent passes are summarised in Table 6.2.

Dataset	Pass	Consensus sequence	Predicted sites/sequences
Cco-n10	2	KYCGCGCGGCGRTCSGC	5/10
cytochrome oxidase	2	GNKCNGCGRNCN	19/22
Cco-n10	3	NNSTTCGTGMNN	10/10
Nar-n6	2	TTTMRGWTGCCWYYAW	4/6
NtrX	2	SNNNNNKNCGGN	37/46

Table 6.2: Summary of subsequent passes of MITSU on NtrX datasets. The pass number and consensus sequence is given for each tested dataset, along with the number of predicted motif occurrences.

The second pass of MITSU on the Cco-n10 and cytochrome oxidase datasets returned the 17nt motif and the 12nt motif shown in Figure 6.16 (top and bottom, respectively). Motif occurrences are predicted in 5 out of 10 sequences and 19 out of 22 sequences, respectively. As may be expected, MITSU converges to motifs which are slightly weaker in terms of conservation than the motifs discovered on the first pass (Figures 6.10 (bottom) and 6.14, respectively). Although there is no clear consensus sequence, there are some similarities between the motifs. The 9nt sequence GCGGCGANC is observed in both motifs, in positions 6-14 of the Cco-n10 motif and positions 3-11 of the cytochrome oxidase motif. It is noted that this sequence partially matches the motif found in the first pass on the Cyd-n12 dataset (Figure 6.12).

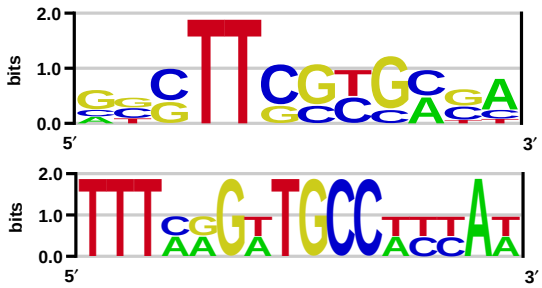

Figure 6.17: Sequence logos representing the motif models predicted by MITSU on the third pass of the Cco-n10 dataset (top) and the second pass of the Nar-n6 dataset (bottom).

This motif had been identified as being a possible weak inverted repeat and the first 6 positions (GCGGCG) of this motif match the first 6 positions of the 9nt sequence above. Inspection of the motif discovered on the second pass of the Cco-n10 dataset (Figure 6.16, top) suggests that positions 6-17 may also be a weak inverted repeat. Evidence for this inverted repeat is not so strong in the motif discovered in the second pass of the cytochrome oxidase dataset (Figure 6.16, bottom). However, the 9nt sequence GCGGCGANC and the possible inverted repeat sequence GCGGCGCGCCGC are candidates for further investigation; experimental testing using site-directed mutagenesis (SDM) and measuring gene expression levels may indicate whether or not these sequences have a role in the regulation of cytochrome oxidase.

Figure 6.17 shows sequence logos representing the motifs discovered by MITSU on the third pass of the Cco-n10 dataset (top) and the second pass of the Nar-n6 dataset (bottom). Motif occurrences are predicted in all 10 sequences in the Cco-n10 dataset and 4 out of 6 sequences in the Nar-n6 dataset. Again, these motifs are generally less well-conserved than the motifs discovered on the first pass. The motif discovered in the Cco-n10 dataset has an 8nt sequence which is reasonably well-conserved (CTTCGTGC); the fact that it is predicted in all input sequences makes this a good candidate for further investigation. The motif discovered in the second pass on the Nar-n6 dataset does not correspond perfectly with the Cco-n10 motif. However, some subsequences of the two motifs match; for example, the TTCG subsequence appears in positions 4-7 of the Cco-n10 motif and positions 2-5 of the Nar-n6 motif and the TGC subsequence appears in positions 8-10 of both motifs. If these two motifs are connected, this may suggest that some conformational change is made to the transcription factor protein, extending the binding site upstream of the *nar* operon.

The motif discovered by MITSU on the second pass of all NtrX input sequences is

Figure 6.18: Sequence logo representing the motif model discovered by MITSU on the second pass of all NtrX input sequences.

shown in Figure 6.18. A clear motif sequence is difficult to determine; however, the motif is predicted to occur in 37 out of the 46 input sequences. It is noted that the final 4 positions (CGGC) provide a partial match with the motifs shown in Figure 6.16. However, despite the partial match and the large number of predicted occurrences, the generally low conservation of the discovered motif means that it is difficult to draw a conclusion for this dataset.

6.2.4 Conclusion

In this section, a case is made for using a sequential strategy based on probabilistic erasing for the discovery of multiple motifs within a single dataset. The MalI/SoxR dataset is used to demonstrate the use of this strategy in principle, in the context of sEM (and MITSU in particular). The strategy is then applied to the discovery of multiple motifs within the NtrX datasets. As noted in Section 2.2.2, the discovery of second and subsequent motifs in a dataset may be affected if the discovery (and therefore erasing) of previous motifs is poor. However, the improved results achieved by MITSU make this issue less important. The ability to discover multiple motifs is important in cases where there genes are thought to be regulated by a combination of transcription factors.

6.3 A higher order Markov background model approximation

Relatively recent motif discovery algorithms such as BioProspector [107] and MotifSampler [166] have replaced the simplistic background models used by older motif discovery algorithms such as MEME and AlignACE with more complex Markov background models. This section introduces a higher order Markov background model as an improved approximation to data which is not part of a motif occurrence. The improvements over a basic multinomial (or ‘frequency’) background model are evaluated,

both as part of deterministic and stochastic motif discovery algorithms. The influence of both the order of the background model and the data used to construct the model is also evaluated and discussed.

6.3.1 Motivation and previous studies

As noted in Section 2.2, motif discovery algorithms such as MEME and AlignACE use a multinomial (or 'frequency') background model, which simply records how many times each nucleotide (A, C, G, or T) appears in the dataset and assumes that every background position is chosen independently and at random with these weightings. This is unlikely to be biologically realistic; for example, CpG islands are regions of DNA well known to have a higher GC-content than could be statistically expected. However, there is currently little biological knowledge on the structure of 'background' sequences which could be used to accurately model these sequences [86]. In the absence of this knowledge and given the linear nature of the data, motif discovery algorithms now often use a Markov model (formally a Markov process with a discrete state space) in order to model the background positions.

A higher order Markov background model takes previous positions into account when assigning probabilities to a particular nucleotide in order to better capture the characteristics of the local DNA environment. That is, while the background probability of a particular nucleotide is constant regardless of position when using the frequency background model, this is not the case when using a Markov background model: the same nucleotide could have different probabilities, given the preceding nucleotide. While it is unlikely that the true biological model is a higher order Markov model, this is generally thought to be an improved approximation to the sequence of nucleotides in background positions, which is likely to be more complex than a Markov model [21]. In the context of probabilistic algorithms for motif discovery (which are generally based on a two-step iterative process, for example, the EM algorithm), algorithmic efficiency is also improved as the background model need only be calculated once. This model is then deemed to be the true background model; the background model parameters are then fixed and reused at each iteration of the algorithm, rather than being initially estimated and then updated at each iteration of the algorithm, as in MEME and AlignACE. Using Liu, *et al.*'s θ_β notation for the background model [107], Algorithm 6.1 demonstrates how a higher order background model may be incorporated within a MEME-type deterministic algorithm, as compared to the original

```
procedure MEME-type algorithm with Markov background model
Create Markov background model θβ
  for N do (where N is the number of motifs to be found)
    for each motif width do
      for each TCM prior parameter value do
        Estimate initial motif model parameters θm, based on width and TCM prior values
        until EM algorithm converges, do
          E-step: reestimate motif position using current θm and θβ
          M-step: reestimate θm using current motif position
        end
      end
    end
    Print discovered motif which maximises motif score
    'Erase' motif from dataset by updating prior
  end
end MEME-type algorithm with Markov background model
```

Algorithm 6.1: Pseudocode demonstrating how a higher order background model may be incorporated within a MEME-type algorithm.

MEME algorithm (Algorithm 2.2, page 23).

As noted in Section 2.2, the MotifSampler algorithm [166] was the first motif discovery algorithm to implement a higher order Markov background model, taking inspiration from gene detection algorithms such as GLIMMER [44], HMMgene [93] and GeneMark.hmm [110]. The motivation for implementing a higher order Markov background model as part of a motif discovery algorithm stems from a need to better model the 'noisy' background sequences upstream of the genes of interest. Considering the large size of the background sequences to the relatively small motif sequences, this seems reasonable. Thijs, *et al.* note in a separate technical paper that the size of the upstream regions is related to the compactness of the genome [165]. This observation suggests that better modelling of upstream sequences may have a more significant impact when searching for motifs in higher eukaryotic species, than in prokaryotic or even lower eukaryotic species (for example, yeast). Thijs, *et al.* also suggest that constructing the background model using intergenic sequences is most appropriate. The effect of using a higher order Markov background model is evaluated on a combination of simulated data and characterised data from the model plant organism *Arabidopsis*

thaliana [166], in particular the G-box regulatory element, which is involved in light-sensitive gene regulation in plants. Thijs, *et al.* show that using a 3rd order Markov background model constructed using intergenic sequence data improves on a 0th order model constructed using the intergenic sequences and a 3rd order model constructed using only the input sequences [166]. However, it is unclear on which statistic this performance increase is measured; it is only noted that over several runs of the algorithm (that is, with different random seeds), using higher order Markov background models increased the number of times the correct motif was discovered. It is concluded that a higher order model can considerably enhance performance in the presence of noisy data; however, as noted above, this increase may be less significant on tests using prokaryotic data.

BioProspector also implements a higher order Markov background model; Liu, *et al.* note that their implementation is motivated by the fact that, in DNA, the presence of a particular nucleotide at a given position usually exerts some influence on the nucleotides at surrounding positions [107]. That is, the frequency model assumption of independence between nucleotides is incorrect. The effect of using a higher order Markov background model is only evaluated on the RAP1 binding site in yeast. However, as with the tests on MotifSampler, it is unclear exactly which statistics are improved by the use of a higher order model. Liu, *et al.* note that the higher order Markov background model reduces the number of false positive motifs, but more detailed results are not provided. It is noted here, however, that this result matches the recent findings of Hartmann, *et al.*, who conclude that using a higher order Markov background model can help to rank down false positive motifs, which are often repetitive, or have a biased nucleotide composition [78]. While the deterministic algorithms MEME, COMODE and cosmo have adopted higher order Markov background models for motif discovery, their authors have provided little further testing in order to evaluate the power of these models. In their evaluation of motif discovery algorithms using prokaryotic (*E. coli*) data, Hu, *et al.* perform tests to evaluate the effect of Markov background models using BioProspector, MEME and MotifSampler [81]. Most notably (and unexpectedly), it was discovered in a test of 70 *E. coli* TFBS motifs that the order of the Markov background model was not found to have a significant effect on the nucleotide-level performance coefficient (nPC)[1]. For example, Hu, *et al.* demonstrate that MEME achieves similar levels of performance when using 1st, 2nd and 3rd order background models constructed using the full *E. coli* genome sequence. Similar

[1]Recall that nPC is calculated as $nTP/(nTP + nFN + nFP)$.

results were also noted in tests on BioProspector and MotifSampler. It is perhaps less surprising that using different data sources to construct the background model led to different prediction accuracy. BioProspector and MEME are shown to achieve higher *nPC* values when using the full *E. coli* genome (by around 8%), while MotifSampler is shown to achieve higher *nPC* values when using a background model constructed using intergenic sequences (by around 4%). However, it is noted here that *nPC* values are shown to be very low in general, typically around 0.15.

As noted in Section 3.4, there are noticeable differences between the nucleotide (and also dinucleotide and trinucleotide) compositions of the full *E. coli* genome and the set of *E. coli* intergenic sequences. It follows that the higher order Markov background models constructed using *E. coli* intergenic sequences and the full *E. coli* genome sequence are very different; even in the 0th order portion of the model, differences are clear. For example, GC-content is around 51% for the full genome, but only around 40% in intergenic sequences. Similar differences are also noticeable in higher order models. If it may be assumed that the background data to be separated from the motif instances is from intergenic regions, it seems intuitive that background models constructed using intergenic data will model this data better than background models constructed using the full genome sequence.

6.3.2 Methods

Constructing background models

Traditional multinomial (or 'frequency') background models (in the context of Markov background models, these are known as 0th order models) are constructed by scanning the dataset, counting the number of occurrences of each nucleotide $k \in \mathcal{L}$ and normalising such that the parameters of the multinomial model sum to 1 and therefore represent the prior probability of observing each nucleotide within the dataset. That is,

$$\sum_{k \in \mathcal{L}} \theta_{0,k} = 1. \tag{6.4}$$

Higher order Markov background models are constructed in a similar way. A 1st order Markov model is constructed by scanning the data and counting the number of occurrences for each dinucleotide $\{\texttt{AA}, \texttt{AC}, \texttt{AG}, \ldots, \texttt{TT}\}$. These counts are then normalised such that the probabilities of seeing any nucleotide given a particular preceding nucleotide sum to 1. That is,

$$\sum_{k \in \mathcal{L}} p(k|n) = 1, \tag{6.5}$$

for any preceding nucleotide $n \in \mathcal{L}$. Models of higher orders are constructed similarly, counting trinucleotide and quadnucleotide (4-mer) occurrences for 2nd and 3rd order Markov models respectively. That is, for a Markov model of order m, the number of occurrences of each W-mer (where $W = m + 1$) are counted. The number of different W-mers is equal to 4^{m+1} and increases exponentially with m.

The size of the available data must be taken into account when choosing the order of the Markov background model (m). This is particularly important when using only the input sequences to construct the background model; as m increases, the background model will become increasingly sparse. In cosmo, Bembom, *et al* use likelihood-based cross-validation to choose m data-adaptively as one of the input parameters to the algorithm [21]. Liu, *et al.* also suggest choosing m data-adaptively to avoid this problem. A simple heuristic in BioProspector chooses m such that:

$$4^m \approx \frac{n}{1,024}, \tag{6.6}$$

where n is the number of nucleotides in the dataset used to construct the model [107]. This heuristic is used in the tests for evaluating Markov background models in this section. In practice, the size of the datasets used in this thesis restricts m to be 1 at most in the case of background models constructed using the input data and 3 at most in the case of models constructed using *E. coli* intergenic sequences; this fits well with the MEME Suite documentation, which recommends that Markov background models be limited to order 3 for motif discovery in DNA sequences. Although the restriction of m in the case of background models constructed using the full *E. coli* genome is less severe, it is also limited to 3 here so as to match the intergenic case.

Using background models to calculate probabilities

Substituting the traditional background model with a higher order Markov background model is relatively straightforward for both deterministic and stochastic EM. The product of probabilities (for instance, in calculating the conditional probability of sequence X_i given the latent variables, as in Equation 4.16, page 86) remains a product of probabilities, but drawing the relevant values from the Markov background model depending on the preceding nucleotide(s). Example 6.1 demonstrates how the background probability for a given subsequence may be calculated using a Markov background model. The start of an input sequence presents a special case as there is no preceding data available. Care must therefore be taken to ensure that such positions are handled correctly. It follows that Markov background models must be created for all orders up

Suppose the probability of interest is the background probability of the subsequence GAGAT in the input sequence CACGAGAT using a 3rd order Markov background model. In order to calculate this, the probabilities of each nucleotide are multiplied as with a 0th order model; however, probabilities for each nucleotide are taken from the 3rd order Markov model, so that:

$$p(GAGAT) = p(G|C,A,C)p(A|G,C,A)p(G|A,G,C)p(A|G,A,G)p(T|A,G,A) \quad (6.7)$$

Note that in this case, the notation $p(G|C,A,C)$ is shorthand for $p(X_{i,j} = G|X_{i,j-1} = C, X_{i,j-2} = A, X_{i,j-3} = C)$, rather than the traditional definition denoting conditional probability.

Example 6.1: Calculating probabilities using a Markov background model.

Suppose the probability of interest is the background probability of the subsequence CACGA in the input sequence CACGAGAT using a 3rd order Markov background model. In order to calculate this, the probabilities of each nucleotide are multiplied as before, but care must be taken in that the subsequence of interest is at the start of the sequence. Probabilities for each nucleotide are taken from the highest order Markov model possible, given the available data, so that:

$$p(CACGA) = p(C)p(A|C)p(C|A,C)p(G|C,A,C)p(A|G,C,A) \quad (6.8)$$

Example 6.2: Calculating probabilities using the highest order Markov model, given the available data.

to and including the chosen order m, in order to ensure the correct handling of positions near the start of an input sequence. Example 6.2 demonstrates how all orders of Markov model are employed in order to calculate background probabilities at the start of a sequence.

