Methods for Identifying Regulatory Grammars

by

Tahin Fahmid Syed

B.S., Computer Science, University of Minnesota (2010)
B.S., Biochemistry, University of Minnesota (2010)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2013

Author ....
Department of Electrical Engineering and Computer Science
November 20, 2012

Certified by ....
David K. Gifford
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by ....
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

Methods for Identifying Regulatory Grammars

by

Tahin Fahmid Syed

Submitted to the Department of Electrical Engineering and Computer Science
on November 20, 2012, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Recent advancements in sequencing technology have made it possible to study the mechanisms of gene regulation, such as protein-DNA binding, at greater resolution and on a greater scale than was previously possible. We present an expectation-maximization learning algorithm that identifies enriched spatial relationships between motifs in sets of DNA sequences. For example, the method will identify spatially constrained motifs colocated in the same regulatory region. We apply our method to biological sequence data and recover previously known prokaryotic promoter spacing constraints demonstrating that joint learning of motifs and spacing constraints is superior to other methods for this task.

Thesis Supervisor: David K. Gifford
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to thank Professor David Gifford for guiding my development as a researcher by demonstrating how scientific problems should be formulated and attacked and for the freedom that he gives his students to pursue their research interests. I would also like to thank Dr. Shaun Mahony who helped me get started in the lab and has provided numerous suggestions. Chris Reeder was also helpful for understanding the ChIA-PET data. Matt Edwards, Yuchun Guo, Tatsu Hasimoto, Charlie O'Donnell, and Jeanne Darling have made working in the Gifford Lab an enjoyable experience.

My friends at MIT and from Minnesota have provided support away from research.

Finally, I must thank Abbu and Ammu for their love and encouragement through all these years - their support made this all possible.

Contents

Contents 7

List of Figures 9

1 Introduction 11

1.1 Biological Background 11

1.2 Problem Statement 13

1.3 Related Work 13

1.4 Thesis Overview 15

2 Identifying grammars in DNA sequences 17

2.1 Preliminaries 17

2.1.1 Motif Representation 17

2.1.2 Grammar Representation 19

2.2 Identifying spaced dyads 19

2.2.1 EM algorithm 22

2.2.2 Significance Testing 22

3 Application 25

3.1 Synthetic Data 25

3.2 Prokaryotic promoters have motif spacing constraints 26

4 Conclusion 31

4.1 Future Work 31

A Derivation of EM algorithm **33**

A.1 M-step 34

A.2 E-step 35

Bibliography **37**

*

List of Figures

2-1 Example motif logo corresponding to the given example PWM table . 19

3-1 Distance distribution between motifs for toy example 26

3-2 Performance of dyad discovery drops as prevalence of dyad in synthetic dataset drops (averaged over multiple runs) 27

3-3 Distance distribution, calculated using our method, between the -10 and -35 motifs for a class of prokaryotic promoters 28

3-4 PWMs for prokaryotic promoter sequences 28

3-5 Distance distribution, calculated by motif scanning, between the -10 motif and -35 motifs for a class of prokaryotic promoters 29

3-6 Distance distribution, calculated by SpaMo, between the -10 motif (at displacement 0bp) and -35 motifs (at displacement -17) for a class of prokaryotic promoters 30

Chapter 1

Introduction

Multicellular organisms develop from the embryonic stage through a series of complex processes. These processes are dependent on the genes present in the organisms genome. Genes are expressed differentially, at specific times or locations, giving rise to hundreds of different cell types, including muscle cells, neurons in the brain, and insulin producing cells in the pancreas. A key overarching goal in biology is to understand the mechanism of differential gene regulation. Understanding this in great detail might reveal how diseases arise and how they can be treated.