The higher order Markov background model is implemented as part of the MITSU algorithm developed in Chapter 5. As in the other algorithms implementing a Markov background model, the parameters of the background model are estimated at the start of the algorithm, then fixed. It is then assumed for the purposes of motif discovery that this is the true background model. As noted by Bembom, *et al.*, this is unlikely to exactly model the biological process by which the 'background' upstream sequences are created; however, the approximation of this process should be improved. Also, as noted previously, it is expected that the computational efficiency of the motif discovery

algorithm is improved, as the background model is not reestimated at each iteration.

It is noted that the background model is also involved in predicting motif occurrences after the discovery phase of the algorithm is complete. In the classification phase of the algorithm outlined in Section 5.3.3, the background model is used as part of the Bayes-optimal classifier used to determine whether or not a given subsequence is a motif occurrence (Equation 5.35, page 138). For simplicity, and following the method used in MEME, only the 0th order portion of the Markov background model is used in this calculation; however, it would be possible to implement the full Markov background model in this step.

6.3.3 Results and Discussion

The effect of using higher order Markov background models was assessed on both a deterministic EM-based algorithm and the stochastic EM-based algorithm (MITSU) developed in Chapter 5. As in the MCOIN evaluations presented in Section 4.4, performance was evaluated using previously characterised *E. coli* and diverse prokaryotic data. Details of these datasets are provided in Sections 3.2 and 3.3, respectively. Three different data sources were used to construct the background models: the input sequences alone, the full set of intergenic sequences (IGS) from the *E. coli* genome and the full *E. coli* genome sequence. As discussed above, 0th and 1st order background models were constructed using the input sequences and models of order 0, 1, 2 and 3 were constructed using the intergenic sequences and the full *E. coli* genome, following Liu, *et al.*'s heuristic (Equation 6.6).

Higher order models improve likelihood

To confirm that higher order Markov background models do indeed improve modelling of the intergenic sequences, background models of varying orders were constructed using both the intergenic sequence (IGS) data and the full *E. coli* genome. The likelihoods for each model were calculated using the intergenic sequence data (434,011nt). To account for increasing model complexity (as the background model order increases, the number of free parameters increases drastically), scores for the three essential information criteria (BIC, AIC and AICc) were computed. These scores are presented in Table 6.3.

It is apparent that increasing the order of the background model does improve modelling of intergenic sequences, even when accounting for the increased number

		IGS data			Full genome		
m	k	BIC	AIC	AICc	BIC	AIC	AICc
0	3	1,187,128	1,187,096	1,187,096	1,206,124	1,206,092	1,206,092
1	12	1,179,547	1,179,416	1,179,416	1,198,109	1,197,978	1,197,978
2	48	1,174,297	1,173,770	1,173,770	1,198,921	1,198,394	1,198,394
3	192	1,170,410	1,168,302	1,168,302	1,196,680	1,194,502	1,194,502

Table 6.3: Scores for the three essential information criteria computed using log likelihoods of varying orders of background model trained on intergenic sequence (IGS) data. Increasing the order of the background model (m) generally improves (decreases) the information criteria scores, even when accounting for the increasing numbers of free parameters (k).

of model parameters, based on the scores for the three essential information criteria. However, the improvement is relatively small (it is noted that the log likelihood for the 3rd order model trained on the IGS data corresponds with a probability of 0.2604 for each nucleotide, only slightly better than an equiprobable model).

As may be expected, the background models trained on intergenic sequence (IGS) data consistently outperform those trained using the full *E. coli* genome, across all information criteria. It is noted that the calculated log likelihood for the 0th order model trained on the IGS data ($-593,545$) is higher than that for a 0th order equiprobable model ($-601,667$). However, the calculated log likelihood for the 0th order model trained on the full *E. coli* genome ($-603,043$) is lower than that for a 0th order equiprobable model. This result illustrates the importance of carefully choosing the data used to construct the background model: even at the 0th order level, the differences in nucleotide distribution between the IGS data and the full *E. coli* genome (noted in Section 3.4) are great enough to lead to a noticeable difference in log likelihood.

To test the significance of the increase in log likelihood between the varying background model orders, a log likelihood ratio test was performed, testing the computed log likelihoods against the null hypothesis of equal performance; the asymptotic p-value was calculated to be $< 2.2 \times 10^{-16}$ in all cases.

Markov background model in deterministic EM

Subsequent tests used higher order Markov background models as part of a motif discovery algorithm. Tables 6.4-6.6 summarise the results of the tests carried out using a deterministic EM-based algorithm. Detailed results are provided in Tables B.7-B.9 in Appendix B. The results on the *E. coli* data collection (Table 6.4) go some way to confirming the conclusions reached by Thijs, *et al.* [166, 165]. Thijs, *et al.* noted an improvement in results when implementing a 3rd order Markov background model constructed from the set of intergenic sequences (IGS), compared to a 0th order model constructed from the same data. However, it is unclear along which dimension this improvement is observed. Table 6.4 demonstrates an improvement in terms of positive predictive value at both the nucleotide (*nPPV*) and site (*sPPV*) levels and also in terms of the nucleotide-level performance coefficient (*nPC*). A slight improvement is also noted in terms of overall motif correctness, based on AUC. Thijs, *et al.* also noted that a 3rd order model constructed using IGS data improved on a 3rd order model constructed using only the input data. While such a model is not tested here, it is noted that the 3rd order model constructed using IGS data also improves on both 0th and 1st order models constructed using only the input data in terms of *sPPV*, *nPPV*, *nPC* and AUC.

As noted above, Liu, *et al.* and Hartmann, *et al.* have suggested that higher order Markov background models have the effect of the reducing the number of false positive predictions [107, 78]. It is noted here that reducing the number of false positive predictions would result in an increase in positive predictive value. This result can be observed in Table 6.4. However, this increase in positive predictive value is coupled with a decrease in sensitivity. The reason for this is that fewer sites are being predicted overall, leading to fewer true positive motif occurrences being predicted.

The results presented in Table 6.4 also broadly agree with the conclusions of Hu, *et al.*, in that the order of the Markov background does not significantly affect results in terms of *nPC*. Although more variation is shown in the *nPC* results for models constructed using IGS data, the difference in *nPC* for the 0th and 3rd order models (0.19 vs. 0.23) is found not to be statistically significant at $p < 0.05$, based on a one-sided Wilcoxon signed rank test. A greater effect is observed based on the data used to construct the background model. In general, the models constructed using IGS data slightly outperformed those using only the input sequences and significantly outperformed those using the full *E. coli* genome. The increase in performance when using

IGS data agrees with Hu *et al.*'s tests on MotifSampler and confirms that a background model constructed using IGS data better models the background positions within the input sequences, compared to a background model constructed using the full *E. coli* genome.

Data source	*m*	*sSn*	*sPPV*	AUC	*nSn*	*nPPV*	*nPC*
Input data	0	0.76	0.33	0.97	0.53	0.25	0.21
	1	0.64	0.31	0.96	0.42	0.22	0.19
IGS data	0	0.77	0.30	0.97	0.53	0.22	0.19
	1	0.78	0.38	**0.98**	0.52	0.27	0.22
	2	0.68	0.32	0.96	0.45	0.23	0.20
	3	0.74	**0.39**	**0.98**	0.49	**0.28**	**0.23**
Full genome	0	0.80	0.18	0.94	**0.60**	0.13	0.12
	1	0.79	0.19	0.97	0.59	0.14	0.13
	2	0.76	0.18	0.96	0.57	0.14	0.12
	3	**0.83**	0.20	0.96	0.59	0.15	0.13

Table 6.4: Mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*), area under the ROC curve (AUC), mean nucleotide-level sensitivity (*nSn*) and mean nucleotide-level positive predictive value (*nPPV*) for a deterministic EM-based algorithm tested on 20 datasets created using previously characterised *E. coli* TFBS sequences. Best mean results are printed in bold. In these tests, motif discovery was only carried out at the experimentally determined motif width.

The results of the four *E. coli* ChIP datasets (Table 6.5) broadly confirm those of the *E. coli* data collection above. Again, the 3rd order Markov background model constructed using IGS data outperforms the 0th order constructed from the same data and both the 0th and 1st order models constructed using only the input sequences. However, the 3rd order model outperforms the other models in all performance measures, including site- and nucleotide-level sensitivity. In general, the models constructed using IGS data outperform those constructed using only the input sequences and those constructed using the full *E. coli* genome. There is more variation between models constructed using the same data in terms of *nPC* (for example, the *nPC* values for the IGS models vary between 0.35 and 0.59). However, this is likely due to the small sample size; again, the difference in *nPC* for the 0th and 3rd order IGS models is found not to be statistically significant at $p < 0.05$, based on a one-sided Wilcoxon signed rank

Data source	*m*	*sSn*	*sPPV*	AUC	*nSn*	*nPPV*	*nPC*
Input data	0	0.64	0.50	**0.99**	0.55	0.44	0.41
	1	0.51	0.51	0.98	0.42	0.42	0.33
IGS data	0	0.67	0.39	0.97	0.61	0.36	0.35
	1	0.70	0.68	0.98	0.66	0.64	0.57
	2	0.73	0.58	**0.99**	0.66	0.53	0.49
	3	**0.74**	**0.71**	**0.99**	**0.69**	**0.66**	**0.59**
Full genome	0	0.49	0.18	0.96	0.39	0.15	0.12
	1	0.65	0.23	0.98	0.52	0.18	0.16
	2	0.51	0.19	0.97	0.38	0.14	0.12
	3	0.39	0.15	0.95	0.28	0.11	0.09

Table 6.5: Mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*), area under the ROC curve (AUC), mean nucleotide-level sensitivity (*nSn*) and mean nucleotide-level positive predictive value (*nPPV*) for a deterministic EM-based algorithm tested on 4 datasets created using previously characterised *E. coli* TFBS sequences determined through ChIP experiments. Best mean results are printed in bold. In these tests, motif discovery was only carried out at the experimentally determined motif width.

Data source	*m*	*sSn*	*sPPV*	AUC	*nSn*	*nPPV*	*nPC*
Input data	0	**0.89**	**0.85**	**1.00**	**0.79**	**0.75**	**0.64**
	1	0.75	0.77	**1.00**	0.61	0.62	0.49

Table 6.6: Mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*), area under the ROC curve (AUC), mean nucleotide-level sensitivity (*nSn*) and mean nucleotide-level positive predictive value (*nPPV*) for a deterministic EM-based algorithm tested on 5 datasets created using previously characterised diverse prokaryotic TFBS sequences determined through ChIP experiments. Background models constructed using intergenic sequence and full genome data were not tested. Best mean results are printed in bold. In these tests, motif discovery was only carried out at the experimentally determined motif width.

test. As with the tests on the *E. coli* data collection (Table 6.4), background models constructed using the full *E. coli* genome perform relatively poorly, in terms of *nPC*.

Table 6.6 presents the results of the five non-*E. coli* datasets from the diverse prokaryotic data collection. In testing background models of orders 0 and 1 constructed using only the input sequences, the background model of order 0 is shown to outperform that of order 1 in all performance measures. However, the differences in performance are found not to be statistically significant for any performance measure at $p < 0.05$, based on one-sided Wilcoxon signed rank tests. Again, the variation in results is likely due to the small sample size. The 0th order background model constructed using the input data is also observed to outperform the 1st order model constructed using the same data in the other deterministic EM tests (Tables 6.4 and 6.5). These results suggest that if the only available data for background model construction is the input data, there is no significant advantage to increasing the order of the background model.

Markov background in MITSU (stochastic EM)

Tables 6.7-6.9 summarise the results of the tests carried out using the MITSU algorithm developed in Chapter 5. Detailed results are provided in Tables B.10-B.12 in Appendix B. In contrast to the deterministic EM results, the results of testing MITSU on the *E. coli* data collection (Table 6.7) show only a slight improvement in *sPPV* from the 0th order IGS background model to the 3rd order IGS background model. All other performance measures show a decrease in performance, in contrasts to the conclusions made by Thijs, *et al.* [166]. Similarly, the 3rd order IGS background model is not shown to improve on the 0th order model constructed using the input data (although it matches in terms of AUC).

The results in Table 6.7 agree with Liu, *et al.*'s conclusion that increasing the order of the Markov background model reduces the number of false positive predictions (hence increasing *sPPV*) [107]. However, while the 3rd order IGS model does indeed slightly increase *sPPV* over the 0th order IGS model, both the 1st and 2nd order IGS models show higher *sPPV* values. No increase in *sPPV* is noted for background models constructed using the input sequences or the full *E. coli* genome.

As with the corresponding deterministic EM results, it is observed that *nPC* is not significantly affected by the order of the Markov model, agreeing with the conclusions of Hu, *et al.* [81]. However, in contrast, there is no clear advantage (in terms of *nPC*) to using different data sources for constructing the background model.

Data source	*m*	*sSn*	*sPPV*	AUC	*nSn*	*nPPV*	*nPC*
Input data	0	**0.68**	**0.86**	**0.99**	**0.50**	**0.61**	**0.42**
	1	0.58	0.80	0.98	0.39	0.52	0.32
IGS data	0	0.67	0.79	0.98	0.48	0.56	0.39
	1	0.63	0.83	**0.99**	0.45	0.57	0.37
	2	0.61	0.84	0.98	0.43	0.57	0.35
	3	0.61	0.82	**0.99**	0.42	0.54	0.33
Full genome	0	0.64	0.77	**0.99**	0.46	0.53	0.37
	1	0.62	0.75	**0.99**	0.47	0.55	0.39
	2	0.64	0.77	**0.99**	0.48	0.56	0.39
	3	0.67	0.77	**0.99**	0.49	0.56	0.40

Table 6.7: Mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*), area under the ROC curve (AUC), mean nucleotide-level sensitivity (*nSn*) and mean nucleotide-level positive predictive value (*nPPV*) for a stochastic EM-based algorithm (MITSU) tested on 20 datasets created using previously characterised *E. coli* TFBS sequences. Best mean results are printed in bold. In these tests, motif discovery was only carried out at the experimentally determined motif width.

The results of the four *E. coli* ChIP datasets (Table 6.8) align well with the conclusions of Thijs, *et al.* [166]. The 3rd order IGS background model does improve on the 0th order IGS background model, based on all performance measures. However, the 3rd order IGS background model only shows a slight improvement over the 0th order background model constructed using the input data. Further, the 1st order background model constructed using the input data outperforms the 3rd order IGS model (and all other tested models) in terms of all performance measures.

An increase in *sPPV* is observed from the 0th order IGS model to the 3rd order IGS model. However, as with the results on the *E. coli* data collection (Table 6.7), there is not a consistent increase as the complexity of the background model is increased. Notably, *sPPV* decreases with increasing background model order for the models constructed using the full *E. coli* genome.

Again, *nPC* is shown to be not significantly affected by the order of the Markov model, agreeing with the conclusions of Hu, *et al.* [81]. Although *nPC* is increased from 0.71 to 0.79 between the 0th and 1st order models created using the input data, this difference is found not to be statistically significant at $p < 0.05$, based on a one-sided Wilcoxon signed rank test. As with the results on the *E. coli* data collection, there is no clear advantage to using different data sources, in terms of *nPC*.

Table 6.9 presents the results of the five non-*E. coli* datasets from the diverse prokaryotic data collection. As before, the 0th order model is shown to slightly outperform the 1st order model; however, the difference in results is not found to be statistically significant.

6.3.4 Conclusions

Higher order Markov background models have been used effectively in algorithms such as MEME [10], MotifSampler [166] and BioProspector [107]. In these algorithms, the use of a Markov background model allows the algorithm to better model the background positions within the input sequences and hence better discriminate between motif and background positions. Evaluations for single algorithms are generally unclear regarding which performance measures are improved by the use of Markov background models and to the extent of the performance increase. However, Hu *et al.*'s evaluation of several motif discovery algorithms suggests that using a higher order Markov background gives a noticeable increase in the nucleotide-level performance coefficient *nPC* [81]. Here, a rigorous study of higher order Markov background mod-

Data source	*m*	*sSn*	*sPPV*	AUC	*nSn*	*nPPV*	*nPC*
Input data	0	0.75	0.76	0.99	0.74	0.74	0.71
	1	**0.88**	**0.89**	**1.00**	**0.86**	**0.87**	**0.79**
IGS data	0	0.74	0.75	0.99	0.71	0.72	0.67
	1	0.75	0.75	0.99	0.73	0.73	0.69
	2	0.72	0.73	0.99	0.72	0.73	0.68
	3	0.80	0.81	0.99	0.76	0.77	0.73
Full genome	0	0.79	0.83	0.99	0.75	0.79	0.72
	1	0.77	0.79	0.99	0.73	0.76	0.69
	2	0.76	0.79	0.99	0.72	0.75	0.67
	3	0.75	0.76	0.99	0.73	0.74	0.68

Table 6.8: Mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*), area under the ROC curve (AUC), mean nucleotide-level sensitivity (*nSn*) and mean nucleotide-level positive predictive value (*nPPV*) for a stochastic EM-based algorithm (MITSU) tested on 4 datasets created using previously characterised *E. coli* TFBS sequences determined through ChIP experiments. Best mean results are printed in bold. In these tests, motif discovery was only carried out at the experimentally determined motif width.