1.1 Biological Background

Genes are encoded in deoxyribonucleic acid (DNA), which consists of four bases, or letter, Adenine (A), Thymine (T), Guanine (G), and Cytosine (C). DNA, abstractly, is present as a long string in the nucleus of a cell and forms the genome of the cell. In humans and mouse, this string is roughly 3 billion bases long. In the process of expression, DNA is first transcribed, or copied, into RNA. This RNA is then translated by ribosomes into proteins, which are often the functional product of the gene, which give rise to specific traits of an organism. A copy of the gene that has been transcribed and, subsequently, translated, is said to have been expressed.

Regulation of gene expression can occur at transcription, RNA processing or translation. DNA itself can be methylated, modulating transcription. Also, epigenetic modifications of histone proteins, which is the scaffold DNA wraps around, acquires chemical modifications, can increase or decrease the rate of transcription depending on the modification. RNA can be degraded as it is produced reducing the amount that can be expressed as protein. Specific RNAs can also bind to other RNA molecules, preventing their expression. There are numerous other examples and, likely, many have yet to be discovered.

We focus on one particular aspect of gene expression, transcription initiation. For a DNA to be transcribed into RNA, RNA polymerase must bind to the promoter of the gene, which is the region immediately before the transcription start site. In order for polymerase to bind, specific proteins called transcription factors (TFs), which interact directly with the DNA, must bind to the appropriate regulatory regions. There are various types of regulatory regions, including the promoter regions mentioned above. Enhancers are another class of regulatory region, in eukaryotes, which are located distal to a gene promoter. When TFs bind to enhancers, they are believed to interact with TFs bound at promoters through looping of the intervening DNA [24]. The set of specific TFs that bind at enhancer and promoter regions are known to be one of the key predictors of gene expression [26].

One of the problems in elucidating transcriptional regulation is identifying locations in the genome where TFs bind. The 3D structure of TFs typically cause them to bind at characteristic DNA sequence motifs. For example, a TF might be known to particularly favor ATTA DNA sequences. But, in a genome that is billion bases long, there are hundreds of thousands of occurrences of each of these motifs. ChIP-seq experiments have shown that a given TF might only bind to tens of thousands of these locations. Recently, the ENCODE project estimated that one out of 430 motif occurrences for a TF were actually bound [27]. Furthermore, only a few hundred out of the bound motifs may be functional and elicit a change in gene expression. Identifying and understanding functional TF binding sites, out of the large sets of non-functional binding sites and motif matches, is thus a key problem in elucidating gene regulatory networks.

Confounding factors complicate the identification of functional TF binding sites.

One complication is that many TFs bind cooperatively with other proteins to allow for signal integration and increased sensitivity of transcription [3] [6]. Sometimes a co-factor will not interact with the DNA directly, but rather through interactions with other TFs [31]. When cooperating factors do interact directly with the DNA, this can often be seen by the presence of a co-factor motif present near the motif of the given primary TF, often with a spatial constraint depending on the configuration of the interaction between the two transcription factors [25]. Furthermore, TFs located at the promoter may require a specific set of TFs to be bound at a distal enhancer region, which then interacts with the promoter. There are also specific requirements for chromatin opening to allow a TF to bind [11]. All of these interaction events are integrated at the gene promoter, which then drives expression [15] [22].

1.2 Problem Statement

We will learn combinations of motifs with particular spacing constraints that regulate gene expression. We term our representation of motif combinations, which we specifically define later, a *spaced dyad.* Most current computational methods for identifying TF binding sites do not take into account co-occurring motifs. We demonstrate that our method performs better than competing approaches.

1.3 Related Work

First, we briefly discuss some of the available motif finding methods and methods that aim to incorporate cofactors in motif searches.

Most current algorithms for finding motifs in a set of biological sequences learn a probabilistic model of the motif for an individual TF. A motif is usually modeled as a product of L multinomial distributions over {A, C, G, T}, where L is the length of the motif. More complex models of motifs have also been proposed, specifically those that break the independence assumption between the positions of a motif. However, independent product multinomials have been shown to be sufficiently accurate in most applications and, as a result, we choose to use them, as well.