Data source	*m*	*sSn*	*sPPV*	AUC	*nSn*	*nPPV*	*nPC*
Input data	0	**0.94**	**0.99**	**1.00**	**0.93**	**0.97**	**0.91**
	1	0.93	0.97	**1.00**	0.92	0.96	0.90

Table 6.9: Mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*), area under the ROC curve (AUC), mean nucleotide-level sensitivity (*nSn*) and mean nucleotide-level positive predictive value (*nPPV*) for a stochastic EM-based algorithm (MITSU) tested on 5 datasets created using previously characterised diverse prokaryotic TFBS sequences determined through ChIP experiments. Background models constructed using intergenic sequence and full genome data were not tested. Best mean results are printed in bold. In these tests, motif discovery was only carried out at the experimentally determined motif width.

els is undertaken in order to clarify which performance measures are improved by using these models, in the context of deterministic EM. Further, the effects of using higher order Markov models are evaluated for the first time in the context of a stochastic EM-based algorithm for motif discovery (namely the MITSU algorithm developed in Chapter 5).

It is noted that the tests carried out on a deterministic EM-based algorithm generally agree with the conclusions reached by Thijs, *et al.* [166], Liu, *et al.* [107] and Hu, *et al.* [81]. Firstly, a 3rd order Markov background model constructed using the intergenic sequences (IGS) is shown to outperform a 0th order model constructed using the same data. The same model is also shown to outperform a 0th order model constructed using the input data sequences alone. In both cases, performance is increased over a number of measures, most notably in terms of site- and nucleotide-level positive predictive value and *nPC*. Secondly, the 3rd order IGS background model is shown to reduce the number of false positive motif predictions, leading to an increase in site-level positive predictive value *sPPV* (however, it is noted that there is not a consistent increase in *sPPV* with increasing model order). Finally, it is noted that although the order of the Markov background model does not significantly affect *nPC*, the source of data used to construct the model is important. Generally, background models constructed from IGS data perform better in terms of *nPC* when compared to models constructed using only the input data sequences, or the full *E. coli* genome.

In contrast, tests carried out on a stochastic EM-based algorithm (MITSU) generally led to different conclusions. The 3rd order IGS background model did not consistently improve on the 0th order IGS model, but was consistently outperformed across all performance measures by lower order models constructed using only the input data. Similarly, although a slight increase in *sPPV* is noticeable between 0th and 3rd order IGS background models, this is not consistent as model order is increased, nor is the increase in *sPPV* noticeable when using models constructed using the full *E. coli* genome. Finally, while the order of the background model does not significantly affect *nPC*, there is no significant improvement in choosing different data sources for constructing the background model.

Based on the results of the above tests, it is not possible to state that implementing a higher order Markov background model improves motif discovery in the context of sEM. Although higher order background models provided increases in some performance measures for individual datasets, these increases were not part of an overall trend.

The most likely reason for the lack of improvement in the context of sEM is the fact that, as noted above, implementing a higher order Markov background increases the likelihood of the background sequences, with a corresponding increase in discriminative power. However, this increase is only very slight. This increase is shown to make a noticeable improvement in the case of deterministic EM. However, comparing Tables 6.4-6.6 to Tables 6.7-6.9, MITSU is shown to generally outperform deterministic EM across all performance measures, consistent with the results presented in Section 5.4. It is likely that the overall improvement in basic performance displayed by MITSU over deterministic EM outweighs the improvements that may be gained by using a higher order Markov background model.

The datasets in the *E. coli* data collection with *nPC* results in the bottom quartile when using the 0th order IGS background model all displayed no change, or a slight increase, in *nPC* when using the 3rd order IGS background model (the mean *nPC* for these datasets was increased from 0.13 to 0.17). This result suggests that in cases where the motif is discovered relatively poorly by sEM, using a higher order Markov background model may provide a slight increase in performance in terms of *nPC*.

As noted above, Thijs, *et al.* have suggested that higher order Markov background models may be more effective when searching for motifs in higher eukaryotic species [165]. While the motifs used in this study were prokaryotic, it is unlikely that this alone explains the lack of improvement gained by applying Markov background models to sEM, since a clear performance advantage is demonstrated in deterministic EM. However, it cannot be ruled out that the advantages of Markov background models for sEM would become more apparent when searching for motifs in higher eukaryotic species.

6.4 An information-theoretic measure of TFBS motif palindromicity

This section introduces a novel and flexible measure of motif palindromicity, based on information theory. The theoretical advantages of this measure over current constraint-based approaches are discussed.

6.4.1 Background and motivation

It has been noted (for example, in Section 2.1) that while position weight matrices (PWMs) assume independence between motif positions, this is often not the case for real motif examples. As discussed in Section 3.2, well-conserved positions in *E. coli* motifs are frequently flanked by other well-conserved positions, and vice versa (this phenomenon had previously been observed in eukaryotic TFBS motifs [56]). In addition to these small-scale dependencies, larger motif structures are also known. For instance, experimental work has shown that motifs often occur as palindromes (or 'inverted repeats'). That is, the inverse complement of the motif is the same as the original motif. Palindromic motifs often indicate that the transcription factor protein has a dimeric structure, binding to the DNA at two separate contact points[2]. Clearly, if a motif to be discovered is thought to be palindromic, it would be useful to incorporate this information into the algorithm in order to guide the motif search.

In their EM algorithm for motif discovery, Lawrence and Reilly [99] use a simple constraint on the parameters of the motif model to consider motifs which are (at least partially) palindromic. This is illustrated using the CRP dataset as described in Section 3.6.1: the motif model is constrained by requiring that the parameters in motif positions 4-8 are the same as the complementary parameters in positions 19-15. A similar approach is used by Bailey and Elkan [11], although this is generalised to all columns of the PWM, such that:

$$\theta = \begin{bmatrix} \theta_{1,A} & \theta_{2,A} & \cdots & \theta_{2,T} & \theta_{1,T} \\ \theta_{1,C} & \theta_{2,C} & \cdots & \theta_{2,G} & \theta_{1,G} \\ \theta_{1,G} & \theta_{2,G} & \cdots & \theta_{2,C} & \theta_{1,C} \\ \theta_{1,T} & \theta_{2,T} & \cdots & \theta_{2,A} & \theta_{1,A} \end{bmatrix}. \tag{6.9}$$

That is, the last column in the PWM is the inverse of the first column, the second last column is the inverse of the second, and so on. A heuristic based on the likelihood ratio test (LRT) is used to determine whether to use the palindrome constraint or not (the constraint is used in cases where it would improve the value of the LRT objective function) [11]. In the BioProspector algorithm, Liu, *et al.* [107] make some minor alterations to the method, modelling only one half of a palindromic motif; again, it is assumed that the second half of the motif is the inverse of the first half.

One disadvantage of Lawrence and Reilly's method is that the palindrome con-

[2]However, a dimeric transcription factor structure does not imply that the motif will be palindromic; the two contact points may have the same structure, this is known as a 'direct repeat'.

straints are required to be calculated separately for each tested motif. In their CRP example, a decision is made to enforce palindromicity between positions 4-8 and 19-15 [99]. However, these constraints would not work well in cases where the motif is of a different length, or cases where the motif is palindromic at different positions or at every position. Although Bailey and Elkan [11] extend the constraints to every column in the PWM, both approaches are relatively inflexible in the assumptions they make. The major assumption behind this approach is that the motif to be discovered is symmetrical in terms of the conservation at motif positions, which need not be the case. For example, the known motif in the CRP dataset (Figure 3.6, page 80) is not symmetrical: it can be observed that the conservation of the `T` nucleotide at position 4 is not the same as that of the `A` nucleotide in position 19. It is also assumed that the motif is perfectly palindromic throughout and, further, that the motif is palindromic around the centre, which again need not be the case. For example, the *C. crescentus* CtrA motif discussed in Section 3.5.2 (with consensus sequence `TTAA-N7-TTAAC`) contains the palindromic subsequence `TTAA-N7-TTAA`, but the addition of the `C` nucleotide at the 3' end means it is not palindromic around the centre.

Keles, *et al.* [90] note that, besides the methods used by Bailey and Elkan and Liu, *et al.* to enforce palindromicity (which are regarded as being 'ad hoc'), there has been little exploration of general methods to supervise the motif search, particularly regarding palindromic motifs. Keles, *et al.* [90] propose 'constrained mixture models' which could be used to allow some general constraints on PWMs, such as constraining the information content profile. It is noted that it is straightforward to similarly add constraints to the symmetry and palindromicity of a PWM, but this is not expanded. Bembom, *et al.* [21] later build upon the work of Keles, *et al.* [90]. The PWM in Bembom, *et al.*'s algorithm, cosmo, is allowed to be split into intervals which can be specified by the user, either in terms of particular positions, or fractions of the motif length. This latter option allows for greater flexibility in cases where the motif width and/or structure is unknown. Palindromicity between two PWM intervals may be enforced, by requiring the parameters of one interval are the reverse complement of the other interval. Unlike MEME and BioProspector, the parameter constraint in cosmo may be relaxed by specifying a tolerance which can allow small deviations between the palindromic positions.

While the general constraint system proposed by Keles, *et al.* [90] and extended by Bembom, *et al.* [21] has advantages over the parameter constraints used in MEME and BioProspector, this system still has some inflexibility, given that motifs may not

be exactly palindromic and are unlikely to be exactly symmetrical in terms of their conservation, even if their consensus sequence is exactly palindromic. This motivates a flexible measure of palindromicity which can score motif models (PWMs). This palindromicity score may be used as a form of model-level prior knowledge and incorporated using the sEM framework in MITSU (as part of the Metropolis mechanism that either accepts or rejects proposed models).

6.4.2 Methods

Measuring motif palindromicity

Given a DNA sequence (for example, `ACATATATG`), it is straightforward to find the reverse complement of the sequence; with a simple scoring method (for instance, 1 for a match, 0 otherwise) the palindromicity of a sequence can be measured (normalising for the length of the sequence). However, measuring the palindromicity of a PWM is more challenging, as each position is not a certain nucleotide (A, C, G or T), but a set of probabilities that each nucleotide will appear at that position. For example, a typical PWM corresponding to the consensus sequence above may look like this:

$$\theta = \begin{bmatrix} 0.9 & 0.1 & 0.8 & 0.0 & 1.0 & 0.0 & 0.9 & 0.2 & 0.1 \\ 0.0 & 0.7 & 0.0 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 \\ 0.0 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 & 0.7 \\ 0.1 & 0.1 & 0.2 & 0.9 & 0.0 & 1.0 & 0.0 & 0.8 & 0.1 \end{bmatrix}. \tag{6.10}$$

Using a similar method to that used in Section 4.4.2, a measure of motif palindromicity can be calculated, given a PWM, by computing the Jensen-Shannon distance (the square root of the Jensen-Shannon divergence, see Equation 4.56) for each column in the PWM to its corresponding column in the reverse complement PWM, summing and normalising by the number of columns in the PWM. Using such a measure, a score of 0 represents a perfect palindrome (that is, each column in the model is exactly the same as its corresponding column in the reverse complement model) and a score of 1 represents a completely non-palindromic sequence. Subtracting this score from 1 gives a measure where 1 represents a perfect palindrome and 0 a perfect non-palindrome. The palindromicity of a PWM θ is therefore defined as:

$$M_P(\theta) = 1 - \left(\frac{1}{W} \sum_{j=1}^{W} \sqrt{D_{\mathrm{JS}}(\theta_j || \theta_j^{(c)})} \right), \tag{6.11}$$

where j represents the columns in the PWM and $\theta^{(c)}$ represents the reverse complement of the PWM θ.

6.4.3 Conclusions

The main advantage, in terms of algorithm design, of the measure of motif palindromicity presented in this section is the improved flexibility over previous approaches which constrained PWM parameter values. As noted above, these parameter constraint-based approaches are too inflexible to accurately model palindromic motifs; known palindromic motifs are unlikely to be exactly symmetrical in terms of their conservation at corresponding positions.

The measure of motif palindromicity presented here also offers an advantage in terms of model comparison. Using this measure means that the number of parameters in the motif model will remain constant (assuming the width remains constant), making it easier to compare non-palindromic motif models with those assumed to be palindromic. This is in contrast to the parameter constraint-based approach adopted by MEME and BioProspector, where the number of free parameters is reduced when the palindromicity heuristic is used. In MEME, the changing number of model parameters has an effect on other heuristics used to compare models, under the assumption that simpler models (with fewer parameters) are better. This is not an issue with the measure of motif palindromicity presented here. As noted above, BioProspector models only one half of a palindromic motif, assuming that the second half of the motif is the inverse of the first half. This requires changing the expressions which update the model at each iteration of the algorithm. In contrast, it would be expected that implementing the palindrome measure presented here would only require a minor change to the update expressions of a motif discovery algorithm in order to bias the algorithm towards motif models with higher palindromicity.

Chapter 7

Conclusions and further work

This chapter briefly summarises the results of the project and provides conclusions with regard to the aims of the study (7.1). Some areas for future work which could further develop the work in this thesis are also suggested (7.2).

7.1 Conclusions and project contributions

The thesis began by presenting the motivation for the transcription factor binding site motif discovery problem. The primary hypothesis, that TFBS motif discovery using stochastic EM improves on deterministic EM-based approaches in terms of previously established metrics, was stated. A literature review of motif discovery algorithms surveyed a number of current approaches, focusing on probabilistic algorithms which take the promoter regions of coregulated genes as input. A comparison of existing algorithms was made, concluding that a strong case could be made for a motif discovery algorithm based on stochastic EM.

7.1.1 Algorithmic contributions

A novel heuristic for automatically determining the most likely width (MCOIN) was presented and evaluated. MCOIN is based on the concepts of motif containment and information content. A measure of motif similarity based on the Jensen-Shannon distance was defined and used to measure motif similarity; the definition of information content at a given motif position introduced by Schneider and Stephens [145] was extended to measure mean information content per position for a given motif.

A set of generalised expressions were derived, defining the ZOOPS sequence model

in the context of stochastic EM for the first time. These expressions were implemented in a novel sEM-based algorithm for motif discovery (MITSU). A previously described cutting heuristic was used in order to implement a model which is constraint free with regard to the distribution of motif occurrences. An entropy-based function previously used in stochastic EM for motif discovery was modified in order to work with the ZOOPS sequence model and the cutting heuristic. A previously described Bayes-optimal classifier for predicting motif occurrences was also extended for use with stochastic EM.

MITSU was evaluated quantitatively on realistic synthetic and previously characterised prokaryotic data; this evaluation confirmed the primary hypothesis, showing that MITSU generally outperformed deterministic EM, particularly in terms of site-level positive predictive value. Further tests demonstrated the ability of stochastic EM to escape insignificant local maxima of the likelihood function which can trap deterministic EM; it is this ability that allows MITSU to achieve increased performance over deterministic EM.

Two extensions of MITSU were presented. A probabilistic erasing function was implemented within MITSU, allowing for sequential discovery of multiple different motifs. A higher order Markov background model was also implemented; for the first time, the effects of incorporating a higher order Markov background model in the context of stochastic EM were evaluated.

7.1.2 Results of Alphaproteobacterial tests

Further validation of MITSU was carried out using the previously characterised Alphaproteobacterial CtrA and FnrL motifs (in *C. crescentus* CB15 and *R. sphaeroides* 2.4.1, respectively); MITSU was shown to correctly discover these motifs.

MITSU was then applied to data from selected Alphaproteobacterial species in order to predict a novel motif and consensus sequence for the previously uncharacterised NtrX binding site. These tests provided a number of novel results. Strong evidence was found to suggest that the FnrL transcription factor plays a role in the regulation of denitrification genes; the FnrL transcription factor was also confirmed to play a role in the regulation of genes coding for cytochrome oxidase. These results also suggest that the regulation of denitrification genes and genes coding for cytochrome oxidase is controlled by more than one transcription factor (at least FnrL and NtrX); these transcription factors may work cooperatively to regulate these genes. Two NtrX candi-

date sequences were identified; further experimental investigation (for example, using site-directed mutagenesis) may confirm the role of these sequences.

The widespread discovery of the FnrL motif within the NtrX datasets motivated the use of the probabilistic erasing function within MITSU. Further passes of MITSU on these datasets identified additional candidate sequences. Again, experimental testing may confirm the role of these sequences.

7.2 Possible future work

The work in this thesis offers a number of opportunities for possible future research. In this section, four possible areas for further research are presented and briefly discussed.

7.2.1 Alternative motif model representations

In line with the majority of probabilistic motif discovery algorithms, MITSU uses position weight matrices (PWMs) to represent motifs. As noted in Section 2.1, PWMs make the assumption that each position within the motif is independent. Further, it is assumed that the width of the motif is fixed. However, there are a number of known motifs with varying (or 'flexible') length. This arises as the relevant transcription factor may bind to the DNA with different structural configurations, leading to the protein interacting varying lengths of the DNA [117]. This change in structural configuration may occur as a result of a second protein binding to the transcription factor, changing its conformation.

Recent research has investigated methods of modelling variable-width motifs using an approach based on Hidden Markov Models (HMMs). One method proposed by Mathelier and Wasserman [118], is known as transcription factor flexible models (TFFMs). TFFMs can model positional dependencies within motifs and also allow motifs to have variable width. In tests on ChIP-seq data from the ENCODE project, TFFMs have been shown to provide improved discrimination of motif sequences from background sequences, in comparison to PWMs and dinucleotide-based methods [118].

In their study, Mathelier and Wasserman use the Baum-Welch algorithm [19] to train the TFFMs on the ChIP-seq data. However, since the Baum-Welch algorithm is a modification of the EM algorithm, it suffers similar drawbacks to the EM algorithm, including convergence to a local maximum. One line of future research would be to

investigate the extent to which TFFMs could be integrated as the motif model within sEM-based motif discovery algorithms (and MITSU in particular). This may involve defining a stochastic version of the Baum-Welch algorithm or modifying the updating procedure within MITSU to accommodate for HMMs. However, this is thought to be feasible and would allow MITSU to discover a much broader range of motifs.

7.2.2 Other sources of biological knowledge

The incorporation of relevant biological knowledge within MITSU was demonstrated in Chapter 6. However, it is possible that other sources of biological knowledge could be incorporated as a result of implementing the stochastic EM algorithm, either at the sequence or model level; here, two possible sources of additional information are briefly described.

Recent research has shown that the 3-dimensional shape of a DNA sequence plays an important role in how transcription factors and other proteins bind to the DNA [186]. Zhou, *et al.* demonstrate that variations in DNA structural features such as the minor groove width, roll, propeller twist and helix twist are correlated with DNA sequence function, using the Fis transcription factor binding site as an example [186]. The prediction of these structural features based on the nucleotide sequence is demonstrated and validated against the results of X-ray crystallography and NMR spectroscopy. The DNAshape web server[1] uses a Monte Carlo approach to predict the DNA structural features above, given a nucleotide sequence as input. It may therefore be possible to use the output of DNAshape as a form of prior information within MITSU, coupling DNA structural information with the sequence-based motif discovery in order to further improve results.

In eukaryotic cells, chromatin is a complex of DNA and proteins that forms chromosomes within the cell nucleus. Chromatin structure has been shown to be important in transcription factor binding [104]. Lim, *et al.* present an approach based on multiple logistic regression which combines DNA sequence information with chromatin modification data in order to model the binding of the human p53 transcription factor on a genome-wide scale [104]. The results of Lim, *et al.*'s study show that the inclusion of chromatin modification data improves the prediction of functionally important p53 binding sites. Again, it may be possible to incorporate a measure of chromatin modification within MITSU to improve the discovery of eukaryotic TFBS motifs.

[1] http://rohslab.cmb.usc.edu/DNAshape/

7.2.3 Alternative background models

With regard to the work presented in Section 6.3, there remain a number of open questions regarding alternative background models.

Firstly, the higher order Markov background model implemented in this thesis was shown to be effective in the context of deterministic EM, but despite improvements in specific cases (particularly in cases where the motif was discovered poorly when using a 0th order model) was shown to be of minimal benefit in general when used in a stochastic EM-based algorithm. Further testing on characterised prokaryotic and eukaryotic motifs may indicate whether the results in this thesis hold for motif discovery in general.

Secondly, alternative complex background models may be considered for implementation and evaluation in the context of sEM. For example, background models based on Markov switching models have been introduced by Down and Hubbard [53]. These models, known as 'mosaic background models' allow switching between several types of background model, all with their own particular nucleotide distribution (which may be 0th order, or higher order). The motivation for such models is that evolutionary constraints may act non-uniformly, even in background sequences [53]; it may therefore be helpful to learn a number of background models and switch between them as required (for example, it is reported that as well as a relatively neutral background model, learned models appear to include a GC-rich background model corresponding to CpG islands and purine- and pyrimidine-rich models). Mosaic background models have previously been disregarded due to their complexity (for instance, Jackson and Fitzgerald note that such models are outwith the scope of their study [86]); however, an evaluation of the effect of these background models in the context of sEM is one possible area for future study.

7.2.4 Extension of work on Alphaproteobacterial regulators

A number of results from the application of MITSU to the discovery of a motif for the uncharacterised Alphaproteobacterial NtrX regulator (described above) require experimental investigation or confirmation. Strong evidence for the role of FnrL in the regulation of denitrification genes was discovered; this may be relatively easy to confirm experimentally. Further testing on additional Alphaproteobacterial species may also strengthen the other candidate sequences discovered in this work.

The experiments on the NtrX regulator carried out in Section 6.1 also pave the way

for motif discovery in other uncharacterised regulators. The PrrA transcription factor binding site motif has been identified as a possible regulator for future study. As noted in Section 3.5.2, the photosynthetic response regulator (PrrBA) two-component system plays an important role in the expression of photosynthesis genes in *R. sphaeroides*. In this system, the PrrA transcription factor serves as a response regulator, binding to the DNA promoter regions. The PrrB transcription factor is a membrane-localised histidine kinase. The PrrBA two component system is essential in activating many of the photosynthesis genes in response to low oxygen tension. Besides regulation of these genes, PrrA has been shown to regulate a number of additional cellular functions such as carbon dioxide fixation, nitrogen fixation, hydrogen uptake and oxidation [58, 60]. The PrrBA two-component system has been shown to play a role as a redox sensor in the regulation of NtrYX in *B. abortus* [36]. Transcriptomic studies have shown that around 25% of the *R. sphaeroides* genome is controlled either directly or indirectly by PrrA, which can act as both an activator or repressor of gene expression [60]. The PrrBA system has a known homologue (known as RegBA) in *Rhodobacter capsulatus* [58] and *Brucella suis* [1]. Consensus sites have been difficult to determine [116]; however, a suggested (weak) consensus for the PrrA (RegA) binding sites in *R. sphaeroides* and *R. capsulatus* is `GYGSSRNNWWNNCRC`.

The difficulty in determining a consensus for PrrA binding sites stems from the fact that there is a great deal of flexibility in the binding of the transcription factor to the DNA. PrrA is thought to bind with two conserved blocks separated by a gap of variable length [116]. PrrA has been shown to bind to two different binding sites in the expression of *hemA* in *R. sphaeroides*; Ranson-Olson, *et al.* have suggested that it is likely that the phosphorylation state of PrrA plays a role in which site is bound by PrrA [139]. While current computational approaches to motif discovery would likely predict at least part of the PrrA motif, the variability in the length of the PrrA motif suggests that further study of methods for modelling motifs with flexible lengths as described in Section 7.2.1 is required.

Appendix A

Dataset listings

A.1 *Caulobacter crescentus* CB15 datasets

Locus ID	Gene product annotation
CC_0233	flagellin modification protein FlmA
CC_0378	modification methylase CcrMI
CC_0430	methyl-accepting chemotaxis protein McpA
CC_0953	flagellar basal-body rod protein FlgB, putative
CC_1035	hypothetical protein
CC_1457	flagellin modification protein FlmG
CC_1458	flbT protein
CC_2062	flagellar protein FliL
CC_2063	flagellar basal-body rod protein FlgF
CC_2552	cell division protein DivB
CC_2628	hfaA protein
CC_2868	neuB protein, putative
CC_2949	hypothetical protein
CC_3286	response regulator
CC_3599	RNA polymerase sigma-54 factor

Table A.1: 15 genes corresponding to cluster D of the genes experimentally determined to be regulated by CtrA in *Caulobacter crescentus* CB15, as reported by Laub, *et al.* [96].

Locus ID	Gene product annotation
CC_0232	hypothetical protein
CC_0350	pentapeptide repeat family protein
CC_0429	hypothetical protein
CC_0792	flagellin FljM
CC_0793	flagellin FljN
CC_1101	conserved hypothetical protein
CC_1307	conserved hypothetical protein
CC_1850	GGDEF family protein
CC_1963	ATP-dependent Clp protease, proteolytic subunit
CC_2324	sensor histidine kinase/response regulator
CC_2640	hypothetical protein
CC_2948	pilus subunit protein PilA
CC_3219	sensor histidine kinase/response regulator
CC_3295	hypothetical protein
CC_3317	hypothetical protein

Table A.2: 15 genes corresponding to cluster E of the genes experimentally determined to be regulated by CtrA in *Caulobacter crescentus* CB15, as reported by Laub, *et al.* [96].

A.2 *Rhodobacter sphaeroides* 2.4.1 datasets

Table A.3 lists the 63 genes in *R. sphaeroides* 2.4.1 which have been shown by experiment to be regulated by FnrL [54]. Following Dufour, *et al.*, the 300nt upstream sequences for each of these genes were extracted in order to create the dataset.

Locus ID	Name	Gene product annotation	Predicted
RSP_3044	*dorS*	sensor histidine kinase/response regulator	
RSP_0690	*rdxI*	putative heavy metal translocating P-type ATPase	Y
RSP_2984	*hemA*	glutamul-tRNA reductase	Y
RSP_1877	*coxI*	cytochrome C oxidase subunit I	Y
RSP_0697	*uspA*	putative universal stress protein, UspA	Y
RSP_1256		enoyl (acyl carrier protein) reductase	
RSP_0104	*nuoF*	NADH quinone oxidoreductase F subunit	
RSP_0110	*nuoL*	NADH dehydrogenase subunit l	
RSP_1826	*coxII*	cytochrome C oxidase subunit II	Y
RSP_2247	*fusA*	elongation factor G	
RSP_0102	*nuoCD*	NADH dehydrogenase subunit D	
RSP_0101	*nuoB*	NADH dehydrogenase subunit B	
RSP_0698	*fnrL*	transcriptional regulator, FnrL	Y
RSP_0106	*nuoH*	NADH dehydrogenase subunit H	
RSP_0112	*nuoN*	NADH dehydrogenase subunit N	
RSP_0105	*nuoG*	NADH dehydrogenase subunit G	
RSP_0100	*nuoA*	NADH dehydrogenase subunit A	
RSP_1257	*phbC*	poly(R) hydroxyalkanoic acid synthase class I	
RSP_0107	*nuoI*	NADH dehydrogenase subunit I	
RSP_1829	*coxIII*	cytochrome C oxidase subunit III	
RSP_1827	*coxX*	cytochrome C oxidase assembly factor	
RSP_1828	*coxXI*	cytochrome C oxidase assembly protein	
RSP_3341		transcriptional regulator BadM/Rrf2 family	
RSP_1254		acetate kinase	

(continues over)

Table A.3: 63 genes corresponding to the experimentally determined FnrL regulon in *Rhodobacter sphaeroides* 2.4.1, as determined by Dufour, *et al.* [54]. The predicted FNR regulated genes (Y) in the 'Predicted' column agree extremely well with the experimentally derived genes.

Locus ID	Name	Gene product annotation	Predicted
RSP_0317	*hemN*	oxygen-independent coproporphyrinogen III oxidase	
RSP_0699	*hemZ*	oxygen-independent coproporphyrinogen III oxidase	Y
RSP_0692	*rdxB*	iron-sulfur binding protein RdxA/RdxB/FixG family	Y
RSP_0693	*ccoP*	cytochrome C oxidase cbb3 type subunit III	Y
RSP_0696	*ccoN*	cytochrome C oxidase cbb3 type subunit I	Y
RSP_0695	*ccoO*	cytochrome C oxidase cbb3 type subunit II	Y
RSP_0468		3-octaprenyl-4-hydroxybenzoate-carboxy-lyase	
RSP_0467	*ubiD*	decarboxylase, UbiD family	
RSP_0691	*rdxH*	trans-membrane cation transporter, FixH family	Y
RSP_2507	*ompW*	putative outer membrane protein, OmpW	Y
RSP_2395	*ccpA2*	cytochrome C peroxidase	
RSP_0689	*rdxS*	cytochrome C oxidase maturation protein cbb3 type	Y
RSP_3642	*exsB*	Putative transcription factor, ExsB family	
RSP_1255		phosphate acetyltransferase	
RSP_0281	*bchE*	putative protoporphyrin monomethyl-ester oxidative cyclase	Y
RSP_0103	*nuoE*	NADH dehydrogenase subunit E	
RSP_0694	*ccoQ*	cytochrome C oxidase cbb3 type subunit IV	Y
RSP_0775		cytochrome C family protein	
RSP_0465		peptidase U32 family	
RSP_1818	*feoB*	ferrous iron transport protein B	
RSP_0466		putative lipid carrier protein	
RSP_0464		peptidase U32 family	
RSP_0277	*bchP*	geranylgeranyl reductase	
RSP_0279	*bchG*	bacteriochlorophyll/chlorophyll A synthase	
RSP_0278	*pucC*	putative light harvesting 1 (b870) complex assembly protein, PucC	
RSP_1876		hypothetical protein	
RSP_0166	*dksA*	putative DnaK suppressor protein	Y
RSP_0276		isopentenyl diphosphate delta isomerase	
RSP_0280	*bchJ*	bacteriochlorophyll synthase, BchJ	
RSP_0820		cytochrome b561	
RSP_0108	*nuoJ*	NADH ubiquinone/plastoquinone oxidoreductase	
RSP_0109	*nuoK*	NADH ubiquinone oxidoreductase	
RSP_1819	*feoA*	ferrous iron transport protein A	
RSP_1817	*feoC*	hypothetical protein	
RSP_2337	*ccpA1*	hypothetical protein	
RSP_2573		hypothetical protein	
RSP_3641		putative PfkB family carbohydrate kinase	
RSP_3643		hypothetical protein	
RSP_3640		hypothetical protein	

Table A.3 (continued)

Table A.4 lists 20 genes (a subset of the 63 genes listed in Table A.3) in *R. sphaeroides* which have been shown by experiment to be regulated by FnrL and part of the predicted core FNR regulon conserved across the 87 Alphaproteobacterial species studied by Dufour, *et al* [54]. As with the larger FnrL dataset, the 300nt upstream sequences for each of these genes were extracted in order to create the dataset.

Locus ID	Name	Gene product annotation	Predicted
RSP_0690	*rdxI*	putative heavy metal translocating P-type ATPase	Y
RSP_2984	*hemA*	glutamul-tRNA reductase	Y
RSP_1877	*coxI*	cytochrome C oxidase subunit I	Y
RSP_0697	*uspA*	putative universal stress protein, UspA	Y
RSP_1826	*coxII*	cytochrome C oxidase subunit II	Y
RSP_0698	*fnrL*	transcriptional regulator, FnrL	Y
RSP_0317	*hemN*	oxygen-independent coproporphyrinogen III oxidase	
RSP_0699	*hemZ*	oxygen-independent coproporphyrinogen III oxidase	Y
RSP_0692	*rdxB*	iron-sulfur binding protein RdxA/RdxB/FixG family	Y
RSP_0693	*ccoP*	cytochrome C oxidase cbb3 type subunit III	Y
RSP_0696	*ccoN*	cytochrome C oxidase cbb3 type subunit I	Y
RSP_0695	*ccoO*	cytochrome C oxidase cbb3 type subunit II	Y
RSP_0691	*rdxH*	trans-membrane cation transporter, FixH family	Y
RSP_2507	*ompW*	putative outer membrane protein, OmpW	Y
RSP_0689	*rdxS*	cytochrome C oxidase maturation protein cbb3 type	Y
RSP_0281	*bchE*	putative protoporphyrin monomethyl-ester oxidative cyclase	Y
RSP_0694	*ccoQ*	cytochrome C oxidase cbb3 type subunit IV	Y
RSP_0465		peptidase U32 family	
RSP_0466		putative lipid carrier protein	
RSP_0166	*dksA*	putative DnaK suppressor protein	Y

Table A.4: 20 genes corresponding to the experimentally determined FnrL regulon in *Rhodobacter sphaeroides* 2.4.1, as determined by Dufour, *et al.* [54]; these genes are part of the predicted core FNR regulon that is conserved across a large number of Alphaproteobacteria. The predicted FNR regulated genes (Y) in the 'Predicted' column agree extremely well with the experimentally derived genes. (Gene names taken from Dufour, *et al.* [54]).