Many such probabilistic learning algorithms for motifs have been developed, but the most popular use expectation-maximization, as in MEME[2], or Gibbs sampling[23] to learn an enriched motif over a background model. These have been shown to do quite well and, even though many of these methods are quite dated, they still show accurate performance on recent large sequencing datasets [18]. There are several variants of these motif finders - one occurrence per sequence (OOPS) and zero or one occurrence per sequence (ZOOPS) models find at most one occurrence of the motif in a given sequence, while two-component mixture models treat all subsequences of a given length independently and are able to find multiple occurrences of a motif in a sequence.

Non-probabilistic motif finding methods have also been developed, such as those that learn consensus sequences with mismatches or do simpler enrichment statistics [9]. A somewhat different class of motif finders use evolutionary conservation between species [20]. Since TFs play an important functional role, mutations in TF binding sites would be thought to have a deleterious effect on an organism. As a result, we would see these mutations selected against, resulting in conservation of these sites between species. We do not cover these methods here.

More recently, methods have been published that take TF cofactor motifs into account. Early methods included many cis-regulatory module finders [19] [32] [16] [13] [5] [12], which would find groups of motifs based on conservation and enrichment. Recent similar methods, such as co-Motif [10], learn two related motifs independently using an EM algorithm in a fashion similar to MEME. Some approaches to learning motif relationships typically learn the motifs independently and attempt to build pairwise relationships out of them by counting co-occurrences.

SpaMo [14], which we shall see again later, takes into account the spacing between the motifs. SpaMo uses previously discovered motifs and performs motif scans using a log-likelihood ratio test statistic to identify primary and secondary motif sites. Using these scanned locations, SpaMo performs a statistical test, under the null hypothesis of no spatial relationship between the motifs (assumed uniform distribution), to determine the significance of identified spacings. SpaMo's requirement of prespecified motifs, however, could limit performance.

Sequencing data is becoming more widespread, especially because of projects such as ENCODE [27] [8], and accurate technologies like the ones described above are generating data that will make analysis of multiple motifs together more common. Already, many transcription factors have been shown to bind together in a constrained fashion, playing an important role in processes like development. As a result, there is a need to develop methods that can perform these analyses to delve deeper into the complexity of gene regulation.

1.4 Thesis Overview

In Chapter 2, we cover some preliminaries about the representation of motifs and grammars as spaced dyads, describe the motivation for our approach, and present our main algorithm. In Chapter 3, we apply the method to a synthetic and a real biological dataset. In Chapter 4, we conclude our discussion with a summary of the thesis and discuss extensions and other related ideas for future work.

Chapter 2

Identifying grammars in DNA sequences

In this chapter, we introduce background required for the algorithms we present later. We then motivate some of the decisions made in this formulation and, finally, present the algorithm itself.

2.1 Preliminaries

2.1.1 Motif Representation

Here, we describe position weight matrix representation of a motif. Each position of a motif is a multinomial distribution over the alphabet A, C, G, T that is independent of the other positions in the motif. An example of a PWM matrix for a 5bp long motif [Table 2.2], along with a pictorial representation of the information content of the same motif [Figure 2-1], is given. Information content is determined by how far a PWM position is from a uniform distribution.

Table of Notation

Symbol	Description
W	Set of all observed subsequences
W_n	N^{th} observed subsequence
$W^k[i]$	Set of substrings corresponding to motif i for k^{th} dyad for all observed subsequences
$W^k_n[i]$	Substrings corresponding to motif i for k^{th} dyad for the n_{th} sequence
$\gamma[B_n]$	$P(B_n = 1 \mid W_n;\ \theta)$ - Responsibility of background for producing subsequence W_n
$\gamma[Z_{nk}]$	$P(Z_n = k \mid W_n, B_n \neq 1;\ \theta)$ - Responsibility of background for producing subsequence W_n
π	Vector of prior probabilities $[\pi_{BG}, \pi_G]$ for background and grammar, where $\pi_G = 1 - \pi_{BG}$
λ	Vector of prior probabilities $[\lambda_1 \ldots \lambda_k]$ for k spaced dyads
θ	Vector of parameters $[\theta_{BG}, \theta_1 \ldots \theta_k]$ for background and k spaced dyads
π_{BG}	$P(B_n = 1)$ - Prior probability of subsequence W_n being from background
λ_k	$P(Z_n = k)$ - Prior probability of subsequence being from spaced dyad k
θ_k	Parameters of spaced dyad k - consists of motif parameters $[\theta^1_k, \theta^2_k]$ and spacing parameter g bp
θ^i_k	Parameters of independent product multinomials for motif i in k^{th} dyad
θ_{BG}	Parameters of background (assumed to be uniform)
Z_n	$Z_n \in \{1..k\}$ indicator of grammar that generated W_n
B_n	$B_n \in \{0, 1\}$ indicates whether background generated W_n
K	Number of dyads
N	Number of subsequences generated from original data
L	Length of each subsequence W_n