A.3 NtrX datasets

Species	1st gene in operon	Locus ID	Gene product annotation
B. abortus str. S19	*narK*	BAbS19_II08320	Nitrite extrusion protein
B. melitensis str. M28	*narK*	BM28_B0295	nitrite transporter
M. nodulans str. ORS 2060	*narG*	Mnod_2128	nitrate reductase, alpha subunit
Caulobacter sp. K31	*narK*	Caul_3862	major facilitator superfamily MFS_1
A. cryptum str. JF-5	*narG*	Acry_1581	respiratory nitrate reductase alpha subunit apoprotein
A. multivorum str. AIU301	*narG*	ACMV_16270	respiratory nitrate reductase alpha subunit

Table A.5: Nar-n6 dataset listing. Gene details correspond to the first gene in the *nar* operon determined in Section 3.5.2.

Species	1st gene in operon	Locus ID	Gene product annotation
B. abortus str. S19	*nirK*	BAbS19_II08720	Copper-containing nitrite reductase precursor
B. melitensis str. M28	*nirK*	BM28_B0251	Copper-containing nitrite reductase precursor
R. palustris str. BisA53	*nirK*	RPE_4071	nitrite reductase, copper-containing
R. etli str. CFN 42	*nirK*	RHE_PF00525	hypothetical protein

Table A.6: Nir-n4 dataset listing. Gene details correspond to the first gene in the *nir* operon determined in Section 3.5.2.

Species	1st gene in operon	Locus ID	Gene product annotation
B. abortus str. S19	*?*	BAbS19_II08840	Cytochrome c heme-binding site
R. palustris str. BisA53	*?*	RPE_0622	Nitric-oxide reductase
R. etli str. CFN 42	*norC*	RHE_PF00516	nitrate reductase protein

Table A.7: Nor-n3 dataset listing. Gene details correspond to the first gene in the *nor* operon determined in Section 3.5.2. Unnamed genes are denoted '*?*'.

Species	1st gene in operon	Locus ID	Gene product annotation
B. abortus str. S19	*hp*	BAbS19_II08590	hypothetical protein
R. palustris str. BisA53	*nosR*	RPC_0427	FMN-binding
R. palustris str. BisB18	*nosR*	RPE_3094	FMN-binding domain protein

Table A.8: Nos-n3 dataset listing. Gene details correspond to the first gene in the *nos* operon determined in Section 3.5.2. Genes coding for hypothetical proteins are denoted '*hp*'.

Species	1st gene in operon	Locus ID	Gene product annotation
G. diazotrophicus str. PAI 5	*nifA*	GDI0429	Protein nifX
M. extorquens str. CM4	*nifA*	Mchl_1311	transcriptional regulator, NifA subfamily, Fis Family
M. nodulans str. ORS 2060	*nifA*	Mnod_4004	transcriptional regulator, NifA, Fis Family
R. etli str. CFN 42	*nifA*	RHE_PD00228	transcriptional regulator NifA protein
R. leguminosarum bv. *trifolii* str. WSM2304	*nifA*	Rleg2_5044	Fis family transcriptional regulator
R. palustris str. BisA53	*nifA*	RPE_4543	transcriptional regulator, NifA, Fis family
R. palustris str. BisB18	*nifA*	RPC_4475	transcriptional regulator, NifA, Fis family
Z. mobilis str. ZM4 ATCC 31821	*nifA*	ZMO1816	4Fe-4S ferredoxin iron-sulfur binding domain protein

Table A.9: Nif-n8 dataset listing. Gene details correspond to the first gene in the *nif* operon determined in Section 3.5.2.

Species	1st gene in operon	Locus ID	Gene product annotation
B. abortus str. S19	*?*	BAbS19_II06810	Transcriptional regulator, ARSR family
R. palustris str. BisB18	*caiC*	RPC_1016	benzoate-CoA ligase
M. extorquens str. CM4	*cydD*	Mchl_1559	ABC transporter related
R. palustris str. BisA53	*acs*	RPE_0595	ABC transporter related
G. oxydans str. 621H	*cydD*	GOX2409	Transport ATP-binding protein CydD
G. diazotrophicus str. PAI 5	*rpoE*	GDI3522	putative phage integrase
R. leguminosarum bv. *trifolii* str. WSM2304	*?*	Rleg2_6537	putative transcriptional regulator protein
R. etli str. CFN 42	*cydC*	RHE_PF00035	probable ribose ABC transporter, ATP-binding protein
C. crescentus str. CB15	*cydD*	CC_0761	ABC transporter, ATP-binding protein CydD
Caulobacter sp. K31	*cydD*	Caul_0633	ABC transporter, CydDC cysteine exporter (CydDC-E) family, permease/ATP-binding protein CydD
A. cryptum str. JF-5	*cydD*	Acry_0554	ABC transporter, transmembrane region, type 1
A. cryptum str. JF-5	*cydD*	Acry_1637	ABC transporter, transmembrane region, type 1

Table A.10: Cyd-n12 dataset listing. Gene details correspond to the first gene in the *cyd* operon determined in Section 3.5.2. Unnamed genes are denoted '*?*'.

Species	1st gene in operon	Locus ID	Gene product annotation
B. abortus str. S19	*ccoN*	BAbS19_I03630	Cytochrome c oxidase cbb3-type
R. palustris str. BisB18	*ccoN*	RPC_0015	cytochrome c oxidase, cbb3-type, subunit I
M. nodulans str. ORS 2060	*ccoN*	Mnod_2111	cytochrome c oxidase, cbb3-type, subunit I
M. nodulans str. ORS 2060	*ccoN*	Mnod_5230	cytochrome c oxidase, cbb3-type, subunit I
R. palustris str. BisA53	*ccoN*	RPE_0018	cytochrome c oxidase, cbb3-type, subunit I
R. leguminosarum bv. *trifolii* str. WSM2304	*ccoN*	Rleg2_5015	cbb3-type cytochrome c oxidase subunit I
R. etli str. CFN 42	*fixN*	RHE_PF00507	nitric-oxide reductase protein
R. etli str. CFN 42	*ccoN*	RHE_PD00296	cytochrome C oxidase, fixN chain protein
C. crescentus str. CB15	*ccoN*	CC_1401	cytochrome c oxidase, CcoN subunit
Caulobacter sp. K31	*ccoN*	Caul_2437	cytochrome c oxidase, cbb3-type, subunit I

Table A.11: Cco-n10 dataset listing. Gene details correspond to the first gene in the *cco* operon determined in Section 3.5.2.

Appendix B

Additional results

Conservation	Known width (w^*)			MCOIN ($w^*\pm4$)			E-values ($w^*\pm4$)		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
2.00	0.84 ± 0.37	0.25 ± 0.43	0.99 ± 0.04	$\mathbf{0.93}^{\dagger\ddagger} \pm 0.26$	$0.42^{\dagger} \pm 0.49$	$\mathbf{1.00}^{\dagger\ddagger} \pm 0.03$	$0.91^{\dagger} \pm 0.28$	$\mathbf{0.79}^{\dagger *} \pm 0.27$	$0.99^{\dagger} \pm 0.03$
1.49	$0.26^{\ddagger} \pm 0.44$	0.07 ± 0.26	0.98 ± 0.05	$\mathbf{0.28}^{\dagger\ddagger} \pm 0.45$	$0.15^{\dagger} \pm 0.36$	$\mathbf{0.99}^{\dagger\ddagger} \pm 0.04$	0.21 ± 0.41	$\mathbf{0.45}^{\dagger *} \pm 0.26$	$0.98^{\dagger} \pm 0.04$
1.08	$\mathbf{0.02}^{*\ddagger} \pm 0.13$	0.01 ± 0.18	$\mathbf{0.96} \pm 0.05$	0.01 ± 0.12	$0.01^{\dagger} \pm 0.11$	$\mathbf{0.96}^{\dagger\ddagger} \pm 0.05$	$0.01^{*} \pm 0.11$	$\mathbf{0.23}^{\dagger *} \pm 0.15$	$\mathbf{0.96} \pm 0.04$
0.76	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.94}^{*} \pm 0.04$	0.00 ± 0.00	0.00 ± 0.00	0.93 ± 0.05	0.00 ± 0.00	$\mathbf{0.12}^{\dagger} \pm 0.09$	$\mathbf{0.94}^{*} \pm 0.04$
0.51	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.93}^{*} \pm 0.03$	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.93} \pm 0.04$	0.00 ± 0.00	$\mathbf{0.09}^{\dagger} \pm 0.06$	$\mathbf{0.93}^{*} \pm 0.03$

Table B.1: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for five collections of realistic synthetic data at varying levels of motif conservation. Best results are printed in bold. In these tests, the motif discovery algorithm was allowed to run as it would normally. $\pm$ values represent standard deviation. Results marked † are statistically significant with regard to the known width, results marked * are statistically significant with regard to the MCOIN heuristic and results marked ‡ are statistically significant with regard to the E-values estimator, all at $p \leq 0.05$ (see main text).

Conservation	Known width (w^*)			MCOIN ($w^*\pm4$)			E-values ($w^*\pm4$)		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
'high' (1.36)	**0.81*** $\pm$ 0.26	0.22‡ $\pm$ 0.09	**0.96** $\pm$ 0.04	0.72 $\pm$ 0.32	**0.29**‡ $\pm$ 0.22	**0.96** $\pm$ 0.06	0.70 $\pm$ 0.37	0.17 $\pm$ 0.11	0.95 $\pm$ 0.06
'low' (0.78)	0.63 $\pm$ 0.25	0.41 $\pm$ 0.24	0.96 $\pm$ 0.04	**0.69** $\pm$ 0.17	**0.51**‡ $\pm$ 0.28	**0.98** $\pm$ 0.02	0.66 $\pm$ 0.20	0.32 $\pm$ 0.20	0.97 $\pm$ 0.03
overall (1.13)	**0.74** $\pm$ 0.27	0.30‡ $\pm$ 0.19	**0.96** $\pm$ 0.04	0.71 $\pm$ 0.27	**0.38**‡ $\pm$ 0.27	**0.96**‡ $\pm$ 0.05	0.68 $\pm$ 0.31	0.23 $\pm$ 0.17	**0.96** $\pm$ 0.05

Table B.2: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for 20 datasets created using real *E. coli* data. Best mean results are printed in bold. $\pm$ values represent standard deviation. Results marked * are significant with regard to the MCOIN heuristic and results marked ‡ are significant with regard to the E-values estimator (see main text).

Conservation	Known width (w^*)			MCOIN ($w^*\pm4$)			E-values ($w^*\pm4$)		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
0.99	**0.75** $\pm$ 0.30	0.67 $\pm$ 0.31	**0.99** $\pm$ 0.02	**0.75** $\pm$ 0.28	**0.68** $\pm$ 0.30	**0.99** $\pm$ 0.01	0.73 $\pm$ 0.34	0.67 $\pm$ 0.31	**0.99** $\pm$ 0.02

Table B.3: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for 9 datasets created using real prokaryotic data determined through ChIP experiments. Best results are printed in bold. $\pm$ values represent standard deviation. The datasets used are summarised in Table 3.2.

Conservation	Deterministic EM			SEAM			MITSU		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
2.00	$0.84^{\ddagger} \pm 0.37$	0.25 ± 0.43	$\mathbf{0.99}^{\ddagger} \pm 0.04$	$\mathbf{1.00}^{\dagger\ddagger} \pm 0.05$	$\mathbf{1.00}^{\dagger\ddagger} \pm 0.05$	-	0.70 ± 0.43	$0.74^{\dagger} \pm 0.43$	0.97 ± 0.06
1.49	0.26 ± 0.44	0.07 ± 0.26	0.98 ± 0.05	$\mathbf{0.93}^{\dagger\ddagger} \pm 0.18$	$0.93^{\dagger} \pm 0.18$	-	$0.90^{\dagger} \pm 0.11$	$\mathbf{0.97}^{\dagger *} \pm 0.10$	$\mathbf{1.00}^{\dagger} \pm 0.02$
1.08	0.02 ± 0.13	0.01 ± 0.18	0.96 ± 0.05	$0.49^{\dagger} \pm 0.34$	$0.49^{\dagger} \pm 0.34$	-	$\mathbf{0.68}^{\dagger *} \pm 0.23$	$\mathbf{0.78}^{\dagger *} \pm 0.26$	$\mathbf{0.99}^{\dagger} \pm 0.03$
0.76	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.94} \pm 0.04$	$0.09^{\dagger} \pm 0.12$	$0.09^{\dagger} \pm 0.12$	-	$\mathbf{0.17}^{\dagger *} \pm 0.21$	$\mathbf{0.20}^{\dagger *} \pm 0.24$	$\mathbf{0.94}^{\dagger} \pm 0.04$
0.51	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.93}^{\ddagger} \pm 0.03$	$0.06^{\dagger} \pm 0.06$	$0.06^{\dagger} \pm 0.06$	-	$\mathbf{0.07}^{\dagger *} \pm 0.07$	$\mathbf{0.08}^{\dagger *} \pm 0.08$	$\mathbf{0.93} \pm 0.03$

Table B.4: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for five collections of realistic synthetic data with varying levels of motif conservation. Best mean results are printed in bold. In these tests, motif discovery was carried out only at the known motif width. $\pm$ values represent standard deviation. Results marked † are statistically significant with regard to deterministic EM, results marked * are statistically significant with regard to SEAM and results marked ‡ are statistically significant with regard to MITSU, all at $p \leq 0.05$ (see main text).

Conservation	Deterministic EM			SEAM			MITSU		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
'high' (1.36)	**0.81**‡ $\pm$ 0.26	0.22 $\pm$ 0.09	0.96 $\pm$ 0.04	0.67‡ $\pm$ 0.35	0.67† $\pm$ 0.35	-	0.54 $\pm$ 0.30	**0.75**† $\pm$ 0.31	**0.98** $\pm$ 0.02
'low' (0.78)	0.63 $\pm$ 0.25	0.41 $\pm$ 0.24	0.96 $\pm$ 0.04	**0.65** $\pm$ 0.30	0.65 $\pm$ 0.30	-	0.57 $\pm$ 0.23	**0.71**† $\pm$ 0.25	**0.97** $\pm$ 0.04
overall (1.13)	**0.74**‡ $\pm$ 0.27	0.30 $\pm$ 0.19	0.96 $\pm$ 0.04	0.66‡ $\pm$ 0.34	0.66† $\pm$ 0.34	-	0.55 $\pm$ 0.27	**0.73**† $\pm$ 0.29	**0.98**† $\pm$ 0.03

Table B.5: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for 20 datasets created using previously characterised *E. coli* TFBS sequences. Best mean results are printed in bold. In these tests, motif discovery was carried out only at the experimentally determined motif width. $\pm$ values represent standard deviation. Results marked † are statistically significant with regard to deterministic EM and results marked ‡ are statistically significant with regard to MITSU, both at $p \leq 0.05$ (see main text).

Conservation	Deterministic EM			SEAM			MITSU		
(mean bits/col)	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC	*sSn*	*sPPV*	AUC
0.99	0.75 ± 0.30	0.67 ± 0.31	0.99 ± 0.02	0.86 ± 0.19	$0.86^{\dagger} \pm 0.19$	-	$\mathbf{0.88}^{\dagger} \pm 0.14$	$\mathbf{0.92}^{\dagger *} \pm 0.13$	$\mathbf{1.00} \pm 0.00$

Table B.6: Mean site-level sensitivity (*sSn*), positive predictive value (*sPPV*) and area under the ROC curve (AUC) for 9 datasets created using real prokaryotic data determined through ChIP experiments. Best mean results are printed in bold. In these tests, motif discovery was carried out only at the experimentally determined motif width. $\pm$ values represent standard deviation. Results marked † are statistically significant with regard to deterministic EM and results marked * are statistically significant with regard to SEAM, both at $p \leq 0.05$ (see main text).

Data source	m	*sSn*	*sPPV*	AUC	*nSn*	*nPPV*	*nPC*
Input data	0	0.76 ± 0.27	0.33 ± 0.25	0.97 ± 0.04	0.53 ± 0.26	0.25 ± 0.24	0.21 ± 0.21
	1	0.64 ± 0.31	0.31 ± 0.23	0.96 ± 0.04	0.42 ± 0.25	0.22 ± 0.22	0.19 ± 0.19
IGS data	0	0.77 ± 0.28	0.30 ± 0.22	0.97 ± 0.04	0.53 ± 0.25	0.22 ± 0.20	0.19 ± 0.17
	1	0.78 ± 0.21	0.38 ± 0.25	$\mathbf{0.98} \pm 0.02$	0.52 ± 0.20	0.27 ± 0.23	0.22 ± 0.19
	2	0.68 ± 0.32	0.32 ± 0.24	0.96 ± 0.05	0.45 ± 0.26	0.23 ± 0.22	0.20 ± 0.19
	3	0.74 ± 0.23	$\mathbf{0.39} \pm 0.24$	$\mathbf{0.98} \pm 0.02$	0.49 ± 0.20	$\mathbf{0.28} \pm 0.23$	$\mathbf{0.23} \pm 0.20$
Full genome	0	0.80 ± 0.26	0.18 ± 0.08	0.94 ± 0.08	$\mathbf{0.60} \pm 0.25$	0.13 ± 0.06	0.12 ± 0.06
	1	0.79 ± 0.26	0.19 ± 0.08	0.97 ± 0.03	0.59 ± 0.23	0.14 ± 0.06	0.13 ± 0.06
	2	0.76 ± 0.28	0.18 ± 0.08	0.96 ± 0.03	0.57 ± 0.23	0.14 ± 0.06	0.12 ± 0.06
	3	$\mathbf{0.83} \pm 0.23$	0.20 ± 0.08	0.96 ± 0.03	0.59 ± 0.22	0.15 ± 0.07	0.13 ± 0.06

Table B.7: Mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*), area under the ROC curve (AUC), mean nucleotide-level sensitivity (*nSn*) and mean nucleotide-level positive predictive value (*nPPV*) for a deterministic EM-based algorithm tested on 20 datasets created using previously characterised *E. coli* TFBS sequences. Best mean results are printed in bold. In these tests, motif discovery was only carried out at the experimentally determined motif width. $\pm$ values represent standard deviation.

Data source	m	*sSn*	*sPPV*	AUC	*nSn*	*nPPV*	*nPC*
Input data	0	0.64 ± 0.36	0.50 ± 0.36	$\mathbf{0.99} \pm 0.02$	0.55 ± 0.38	0.44 ± 0.36	0.41 ± 0.35
	1	0.51 ± 0.37	0.51 ± 0.37	0.98 ± 0.03	0.42 ± 0.34	0.42 ± 0.34	0.33 ± 0.28
IGS data	0	0.67 ± 0.40	0.39 ± 0.31	0.97 ± 0.04	0.61 ± 0.37	0.36 ± 0.31	0.35 ± 0.32
	1	0.70 ± 0.40	0.68 ± 0.37	0.98 ± 0.03	0.66 ± 0.39	0.64 ± 0.36	0.57 ± 0.35
	2	0.73 ± 0.39	0.58 ± 0.32	$\mathbf{0.99} \pm 0.02$	0.66 ± 0.37	0.53 ± 0.32	0.49 ± 0.31
	3	$\mathbf{0.74} \pm 0.34$	$\mathbf{0.71} \pm 0.33$	$\mathbf{0.99} \pm 0.02$	$\mathbf{0.69} \pm 0.34$	$\mathbf{0.66} \pm 0.32$	$\mathbf{0.59} \pm 0.32$
Full genome	0	0.49 ± 0.22	0.18 ± 0.07	0.96 ± 0.02	0.39 ± 0.17	0.15 ± 0.07	0.12 ± 0.05
	1	0.65 ± 0.09	0.23 ± 0.02	0.98 ± 0.01	0.52 ± 0.11	0.18 ± 0.03	0.16 ± 0.03
	2	0.51 ± 0.24	0.19 ± 0.09	0.97 ± 0.02	0.38 ± 0.18	0.14 ± 0.06	0.12 ± 0.06
	3	0.39 ± 0.24	0.15 ± 0.09	0.95 ± 0.03	0.28 ± 0.18	0.11 ± 0.07	0.09 ± 0.06

Table B.8: Mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*), area under the ROC curve (AUC), mean nucleotide-level sensitivity (*nSn*) and mean nucleotide-level positive predictive value (*nPPV*) for a deterministic EM-based algorithm tested on 4 datasets created using previously characterised *E. coli* TFBS sequences determined through ChIP experiments. Best mean results are printed in bold. In these tests, motif discovery was only carried out at the experimentally determined motif width. $\pm$ values represent standard deviation.

Data source	m	*sSn*	*sPPV*	AUC	*nSn*	*nPPV*	*nPC*
Input data	0	**0.89** $\pm$ 0.10	**0.85** $\pm$ 0.09	**1.00** $\pm$ 0.00	**0.79** $\pm$ 0.13	**0.75** $\pm$ 0.13	**0.64** $\pm$ 0.16
	1	0.75 $\pm$ 0.26	0.77 $\pm$ 0.16	**1.00** $\pm$ 0.01	0.61 $\pm$ 0.28	0.62 $\pm$ 0.21	0.49 $\pm$ 0.26

Table B.9: Mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*), area under the ROC curve (AUC), mean nucleotide-level sensitivity (*nSn*) and mean nucleotide-level positive predictive value (*nPPV*) for a deterministic EM-based algorithm tested on 5 datasets created using previously characterised diverse prokaryotic TFBS sequences determined through ChIP experiments. Background models constructed using intergenic sequence and full genome data were not tested. Best mean results are printed in bold. In these tests, motif discovery was only carried out at the experimentally determined motif width. $\pm$ values represent standard deviation.

Data source	m	*sSn*	*sPPV*	AUC	*nSn*	*nPPV*	*nPC*
Input data	0	**0.68** $\pm$ 0.26	**0.86** $\pm$ 0.20	**0.99** $\pm$ 0.03	**0.50** $\pm$ 0.28	**0.61** $\pm$ 0.24	**0.42** $\pm$ 0.29
	1	0.58 $\pm$ 0.31	0.80 $\pm$ 0.28	0.98 $\pm$ 0.02	0.39 $\pm$ 0.30	0.52 $\pm$ 0.26	0.32 $\pm$ 0.28
IGS data	0	0.67 $\pm$ 0.27	0.79 $\pm$ 0.24	0.98 $\pm$ 0.03	0.48 $\pm$ 0.28	0.56 $\pm$ 0.26	0.39 $\pm$ 0.29
	1	0.63 $\pm$ 0.29	0.83 $\pm$ 0.22	**0.99** $\pm$ 0.02	0.45 $\pm$ 0.29	0.57 $\pm$ 0.24	0.37 $\pm$ 0.29
	2	0.61 $\pm$ 0.25	0.84 $\pm$ 0.21	0.98 $\pm$ 0.03	0.43 $\pm$ 0.26	0.57 $\pm$ 0.23	0.35 $\pm$ 0.26
	3	0.61 $\pm$ 0.28	0.82 $\pm$ 0.24	**0.99** $\pm$ 0.02	0.42 $\pm$ 0.26	0.54 $\pm$ 0.22	0.33 $\pm$ 0.24
Full genome	0	0.64 $\pm$ 0.32	0.77 $\pm$ 0.31	**0.99** $\pm$ 0.03	0.46 $\pm$ 0.29	0.53 $\pm$ 0.28	0.37 $\pm$ 0.28
	1	0.62 $\pm$ 0.34	0.75 $\pm$ 0.32	**0.99** $\pm$ 0.02	0.47 $\pm$ 0.32	0.55 $\pm$ 0.29	0.39 $\pm$ 0.31
	2	0.64 $\pm$ 0.32	0.77 $\pm$ 0.29	**0.99** $\pm$ 0.01	0.48 $\pm$ 0.30	0.56 $\pm$ 0.26	0.39 $\pm$ 0.29
	3	0.67 $\pm$ 0.33	0.77 $\pm$ 0.29	**0.99** $\pm$ 0.02	0.49 $\pm$ 0.28	0.56 $\pm$ 0.26	0.40 $\pm$ 0.28

Table B.10: Mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*), area under the ROC curve (AUC), mean nucleotide-level sensitivity (*nSn*) and mean nucleotide-level positive predictive value (*nPPV*) for a stochastic EM-based algorithm (MITSU) tested on 20 datasets created using previously characterised *E. coli* TFBS sequences. Best mean results are printed in bold. In these tests, motif discovery was only carried out at the experimentally determined motif width. $\pm$ values represent standard deviation.

Data source	m	*sSn*	*sPPV*	AUC	*nSn*	*nPPV*	*nPC*
Input data	0	0.75 ± 0.37	0.76 ± 0.36	0.99 ± 0.02	0.74 ± 0.39	0.74 ± 0.39	0.71 ± 0.40
	1	$\mathbf{0.88} \pm 0.13$	$\mathbf{0.89} \pm 0.11$	$\mathbf{1.00} \pm 0.00$	$\mathbf{0.86} \pm 0.14$	$\mathbf{0.87} \pm 0.13$	$\mathbf{0.79} \pm 0.21$
IGS data	0	0.74 ± 0.34	0.75 ± 0.34	0.99 ± 0.01	0.71 ± 0.37	0.72 ± 0.37	0.67 ± 0.39
	1	0.75 ± 0.37	0.75 ± 0.37	0.99 ± 0.01	0.73 ± 0.37	0.73 ± 0.37	0.69 ± 0.39
	2	0.72 ± 0.37	0.73 ± 0.37	0.99 ± 0.01	0.72 ± 0.38	0.73 ± 0.38	0.68 ± 0.39
	3	0.80 ± 0.29	0.81 ± 0.27	0.99 ± 0.01	0.76 ± 0.35	0.77 ± 0.34	0.73 ± 0.37
Full genome	0	0.79 ± 0.25	0.83 ± 0.25	0.99 ± 0.01	0.75 ± 0.32	0.79 ± 0.33	0.72 ± 0.36
	1	0.77 ± 0.29	0.79 ± 0.29	0.99 ± 0.01	0.73 ± 0.33	0.76 ± 0.33	0.69 ± 0.36
	2	0.76 ± 0.29	0.79 ± 0.29	0.99 ± 0.01	0.72 ± 0.33	0.75 ± 0.33	0.67 ± 0.37
	3	0.75 ± 0.33	0.76 ± 0.32	0.99 ± 0.01	0.73 ± 0.35	0.74 ± 0.35	0.68 ± 0.38

Table B.11: Mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*), area under the ROC curve (AUC), mean nucleotide-level sensitivity (*nSn*) and mean nucleotide-level positive predictive value (*nPPV*) for a stochastic EM-based algorithm (MITSU) tested on 4 datasets created using previously characterised *E. coli* TFBS sequences determined through ChIP experiments. Best mean results are printed in bold. In these tests, motif discovery was only carried out at the experimentally determined motif width. $\pm$ values represent standard deviation.

Data source	*m*	*sSn*	*sPPV*	AUC	*nSn*	*nPPV*	*nPC*
Input data	0	**0.94** $\pm$ 0.05	**0.99** $\pm$ 0.02	**1.00** $\pm$ 0.00	**0.93** $\pm$ 0.06	**0.97** $\pm$ 0.04	**0.91** $\pm$ 0.08
	1	0.93 $\pm$ 0.06	0.97 $\pm$ 0.05	**1.00** $\pm$ 0.00	0.92 $\pm$ 0.08	0.96 $\pm$ 0.07	0.90 $\pm$ 0.12

Table B.12: Mean site-level sensitivity (*sSn*), mean site-level positive predictive value (*sPPV*), area under the ROC curve (AUC), mean nucleotide-level sensitivity (*nSn*) and mean nucleotide-level positive predictive value (*nPPV*) for a stochastic EM-based algorithm (MITSU) tested on 5 datasets created using previously characterised diverse prokaryotic TFBS sequences determined through ChIP experiments. Background models constructed using intergenic sequence and full genome data were not tested. Best mean results are printed in bold. In these tests, motif discovery was only carried out at the experimentally determined motif width. $\pm$ values represent standard deviation.

Bibliography

[1] E. Abdou, A. Deredjian, M. P. Jiménez de Bagüés, S. Köhler, and V. Jubier-Maurin. RegA, the regulator of the two-component system RegB/RegA of *Brucella suis*, is a controller of both oxidative respiration and denitrification required for chronic infection in mice. *Infection and Immunity*, 81(6):2053–2061, 2013.

[2] M. Aitkin and D. B. Rubin. Estimation and hypothesis testing in finite mixture models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(1):67–75, 1985.

[3] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

[4] M. Amin, S. L. Porter, and O. S. Soyer. Split histidine kinases enable ultrasensitivity and bistability in two-component signaling networks. *PLoS Computational Biology*, 9(3):e1002949, 2013.

[5] W. Ao, J. Gaudet, W. J. Kent, S. Muttumu, and S. E. Mango. Environmentally induced foregut remodeling by PHA-4/FoxA and DAF-12/NHR. *Science*, 305(5691):1743–6, 2004.

[6] T. L. Bailey. *Discovering motifs in DNA and protein sequences: The approximate common substring problem*. PhD thesis, University of California, San Diego, 1995.

[7] T. L. Bailey. DREME: Motif discovery in transcription factor ChIP-seq data. *Bioinformatics*, 27(12):1653–1659, 2011.

[8] T. L. Bailey, M. Bodén, F. A. Buske, M. Frith, C. E. Grant, L. Clementi, J. Ren, W. W. Li, and W. S. Noble. MEME Suite: tools for motif discovery and searching. *Nucleic Acids Research*, 37(suppl 2):W202–W208, 2009.

[9] T. L. Bailey, M. Bodén, T. Whitington, and P. Machanick. The value of position-specific priors in motif discovery using MEME. *BMC Bioinformatics*, 11(1):179, 2010.

[10] T. L. Bailey and C. Elkan. Fitting a mixture model by Expectation Maximization to discover motifs in biopolymers. In *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology*, pages 28–36, 1994.

[11] T. L. Bailey and C. Elkan. The value of prior knowledge in discovering motifs with MEME. In *Proceedings of the 3rd International Conference on Intelligent Systems for Molecular Biology*, pages 21–29, 1995.

[12] T. L. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using Expectation Maximization. *Machine Learning*, 21:51–80, 1995.

[13] T. L. Bailey and M. Gribskov. Combining evidence using *p*-values: application to sequence homology searches. *Bioinformatics*, 14(1):48–54, 1998.

[14] T. L. Bailey and P. Machanick. Inferring direct DNA binding from ChIP-seq. *Nucleic Acids Research*, 40(17):e128, 2012.

[15] M. E. Baker, W. N. Grundy, and C. P. Elkan. A common ancestor for a subunit in the mitochondrial proton-translocating NADH:ubiquinone oxidoreductase (complex I) and short-chain dehydrogenases/reductases. *Cellular and Molecular Life Sciences*, 55:450–455, 1999.

[16] D. Banerjee and F. Slack. Temporal and spatial patterning of an organ by a single transcription factor. *Genome Biology*, 6(2):205, 2005.

[17] Y. Barash, G. Bejerano, and N. Friedman. A simple hyper-geometric approach for discovering putative transcription factor binding sites. In *Algorithms in Bioinformatics*, pages 278–293. Springer, 2001.

[18] Y. Barash, G. Elidan, N. Friedman, and T. Kaplan. Modeling dependencies in protein-DNA binding sites. In *Proceedings of the 7th Annual International Conference on Research in Computational Molecular Biology*, pages 28–37, 2003.

[19] L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:1–8, 1972.

[20] M. J. Beal and Z. Ghahramani. The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 7*. Oxford University Press, 2003.

[21] O. Bembom, S. Keles, and M. van der Laan. Supervised detection of conserved motifs in DNA sequences with cosmo. *Statistical Applications in Genetics and Molecular Biology*, 6(1):Article 8, 2007.

[22] E. Berezikov, V. Guryev, R. H. Plasterk, and E. Cuppen. CONREAL: conserved regulatory elements anchored alignment algorithm for identification of transcription factor binding sites by phylogenetic footprinting. *Genome Research*, 14(1):170–178, 2004.

[23] C. Bi. SEAM: a stochastic EM-type algorithm for motif-finding in biopolymer sequences. *Journal of Bioinformatics and Computational Biology*, 5(1):47–77, 2007.

[24] C. Bi. A Monte Carlo EM algorithm for de novo motif discovery in biomolecular sequences. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6:370–386, 2009.

[25] C. Bi. Comparison of optimization techniques for sequence pattern discovery by maximum-likelihood. *Pattern Recognition Letters*, 31(14):2147–2160, 2010.

[26] C. Bi, J. S. Leeder, and C. A. Vyhlidal. A comparative study on computational two-block motif detection: Algorithms and applications. *Molecular Pharmaceutics*, 5(1):3–16, 2008.

[27] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[28] J. G. Booth and J. P. Hobert. Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 61(1):265–285, 1999.

[29] J. G. Booth, J. P. Hobert, and W. Jank. A survey of Monte Carlo algorithms for maximizing the likelihood of a two-stage hierarchical model. *Statistical Modelling*, 1(4):333–349, 2001.