Table 2.1: Table of Notation used in this chapter

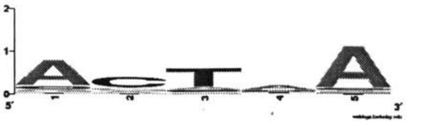


Figure 2-1: Example motif logo corresponding to the given example PWM table

A	0.75	.1	.1	.5	.85
C	0.1	.6	.1	.2	.05
G	0.1	.2	.1	.1	.05
T	0.05	.1	.7	.2	.05

Table 2.2: PWM table

Given a PWM matrix, we can calculate the likelihood of a sequence having been generated by that PWM by simply evaluating the likelihood of generating the observed base at each position from each multinomial.

2.1.2 Grammar Representation

A *spaced dyad* is a pair of motifs and single number, g, indicating the spatial preference between the two motifs in base pairs. We call this representation a spaced dyad. To improve the performance of our method, we trim weakly informative bases off the end of each motif. Other methods, such as SpaMo have adopted similar conventions for grammars. For example, an example of a grammar that has previously been supported is a Stat3 motif located 1 base pair (bp) upstream of a YY1 motif. More complex grammars can be assembled out of these simple ones. For example, Stat3 has also been shown to be preferentially present 7bp upstream of a Hdx motif. Combining these two simple dyads, we can construct a complex grammar involving Stat3 and both YY1 and Hdx.

Formally, a spaced dyad model is two motif models whose motif parameters are θ_1 and θ_2, respectively, and a spacing parameter in base pairs, g, between them.

2.2 Identifying spaced dyads

In a two-component mixture model for motif finding, a dataset of DNA sequences is split such that all subsequences in the dataset of a given length L are treated independently. Then, for each subsequence, the model assigns responsibilities for this subsequence having been generated from a background or a motif model by performing maximum likelihood estimation by an expectation maximization (EM) algorithm. The advantage this has over the other motif finding models described earlier (such as OOPS and ZOOPS) is that it allows us to find multiple occurrences of a motif in the original sequences in the dataset.

Considering the advantages of the two-component mixture (TCM) model above, we adopt a similar framework for spaced dyads, since each sequence in the original dataset could have multiple occurrences of the spaced dyads we described earlier. Finding multiple spaced dyads in each sequence could also allow us to construct more complex grammars.

Like TCM, we first split the input sequences into subsequences of a given length. Then, the generative process for each subsequence is as follows:

For each subsequence

1. Choose whether the sequence was generated by background or a dyad
2. If generated by a dyad,
   - Choose a dyad, out of a user specified set (which vary by either motif or spacing), which generated the sequence
3. Generate the sequence from background or the appropriate dyad

We further justify splitting our dataset of sequences into subsequences of a given length. If we take a subsequence that is known to be generated from a particular dyad and shift one base in the original sequence to get a new subsequence, it is likely that we have significantly reduced the likelihood of this subsequence being generated from the same dyad. Also, since we are dealing with all possible subsequences of a given length, we assume that if a subsequence is generated by a dyad, it starts at the first position of the subsequence. In other words, the first motif in the dyad is lined up with the beginning of the subsequence. As a result, this has the effect of simply changing responsibilities in a way that, intuitively, "slide" the second motif in the dyad to the appropriate spot, such that the dyad is most likely to have generated the given subsequence.