[30] B. Boussau, E. O. Karlberg, A. C. Frank, B.-A. Legault, and S. G. E. Andersson. Computational inference of scenarios for α-proteobacterial genome evolution. *Proceedings of the National Academy of Sciences of the United States of America*, 101(26):9722–9727, 2004.

[31] D. R. Buelow and T. L. Raivio. Three (and more) component regulatory systems auxiliary regulators of bacterial histidine kinases. *Molecular Microbiology*, 75(3):547–566, 2010.

[32] M. L. Bulyk, P. L. F. Johnson, and G. M. Church. Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucleic Acids Research*, 30(5):1255–1261, 2002.

[33] B. S. Caffo, W. Jank, and G. L. Jones. Ascent-based Monte Carlo Expectation Maximization. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 67:235–251, 2005.

[34] C. S. Carmack, L. McCue, L. Newberg, and C. Lawrence. PhyloScan: identification of transcription factor binding sites using cross-species evidence. *Algorithms for Molecular Biology*, 2(1):1, 2007.

[35] M. d. C. Carrica, I. Fernandez, M. A. Martí, G. Paris, and F. A. Goldbaum. The NtrY/X two-component system of *Brucella* spp. acts as a redox sensor and regulates the expression of nitrogen respiration enzymes. *Molecular Microbiology*, 85(1):39–50, 2012.

[36] M. d. C. Carrica, I. Fernandez, R. Sieira, G. Paris, and F. A. Goldbaum. The two-component systems PrrBA and NtrYX co-ordinately regulate the adaptation of *Brucella abortus* to an oxygen-limited environment. *Molecular Microbiology*, 88(2):222–233, 2013.

[37] G. Celeux, D. Chauveau, and J. Diebolt. On stochastic versions of the EM algorithm. *Rapport de Recherche-Institut National de Recherche en Informatique et en Automatique*, (2514), 1995.

[38] G. Celeux and J. Diebolt. A stochastic approximation type EM algorithm for the mixture problem. *Stochastics and Stochastic Reports*, 41(1-2):119–134, 1992.

[39] B.-K. Cho, S. A. Federowicz, M. Embree, Y.-S. Park, D. Kim, and B. Ø. Palsson. The PurR regulon in *Escherichia coli* K-12 MG1655. *Nucleic Acids Research*, 39:6456–6464, 2011.

[40] S. A. Dahouk, S. Loisel-Meyer, H. C. Scholz, H. Tomaso, M. Kersten, A. Harder, H. Neubauer, S. Köhler, and V. Jubier-Maurin. Proteomic analysis of *Brucella suis* under oxygen deficiency reveals flexibility in adaptive expression of various pathways. *PROTEOMICS*, 9(11):3011–3021, 2009.

[41] M. K. Das and H.-K. Dai. A survey of DNA motif finding algorithms. *BMC Bioinformatics*, 8 Suppl 7:S21, Jan. 2007.

[42] B. W. Davies, R. W. Bogard, and J. J. Mekalanos. Mapping the regulon of *Vibrio cholerae* ferric uptake regulator expands its known network of gene regulation. *Proceedings of the National Academy of Sciences of the United States of America*, 108:12467–12472, 2011.

[43] N. G. de Jong. Discovering the BMP signaling pathway by using motif enrichment and conservation. Technical report, Delft University of Technology, 2007.

[44] A. L. Delcher, D. Harmon, S. Kasif, O. White, and S. L. Salzberg. Improved microbial gene identification with GLIMMER. *Nucleic Acids Research*, 27(23):4636–4641, 1999.

[45] V. G. DelVecchio, V. Kapatral, R. J. Redkar, G. Patra, C. Mujer, T. Los, N. Ivanova, I. Anderson, A. Bhattacharyya, A. Lykidis, G. Reznik, L. Jablonski, N. Larsen, M. D'Souza, A. Bernal, M. Mazur, E. Goltsman, E. Selkov, P. H. Elzer, S. Hagius, D. O'Callaghan, J.-J. Letesson, R. Haselkorn, N. Kyrpides, and R. Overbeek. The genome sequence of the facultative intracellular pathogen *Brucella melitensis*. *Proceedings of the National Academy of Sciences of the United States of America*, 99(1):443–448, 2002.

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

[47] T. Z. DeSantis, I. Dubosarskiy, S. R. Murray, and G. L. Andersen. Comprehensive aligned sequence construction for automated design of effective probes (CASCADE-P) using 16S rDNA. *Bioinformatics*, 19(12):1461–1468, 2003.

[48] P. D'haeseleer. What are DNA sequence motifs? *Nature Biotechnology*, 24(4):423–425, 2006.

[49] J. Diebolt and E. H. S. Ip. A stochastic EM algorithm for approximating the maximum likelihood estimate. Technical Report 301, Department of Statistics, Stanford University, 1994.

[50] J. Diebolt and C. Robert. Bayesian estimation of finite mixture distributions: part II, Sampling implementation. Technical Report 111, Laboratoire de Statistique Théorique et Appliquée, Université Paris VI, 1990.

[51] J. Diebolt and C. Robert. Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(2):363–75, 1994.

[52] T. G. Dong and J. J. Mekalanos. Characterization of the RpoN regulon reveals differential regulation of T6SS and new flagellar operons in *Vibrio cholerae* O37 strain V52. *Nucleic Acids Research*, 40(16):7766–7775, 2012.

[53] T. A. Down and T. J. P. Hubbard. NestedMICA: sensitive inference of over-represented motifs in nucleic acid sequence. *Nucleic Acids Research*, 33(5):1445–53, 2005.

[54] Y. S. Dufour, P. J. Kiley, and T. J. Donohue. Reconstruction of the core and extended regulons of global transcription factors. *PLoS Genetics*, 6(7):e1001027, 2010.

[55] S. R. Eddy. Multiple alignment using hidden Markov models. In *Proceedings of the 3rd International Conference on Intelligent Systems for Molecular Biology*, pages 114–20, 1995.

[56] M. Eisen. All motifs are NOT created equal: structural properties of transcription factor-DNA interactions and the inference of sequence specificity. *Genome Biology*, 6:P7, 2005.

[57] W. H. Elliot and D. C. Elliot. *Biochemistry and Molecular Biology*. Oxford University Press, second edition, 2001.

[58] S. Elsen, L. R. Swem, D. L. Swem, and C. E. Bauer. RegB/RegA, a highly conserved redox-responding global two-component regulatory system. *Microbiology and Molecular Biology Reviews*, 68(2):263–279, 2004.

[59] D. Endres and J. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858 – 1860, 2003.

[60] J. M. Eraso, J. H. Roh, X. Zeng, S. J. Callister, M. S. Lipton, and S. Kaplan. Role of the global transcriptional regulator PrrA in *Rhodobacter sphaeroides* 2.4.1: Combined transcriptome and proteome analysis. *Journal of Bacteriology*, 190(14):4831–4848, 2008.

[61] M. P. Ferla, J. C. Thrash, S. J. Giovannoni, and W. M. Patrick. New rRNA gene-based phylogenies of the Alphaproteobacteria provide perspective on major groups, mitochondrial ancestry and phylogenetic instability. *PLoS ONE*, 8(12):e83383, 12 2013.

[62] A. O. Ferreira, C. R. Myers, J. S. Gordon, G. B. Martin, M. Vencato, A. Collmer, M. D. Wehling, J. R. Alfano, G. Moreno-Hagelsieb, W. F. Lamboy, G. DeClerck, D. J. Schneider, and S. W. Cartinhour. Whole-genome expression profiling defines the HrpL regulon of *Pseudomonas syringae* pv. *tomato* DC3000, allows de novo reconstruction of the Hrp *cis* element, and identifies novel coregulated genes. *Molecular Plant-Microbe Interactions*, 19(11):1167–1179, 2006.

[63] M. C. Frith, N. F. W. Saunders, B. Kobe, and T. L. Bailey. Discovering sequence motifs with arbitrary insertions and deletions. *PLoS Computational Biology*, 4(5):e1000071, 2008.

[64] A. Galgano, M. Forrer, L. Jaskiewicz, A. Kanitz, M. Zavolan, and A. Gerber. Comparative analysis of mRNA targets for human PUF-family proteins suggests extensive interaction with the miRNA regulatory system. *PLoS ONE*, 3(9):e3164, 2008.

[65] S. Gama-Castro, H. Salgado, M. Peralta-Gil, A. Santos-Zavaleta, L. Muñiz Rascado, H. Solano-Lira, V. Jimenez-Jacinto, V. Weiss, J. S. García-Sotelo, A. López-Fuentes, L. Porrón-Sotelo, S. Alquicira-Hernández, A. Medina-Rivera, I. Martínez-Flores, K. Alquicira-Hernández, R. Martínez-Adame, C. Bonavides-Martínez, J. Miranda-Ríos, A. M. Huerta, A. Mendoza-Vargas, L. Collado-Torres, B. Taboada, L. Vega-Alvarado, M. Olvera, L. Olvera, R. Grande, E. Morett, and J. Collado-Vides. RegulonDB version 7.0: transcriptional regulation of *Escherichia coli* K-12 integrated within genetic sensory response units (Gensor Units). *Nucleic Acids Research*, 39(Suppl 1):D98–D105, 2011.

[66] D. Gamerman and H. Lopes. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman & Hall, 2006.

[67] R. Gao and A. M. Stock. Biological insights from structures of two-component proteins. *Annual Review of Microbiology*, 63(1):133–154, 2009.

[68] J. Gaudet, S. Muttumu, M. Horner, and S. E. Mango. Whole-genome analysis of temporal gene expression during foregut development. *PLoS Biology*, 2(11):e352, 2004.

[69] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–511, 1992.

[70] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721 –741, 1984.

[71] D. C. Grainger, H. Aiba, D. Hurd, D. F. Browning, and S. J. W. Busby. Transcription factor distribution in *Escherichia coli*: studies with FNR protein. *Nucleic Acids Research*, 35(1):269–278, 2007.

[72] D. C. Grainger, D. Hurd, M. Harrison, J. Holdstock, and S. J. W. Busby. Studies of the distribution of *Escherichia coli* cAMP-receptor protein and RNA polymerase along the *E. coli* chromosome. *Proceedings of the National Academy of Sciences of the United States of America*, 102(49):17693–17698, 2005.

[73] J. Gregor, T. Zeller, A. Balzer, K. Haberzettl, and G. Klug. Bacterial regulatory networks include direct contact of response regulator proteins: interaction of RegA and NtrX in *Rhodobacter capsulatus*. *Journal of Molecular Microbiology and Biotechnology*, 13(1-3):126–139, 2007.

[74] M. G. Gu and H.-T. Zhu. Maximum likelihood estimation for spatial models by Markov chain Monte Carlo stochastic approximation. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 63:339–355, 2001.

[75] D. GuhaThakurta. Computational identification of transcriptional regulatory elements in DNA sequence. *Nucleic Acids Research*, 34(12):3585–3598, 2006.

[76] D. GuhaThakurta and G. D. Stormo. Identifying target sites for cooperatively binding factors. *Bioinformatics*, 17(7):608–621, 2001.

[77] R. Gupta and A. Mok. Phylogenomics and signature proteins for the alpha Proteobacteria and its main groups. *BMC Microbiology*, 7:106, 2007.

[78] H. Hartmann, E. W. Guthöhrlein, M. Siebert, S. Luehr, and J. Söding. *P*-value based regulatory motif discovery using positional weight matrices. *Genome Research*, 23(1):181–194, 2013.

[79] L. Heikkinen, S. Asikainen, and G. Wong. Identification of phylogenetically conserved sequence motifs in microRNA 5' flanking sites from *C. elegans* and *C. briggsae*. *BMC Molecular Biology*, 9(1):105, 2008.

[80] G. Hertz and G. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7-8):563–577, 1999.

[81] J. Hu, B. Li, and D. Kihara. Limitations and potentials of current motif discovery algorithms. *Nucleic Acids Research*, 33(15):4899–4913, 2005.

[82] 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*. *Journal of Molecular Biology*, 296(5):1205–1214, 2000.

[83] R. Hughey and A. Krogh. Hidden Markov models for sequence analysis: extension and analysis of the basic method. *Computer Applications in the Biosciences: CABIOS*, 12(2):95–107, 1996.

[84] D. Husmeier. *Neural networks for conditional probability estimation*. Springer, 1999.

[85] E. H. S. Ip. A stochastic EM estimator in the presence of missing data - theory and applications. Technical Report 304, Department of Statistics, Stanford University, 1994.

[86] E. S. Jackson and W. J. Fitzgerald. A sequential Monte Carlo EM approach to the transcription factor binding site identification problem. *Bioinformatics*, 23(11):1313–1320, 2007.

[87] W. Jank. Stochastic variants of EM: Monte Carlo, Quasi-Monte Carlo and more. In *Proceedings of the American Statistical Association*, 2005.

[88] S. T. Jensen, X. S. Liu, Q. Zhou, and J. S. Liu. Computational discovery of gene regulatory binding motifs: a Bayesian perspective. *Statistical Science*, 19(1):188–204, 2004.

[89] I. Jonassen, J. Collins, and D. Higgins. Finding flexible patterns in unaligned protein sequences. *Protein Science*, 4(8):1587–95, 1995.

[90] 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):Article 5, 2003.

[91] A. M. Kilpatrick, B. Ward, and S. Aitken. MCOIN: a novel heuristic for determining transcription factor binding site motif width. *Algorithms for Molecular Biology*, 8(16), 2013.

[92] A. M. Kilpatrick, B. Ward, and S. Aitken. Stochastic EM-based TFBS motif discovery with MITSU. *Bioinformatics*, 30(12):i310–i318, 2014.

[93] A. Krogh. Two methods for improving performance of a HMM and their application for gene finding. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*, pages 179–186, 1997.

[94] I. V. Kulakovskiy, V. A. Boeva, A. V. Favorov, and V. J. Makeev. Deep and wide digging for binding motifs in ChIP-Seq data. *Bioinformatics*, 26(20):2622–2623, 2010.

[95] I. V. Kulakovskiy, V. Levitsky, D. Oshchepkov, L. Bryzgalov, I. Vorontsov, and V. J. Makeev. From binding motifs in ChIP-Seq data to improved models of transcription factor binding sites. *Journal of Bioinformatics and Computational Biology*, 11(1):1340004, 2013.

[96] M. T. Laub, S. L. Chen, L. Shapiro, and H. H. McAdams. Genes directly controlled by CtrA, a master regulator of the *Caulobacter* cell cycle. *Proceedings of the National Academy of Sciences of the United States of America*, 99(7):4632–4637, 2002.

[97] M. T. Laub, H. H. McAdams, T. Feldblyum, C. M. Fraser, and L. Shapiro. Global analysis of the genetic network controlling a bacterial cell cycle. *Science*, 290(5499):2144–2148, 2000.

[98] 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(5131):208–14, 1993.

[99] C. E. Lawrence and A. A. Reilly. An Expectation Maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, 7(1):41–51, 1990.

[100] S. Le Crom, F. Devaux, P. Marc, X. Zhang, W. S. Moye-Rowley, and C. Jacq. New insights into the pleiotropic drug resistance network from genome-wide characterization of the YRR1 transcription factor regulation system. *Molecular and Cellular Biology*, 22(8):2642–2649, 2002.

[101] H. C. M. Leung and F. Y. L. Chin. Finding motifs from all sequences with and without binding sites. *Bioinformatics*, 22(18):2217–2223, 2006.

[102] G. Li, B. Liu, and Y. Xu. Accurate recognition of *cis*-regulatory motifs with the correct lengths in prokaryotic genomes. *Nucleic Acids Research*, 38(2):e12, 2010.

[103] L. Li, R. L. Bass, and Y. Liang. fdrMotif: identifying *cis*-elements by an EM algorithm coupled with false discovery rate control. *Bioinformatics*, 24(5):629–636, 2008.

[104] J.-H. Lim, R. D. Iggo, and D. Barker. Models incorporating chromatin modification data identify functionally important p53 binding sites. *Nucleic Acids Research*, 41(11):5582–5593, 2013.

[105] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37:145–151, 1991.

[106] K. Lin and D. Husmeier. Modelling transcriptional regulation with a mixture of factor analyzers and variational Bayesian Expectation Maximization. *EURASIP Journal on Bioinformatics and Systems Biology*, 2009:601068, 2009.

[107] X. Liu, D. Brutlag, and J. Liu. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. In *Proceedings of the 6th Pacific Symposium on Biocomputing*, pages 127–138, 2001.

[108] H. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, and J. E. Darnell. *Molecular Cell Biology*. W. H. Freeman and Company, 2001.

[109] S. Loisel-Meyer, M. P. J. de Bagüés, S. Köhler, J.-P. Liautard, and V. Jubier-Maurin. Differential use of the two high-oxygen-affinity terminal oxidases of *Brucella suis* for in vitro and intramacrophagic multiplication. *Infection and Immunity*, 73(11):7768–7771, 2005.