It is still necessary to include a background model, which will prevent probability mass from those sequences which are dissimilar to any of the spaced dyads from diluting the motif models or the spacings. For now, we use a fixed background, assuming bases are generated according to a uniform distribution.

Here, we describe our formulation and optimization, with more details presented in Appendix A. Given a dyad model, we can evaluate the likelihood of a sequence according to the model, $P(W_n \mid Z_n = k;\ \theta_k, \lambda_k)$, where $Z_n = k$ specifies the latent dyad model, $\theta_k = [\theta_k^1, \theta_k^2, g]$, out of K different models. If we are given an observed subsequence W_n, we test for an occurrence of the first motif, θ_k^1, at the beginning of W_n and an independently test an occurrence of the second motif, θ_k^2, g bp after the end of the first motif. The remaining portion of the sequence is assumed to be background. The log-likelihood of a particular sequence is given by:

$$P(W_n, Z_n;\ \pi, \lambda, \theta) = \pi_{BG} P(W_n \mid B_n = 1;\ \theta_{BG}) + (1 - \pi_{BG}) \sum_{k=1}^{K} P(W_n \mid Z_n = k;\ \theta_k, \lambda_k) \tag{2.1}$$

Here, B_n is a latent binary variable indicating whether the sequence does not correspond to any grammar (ie. the sequence is generated by the uniform background).

To find the parameters of the dyads we would like to optimize the observed data log-likelihood. Optimization of this function is difficult. So, as is usually done in mixture model settings with latent variables, we will, instead, optimize the complete-data log-likelihood [Eq. 2.2] using an EM algorithm [1].

$$P(W, Z;\ \pi, \lambda, \theta) = \sum_{n} \log P(W_n, Z_n;\ \pi, \lambda, \theta) \tag{2.2}$$

Note that if we fix the motif parameters and do not update them as part of the optimization, the only difference between dyads will be the spacing. As a result, we can learn a distribution over motif spacings. However, in the algorithm presented below, we do update the motif parameters.

2.2.1 EM algorithm

We present an EM algorithm to iteratively update the log-likelihood and learn the desired parameters of the spaced dyads. The EM algorithm consists of two steps [4]. The E-step computes the responsibility of each component for producing the data, using the parameters given in initialization or computed in the immediately prior M-step. Bayes' theorem is used to compute the conditional probability of the latent variable, in this case the assignment to a particular dyad, given the data.

Now, given the responsibilities computed from the E-step, we reestimate the parameters of the model, through optimization of the complete data log-likelihood.

EM only converges to a local optimum and typically requires around 30 iterations in this particular application. A detailed derivation of EM is provided in Appendix A.

2.2.2 Significance Testing

We apply a parametric statistical test to evaluate the significance of the recovered spacings. Briefly, we test whether the number of occurrences of a secondary motif at a particular distance from the primary motif is greater than would be expected by a uniform distribution. We apply a binomial test at each spacing distance with a Bonferroni correction for the number of spacing distances tested. This significance test is similar to the one presented in the SpaMo paper.

Algorithm 1: EM performs an iterative update of the dyad models

```
Input: DNA subsequences of length $L$ extracted from the original dataset and
       initial dyad parameter settings
Output: Parameters for dyads and responsibilities
while Not converged or termination condition not reached do
    // E-step - iteratively estimate responsibilities for each subsequence
    for $i \leftarrow 1$ to $N$ do
        // Responsibility of background for producing $W_n$
        $\gamma[B_n] = \frac{\pi_{BG} P(W_n|B_n{=}1;\, \theta_{BG})}{\pi_{BG} P(W_n|B_n{=}1;\, \theta_{BG}) + (1-\pi_{BG}) \sum_{k=1}^{K} \lambda_k P(W_n|Z_n{=}k;\, \theta)}$
        // Relative responsibility of each dyad for producing $W_n$
        $\gamma[Z_{nk}] = \frac{\lambda_k P(W_n|Z_n{=}k;\, \theta)}{\sum_{k=1}^{K} \lambda_k P(W_n|Z_n{=}k;\, \theta)}$
    // M-step - iteratively update parameters of each dyad model
    for $k \leftarrow 1$ to $K$ do
        // Update first motif in dyad $k$
        $\theta_k^1 = \text{LEARN-MOTIF}(W^k[1])$
        // Update second motif in dyad $k$
        $\theta_k^2 = \text{LEARN-MOTIF}(W^k[2])$
        // Update mixture weights
        $\lambda_k = \sum_{n=1}^{N} \gamma[Z_{nk}] \;/\; N$
    $\pi_{BG} = \sum_{n=1}^{N} \gamma[B_n] \;/\; N$
```

Algorithm 2: LEARN-MOTIF Learns a PWM model from a set of sequences and associated weights

```
Input: N DNA subsequences each of length J (the length of the motif) and
       responsibilities of $k^{th}$ dyad (the one currently being updated) and
       background for those sequences
Output: PWM table with elements $p_{ck}$
$\mathbb{A} = \{A,C,G,T\}$
// Over each position in the motif
for j ← 1 to J do
    for c ∈ $\mathbb{A}$ do
        $p_{cj} = \frac{\sum_{n=1}^{N} \mathbf{1}[a_{nj}=c]\gamma[Z_{nk}](1-\gamma[B_n])}{\sum_{n=1}^{N} \sum_{c' \in \mathbb{A}} \mathbf{1}[a_{nj}=c']\gamma[Z_{nk}](1-\gamma[B_n])}$
        // $a_{nj}$ is the character at position j in subsequence N; $\mathbb{1}[a_{nj} = c] = 1$ iff
        that character is c
        // The gamma variables indicate responsibilities - more details in Table
        of Notation and Appendix A
return p
```

Chapter 3

Application

In this chapter, we test the hypothesis that joint learning of motifs and spacing constraints performs better than a naive motif scanning approach and SpaMo at identifying biologically significant grammars by applying these methods to prokaryotic promoter sequences.

3.1 Synthetic Data

We begin with a small toy example to demonstrate how our method should work. We plant artificial dyads of ATGCA and TGCA with a spacing of 3bp at random positions in 100 sequences of length 50 on the same DNA strand. We first initialize the algorithm and limit ourselves to the interval of -5 to +10bp to search for spacings enrichment. The output we get is a histogram showing the enrichment of particular spacings. As we can see, the 3bp spacing we expect to see is enriched ($p \simeq 10^{-10}$) [Figure 3-1]. But, we see that over half of the probability mass has been allotted to other spacings. This is due to subsequences which are not instances of the grammar being assigned limited probability mass to background. This dilutes the quality of the grammars we hope to recover.