[110] A. V. Lukashin and M. Borodovsky. GeneMark.hmm: New solutions for gene finding. *Nucleic Acids Research*, 26(4):1107–1115, 1998.

[111] D. Lun, A. Sherrid, B. Weiner, D. Sherman, and J. Galagan. A blind deconvolution approach to high-resolution mapping of transcription factor binding sites from ChIP-seq data. *Genome Biology*, 10:R142, 2009.

[112] X. Ma, A. Kulkarni, Z. Zhang, Z. Xuan, R. Serfling, and M. Q. Zhang. A highly efficient and effective motif discovery method for ChIP-seq/ChIP-chip data using positional information. *Nucleic Acids Research*, 40(7):e50, 2012.

[113] P. Machanick and T. L. Bailey. MEME-ChIP: motif analysis of large DNA datasets. *Bioinformatics*, 27(12):1696–1697, 2011.

[114] K. D. MacIsaac and E. Fraenkel. Practical strategies for discovering regulatory DNA sequence motifs. *PLoS Computational Biology*, 2(4):e36, 2006.

[115] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[116] L. Mao, C. Mackenzie, J. H. Roh, J. M. Eraso, S. Kaplan, and H. Resat. Combining microarray and genomic data to predict DNA binding motifs. *Microbiology*, 151(10):3197–3213, 2005.

[117] A. Mathelier and W. W. Wasserman. Predicting Transcription Factor Binding Sites with Hidden Markov Models using ChIP-Seq Data. Poster session presented at 9th Annual Rocky Mountain Bioinformatics Conference; 8-10 December 2011; Snowmass, Colorado, USA, 2011.

[118] A. Mathelier and W. W. Wasserman. The next generation of transcription factor binding site prediction. *PLoS Computational Biology*, 9(9):e1003214, 2013.

[119] A. M. Mehdi, M. S. B. Sehgal, B. Kobe, T. L. Bailey, and M. Bodén. DLocalMotif: A discriminative approach for discovering local motifs in protein sequences. *Bioinformatics*, 29(1):39–46, 2012.

[120] X.-L. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

[121] L. A. Mirny and M. S. Gelfand. Structural analysis of conserved base pairs in protein-DNA complexes. *Nucleic Acids Research*, 30(7):1704–1711, 2002.

[122] C. D. Mohr, J. K. MacKichan, and L. Shapiro. A membrane-associated protein, FliX, is required for an early step in *Caulobacter* flagellar assembly. *Journal of Bacteriology*, 180(8):2175–2185, 1998.

[123] V. Molle, M. Fujita, S. T. Jensen, P. Eichenberger, J. E. González-Pastor, J. S. Liu, and R. Losick. The Spo0A regulon of *Bacillus subtilis*. *Molecular Microbiology*, 50(5):1683–1701, 2003.

[124] F. Mordelet, J. Horton, A. J. Hartemink, B. E. Engelhardt, and R. Gordân. Stability selection for regression-based models of transcription factor-DNA binding specificity. *Bioinformatics*, 29(13):i117–i125, 2013.

[125] A. Moses, D. Chiang, and M. Eisen. Phylogenetic motif detection by Expectation-Maximization on evolutionary mixtures. In *Proceedings of the 9th Pacific Symposium on Biocomputing*, pages 324–335, 2004.

[126] N. J. Mouncey and S. Kaplan. Oxygen regulation of the *ccoN* gene encoding a component of the cbb_3 oxidase in *Rhodobacter sphaeroides* $2.4.1^{T}$: involvement of the FnrL protein. *Journal of Bacteriology*, 180(8):2228–2231, 1998.

[127] S. Mukherjee, M. F. Berger, G. Jona, X. S. Wang, D. Muzzey, M. Snyder, R. A. Young, and M. L. Bulyk. Rapid analysis of the DNA-binding specificities of transcription factors with DNA microarrays. *Nature Genetics*, 36(12):1331–1339, 2004.

[128] L. Narlikar, R. Gordân, U. Ohler, and A. J. Hartemink. Informative priors based on transcription factor structural class improve *de novo* motif discovery. *Bioinformatics*, 22(14):e384–e392, 2006.

[129] R. M. Neal. Slice sampling. *Annals of Statistics*, 31:705–767, 2003.

[130] R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*. MIT Press, 1999.

[131] G. Pavesi, G. Mauri, and G. Pesole. *In silico* representation and discovery of transcription factor binding sites. *Briefings in Bioinformatics*, 5(3):217–36, 2004.

[132] M. Petersen, P. Brodersen, H. Naested, E. Andreasson, U. Lindhart, B. Johansen, H. B. Nielsen, M. Lacy, M. J. Austin, J. E. Parker, S. B. Sharma, D. F. Klessig, R. Martienssen, O. Mattsson, A. B. Jensen, and J. Mundy. *Arabidopsis* MAP kinase 4 negatively regulates systemic acquired resistance. *Cell*, 103(7):1111–1120, 2000.

[133] P. A. Pevzner and S.-H. Sze. Combinatorial approaches to finding subtle signals in DNA sequences. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, pages 269–278, 2000.

[134] S. L. Porter, M. A. J. Roberts, C. S. Manning, and J. P. Armitage. A bifunctional kinase-phosphatase in bacterial chemotaxis. *Proceedings of the National Academy of Sciences of the United States of America*, 105(47):18531–18536, 2008.

[135] A. Prakash, M. Blanchette, S. Sinha, and M. Tompa. Motif discovery in heterogeneous sequence data. In *Proceedings of the 9th Pacific Symposium on Biocomputing*, pages 348–59, 2004.

[136] T. Pramila, S. Miles, D. GuhaThakurta, D. Jemiolo, and L. L. Breeden. Conserved homeodomain proteins interact with MADS box protein Mcm1 to restrict ECB-dependent transcription to the M/G1 phase of the cell cycle. *Genes & Development*, 16(23):3034–3045, 2002.

[137] M. N. Price, K. H. Huang, E. J. Alm, and A. P. Arkin. A novel method for accurate operon predictions in all sequenced prokaryotes. *Nucleic Acids Research*, 33(3):880–892, 2005.

[138] K. C. Quon, G. T. Marczynski, and L. Shapiro. Cell cycle control by an essential bacterial two-component signal transduction protein. *Cell*, 84(1):83–93, 1996.

[139] B. Ranson-Olson, D. F. Jones, T. J. Donohue, and J. H. Zeilstra-Ryalls. In vitro and in vivo analysis of the role of PrrA in *Rhodobacter sphaeroides* 2.4.1 *hemA* gene expression. *Journal of Bacteriology*, 188(9):3208–3218, 2006.

[140] E. Redhead and T. Bailey. Discriminative motif discovery in DNA and protein sequences using the DEME algorithm. *BMC Bioinformatics*, 8(1):385, 2007.

[141] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 1999.

[142] D. A. Rodionov and M. S. Gelfand. Computational identification of BioR, a transcriptional regulator of biotin metabolism in Alphaproteobacteria, and of its binding signal. *FEMS Microbiology Letters*, 255(1):102–107, 2006.

[143] F. P. Roth, J. D. Hughes, P. W. Estep, and G. M. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nature Biotechnology*, 16(10):939–945, 1998.

[144] K. E. Rudd. EcoGene: a genome sequence database for *Escherichia coli* K-12. *Nucleic Acids Research*, 28(1):60–64, 2000.

[145] T. D. Schneider and R. M. Stephens. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Research*, 18(20):6097–6100, 1990.

[146] G. Schwarz. Estimating the Dimension of a Model. *Annals of Statistics*, 6(2):461–464, 1978.

[147] T. Shimada, A. Ishihama, S. J. W. Busby, and D. C. Grainger. The *Escherichia coli* RutR transcription factor binds at targets within genes as well as intergenic regions. *Nucleic Acids Research*, 36(12):3950–3955, 2008.

[148] R. Siddharthan. Dinucleotide weight matrices for predicting transcription factor binding sites: Generalizing the position weight matrix. *PLoS ONE*, 5(3):e9722, 2010.

[149] R. Siddharthan, E. D. Siggia, and E. van Nimwegen. PhyloGibbs: A Gibbs sampling motif finder that incorporates phylogeny. *PLoS Computational Biology*, 1(7):e67, 2005.

[150] C. J. A. Sigrist, L. Cerutti, E. de Castro, P. S. Langendijk-Genevaux, V. Bulliard, A. Bairoch, and N. Hulo. PROSITE, a protein domain database for functional characterization and annotation. *Nucleic Acids Research*, 38(suppl 1):D161–D166, 2010.

[151] T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer. ROCR: visualizing classifier performance in R. *Bioinformatics*, 21(20):3940–3941, 2005.

[152] S. Sinha. On counting position weight matrix matches in a sequence, with application to discriminative motif finding. *Bioinformatics*, 22(14):e454–e463, 2006.

[153] S. Sinha, M. Blanchette, and M. Tompa. PhyME: a probabilistic algorithm for finding motifs in sets of orthologous sequences. *BMC Bioinformatics*, 5(170), 2004.

[154] S. Sinha, E. van Nimwegen, and E. D. Siggia. A probabilistic method to detect regulatory modules. *Bioinformatics*, 19(suppl 1):i292–i301, 2003.

[155] A. D. Smith, P. Sumazin, and M. Q. Zhang. Identifying tissue-selective transcription factor binding sites in vertebrate promoters. *Proceedings of the National Academy of Sciences of the United States of America*, 102(5):1560–1565, 2005.

[156] K. L. Smollett, K. M. Smith, C. Kahramanoglou, K. B. Arnvig, R. S. Buxton, and E. O. Davis. Global analysis of the regulon of the transcriptional repressor LexA, a key component of SOS response in *Mycobacterium tuberculosis*. *Journal of Biological Chemistry*, 287(26):22004–22014, 2012.

[157] M. Spivakov, J. Akhtar, P. Kheradpour, K. Beal, C. Girardot, G. Koscielny, J. Herrero, M. Kellis, E. Furlong, and E. Birney. Analysis of variation at transcription factor binding sites in *Drosophila* and humans. *Genome Biology*, 13:R49, 2012.

[158] E. Stackebrandt, R. G. E. Murray, and H. G. Trüper. *Proteobacteria* classis nov., a name for the phylogenetic taxon that includes the "purple bacteria and their relatives". *International Journal of Systematic Bacteriology*, 38(3):321–325, 1988.

[159] A. M. Stock, V. L. Robinson, and P. N. Goudreau. Two-component signal transduction. *Annual Review of Biochemistry*, 69(1):183–215, 2000.

[160] G. D. Stormo. DNA binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23, 2000.

[161] G. D. Stormo. Motif discovery using Expectation Maximization and Gibbs' sampling. In I. Ladunga, editor, *Computational Biology of Transcription Factor Binding*, volume 674 of *Methods in Molecular Biology*, pages 85–95. Springer, 2010.

[162] G. D. Stormo and G. W. Hartzell. Identifying protein-binding sites from unaligned DNA fragments. *Proceedings of the National Academy of Sciences of the United States of America*, 86(4):1183–1187, 1989.

[163] D. L. Swem and C. E. Bauer. Coordination of ubiquinol oxidase and cytochrome cbb_3 oxidase expression by multiple regulators in *Rhodobacter capsulatus*. *Journal of Bacteriology*, 184(10):2815–2820, 2002.

[164] K. Tamura, D. Peterson, N. Peterson, G. Stecher, M. Nei, and S. Kumar. MEGA5: Molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods. *Molecular Biology and Evolution*, 28(10):2731–2739, 2011.

[165] G. Thijs, M. Lescot, K. Marchal, S. Rombauts, B. De Moor, P. Rouzé, and Y. Moreau. A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics*, 17(12):1113–1122, 2001.

[166] G. Thijs, K. Marchal, M. Lescot, S. Rombauts, B. De Moor, P. Rouzé, and Y. Moreau. A Gibbs sampling method to detect over-represented motifs in the upstream regions of co-expressed genes. In *Proceedings of the 5th Annual International Conference on Computational Biology*, pages 305–312, 2001.

[167] W. Thompson, E. C. Rouchka, and C. E. Lawrence. Gibbs recursive sampler: finding transcription factor binding sites. *Nucleic Acids Research*, 31(13):3580–3585, 2003.

[168] M. Tompa, N. Li, T. L. Bailey, G. M. Church, B. De Moor, E. Eskin, A. V. Favorov, M. C. Frith, Y. Fu, W. J. Kent, V. J. Makeev, A. A. Mironov, W. S. Noble, G. Pavesi, G. Pesole, M. Régnier, N. Simonis, S. Sinha, G. Thijs, J. van Helden, M. Vandenbogaert, Z. Weng, C. Workman, C. Ye, and Z. Zhu. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23(1):137–44, Jan. 2005.

[169] J. T. Wade, D. B. Hall, and K. Struhl. The transcription factor Ifh1 is a key regulator of yeast ribosomal protein genes. *Nature*, 432(7020):1054–1058, 2004.

[170] J. T. Wade, N. B. Reppas, G. M. Church, and K. Struhl. Genomic analysis of LexA binding reveals the permissive nature of the *Escherichia coli* genome and identifies unconventional target sites. *Genes & Development*, 19(21):2619–2630, 2005.

[171] D. Wang, H. Xue, Y. Wang, R. Yin, F. Xie, and L. Luo. The *Sinorhizobium meliloti* ntrX gene is involved in succinoglycan production, motility, and symbiotic nodulation on alfalfa. *Applied and Environmental Microbiology*, 79(23):7150–7159, 2013.

[172] T. Wang and G. D. Stormo. Identifying the conserved network of cis-regulatory sites of a eukaryotic genome. *Proceedings of the National Academy of Sciences of the United States of America*, 102(48):17400–17405, 2005.

[173] G. C. Wei and M. Tanner. A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

[174] W. Wei and X.-D. Yu. Comparative analysis of regulatory motif discovery tools for transcription factor binding sites. *Genomics, Proteomics & Bioinformatics*, 5(2):131–142, 2007.

[175] T. Whitfield, J. Wang, P. Collins, E. C. Partridge, S. Aldred, N. Trinklein, R. Myers, and Z. Weng. Functional analysis of transcription factor binding sites in human promoters. *Genome Biology*, 13:R50, 2012.

[176] D. E. Whitworth. Classification and organisation of two component systems. In R. Gross and D. Beier, editors, *Two-Component Systems in Bacteria*, chapter 1. Caister Academic Press, 2012.

[177] K. P. Williams, B. W. Sobral, and A. W. Dickerman. A robust species tree for the Alphaproteobacteria. *Journal of Bacteriology*, 189(13):4578–4586, 2007.

[178] E. Wingender, P. Dietze, H. Karas, and R. Knüppel. TRANSFAC: a database on transcription factors and their DNA binding sites. *Nucleic Acids Research*, 24(1):238–241, 1996.

[179] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2nd edition, 2005.

[180] C. F. J. Wu. On the convergence properties of the EM algorithm. *Annals of Statistics*, 11(1):95–103, 1983.

[181] E. Xing, W. Wu, M. Jordan, and R. Karp. LOGOS: a modular Bayesian model for *de novo* motif detection. *Journal of Bioinformatics and Computational Biology*, 2(1):127–154, 2004.

[182] E. P. Xing, M. I. Jordan, R. M. Karp, and S. Russell. A hierarchical Bayesian Markovian model for motifs in biopolymer sequences. In *Proceedings of Advances in Neural Information Processing Systems*, pages 200–3, 2003.

[183] K. Yip, C. Cheng, N. Bhardwaj, J. Brown, J. Leng, A. Kundaje, J. Rozowsky, E. Birney, P. Bickel, M. Snyder, and M. Gerstein. Classification of human genomic regions based on experimentally determined binding sites of more than 100 transcription-related factors. *Genome Biology*, 13:R48, 2012.

[184] J. H. Zeilstra-Ryalls and S. Kaplan. Aerobic and anaerobic regulation in *Rhodobacter sphaeroides* 2.4.1: the role of the fnrL gene. *Journal of Bacteriology*, 177(22):6422–31, 1995.

[185] Y. Zhao and G. D. Stormo. Quantitative analysis demonstrates most transcription factors require only simple models of specificity. *Nature Biotechnology*, 29(6):480–483, 2011.

[186] T. Zhou, L. Yang, Y. Lu, I. Dror, A. C. D. Machado, T. Ghane, R. D. Felice, and R. Rohs. DNAshape: a method for the high-throughput prediction of DNA structural features on a genomic scale. *Nucleic Acids Research*, 41:W56–W62, 2013.

[187] J. Zhu and M. Q. Zhang. SCPD: a promoter database of the yeast *Saccharomyces cerevisiae*. *Bioinformatics*, 15(7):607–611, 1999.