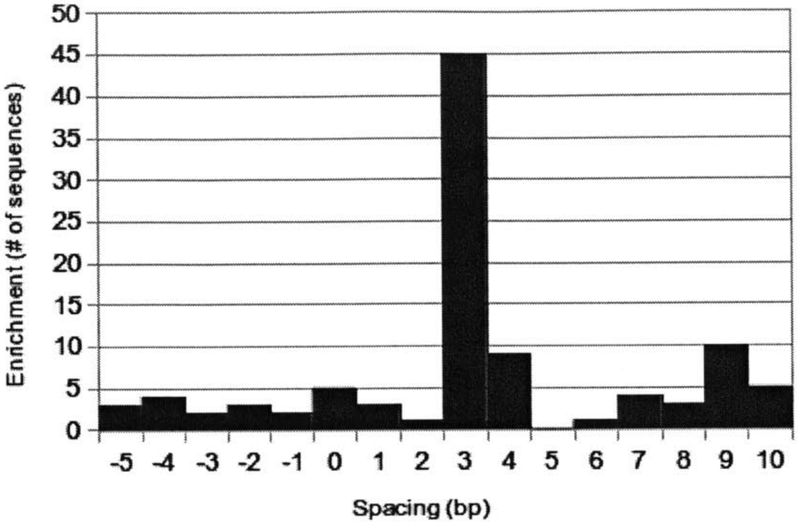


Figure 3-1: Distance distribution between motifs for toy example

We also varied the number of sequences which contained the dyad to estimate how robust the method is to noisy data. We see a drop in the recovery of dyads as the presence of the dyad in the dataset drops [Figure 3-2]. When only half the dataset contains the dyad, we see recovery of under 1/5 of the sequences containing the dyad.

3.2 Prokaryotic promoters have motif spacing constraints

Promoter regions have long been known to contain specific sets of binding motifs with spacers in between them. These spacers are a signature of the transcriptional machinery, comprised of general TFs and RNA polymerase, that forms when a gene is expressed. Prokaryotic promoters have been particularly well studied, as they seem to exhibit greater sequence conservation when compared to eukaryotes. Since prokaryotic organisms have smaller genomes and lack enhancers, most transcriptional regulation occurs at these gene proximal regions. Several databases of prokaryotic promoter sequences have been published and here we apply the method to data from

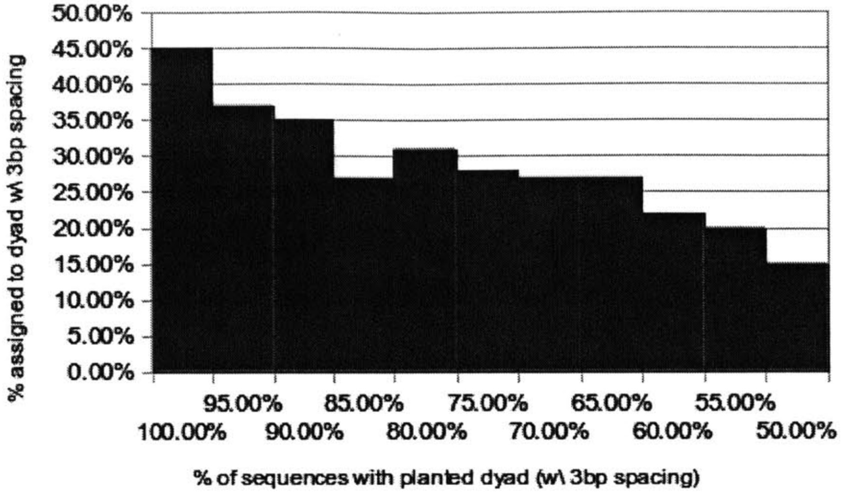


Figure 3-2: Performance of dyad discovery drops as prevalence of dyad in synthetic dataset drops (averaged over multiple runs)

the PromEC database of E. coli σ-70 promoters [29]. These promoters have been shown to have a consensus TTGATC motif at -35, a conserved 15-19bp spacer, and a TATAAA -10 motif near the transcription start site [17] . In particular, we apply our method to find the distribution over the distances between the two motifs, with motif parameters fixed, as a proof of concept. We apply what we know about the motifs present in these promoter regions and limit our search to spacings upto 30bp long. For all graphs, we only show results corresponding to the strand with greatest enrichment at a particular position.

We see that the most significant spacings found are 15-18bp (p $\simeq 10^{-6}$ at 17bp) [Figure 3-3]. Also, if we allow the motif parameters to update, we obtain PWMs that are similar to the consensus sequences [Figure 3-4].

We compared our method to a naive scanning approach. Using published -35 and -10 motif PWMs [7], we scanned the PromEC sequences for motif occurrences and constructed a histogram of the displacements between the motifs [Figure 3-5]. The histogram reveals very little about the structure of the underlying grammar. This

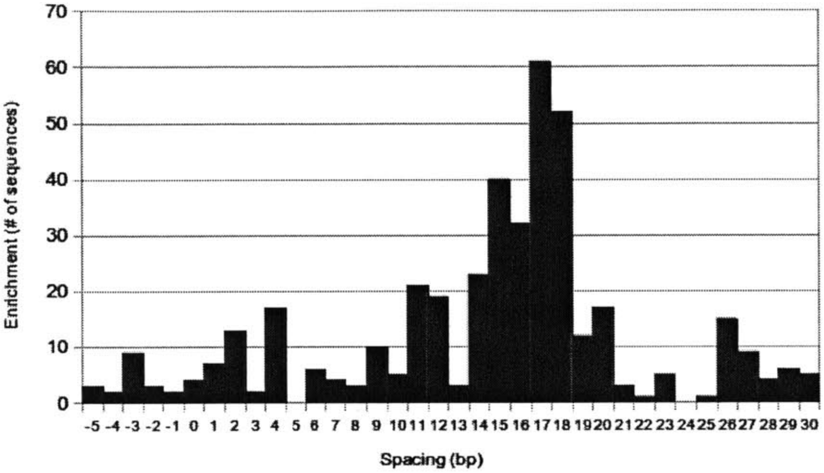
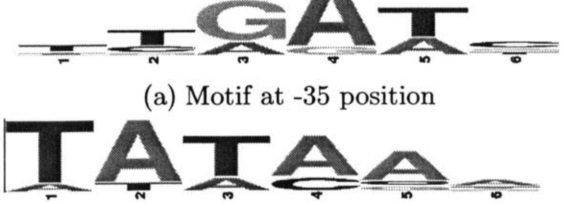


Figure 3-3: Distance distribution, calculated using our method, between the -10 and -35 motifs for a class of prokaryotic promoters



(a) Motif at -35 position

(b) Motif at -10 position

Figure 3-4: PWMs for prokaryotic promoter sequences

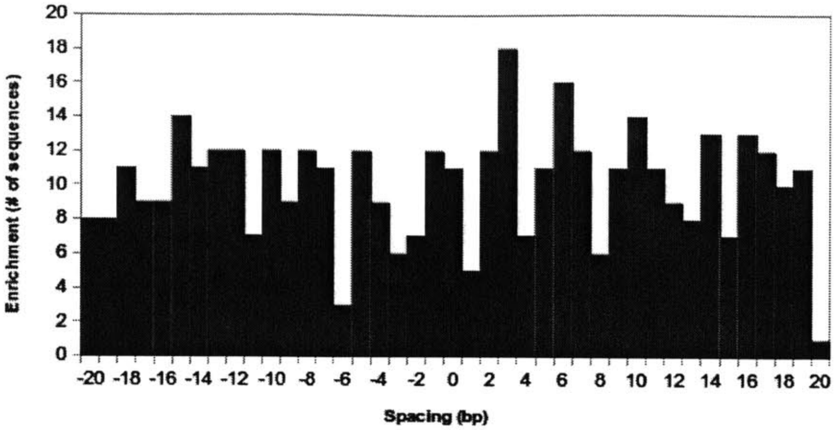


Figure 3-5: Distance distribution, calculated by motif scanning, between the -10 motif and -35 motifs for a class of prokaryotic promoters

may be caused by some weakly informative bases in the input motif, which may lead to falsely calling motif occurrences.

Finally, we applied SpaMo to the PromEC database using the same published PWM matrices as above [Figure 3-6]. SpaMo locates the -35 motif at a displacement of -17bp from the -10 motif, as we would expect (p $\simeq 10^{-10}$). However, the signal is considerably weaker, with only 35 occurrences of the most significant spacing found by SpaMo, compared to 61 occurrences found by our method. It also picks up a few more weakly significant spacings at -3bp and 7bp, which do not reflect any known biology.

This application to prokaryotic promoter data shows that learning motifs and spacing constraints simultaneously avoids the problems we see in the naive approach and SpaMo. By allowing the motifs to inform the learned spacing preferences and vice versa and not requiring prespecified, potentially weak, motifs, we are able to observe biologically relevant spacing constraints and fewer false positives in our learning task.

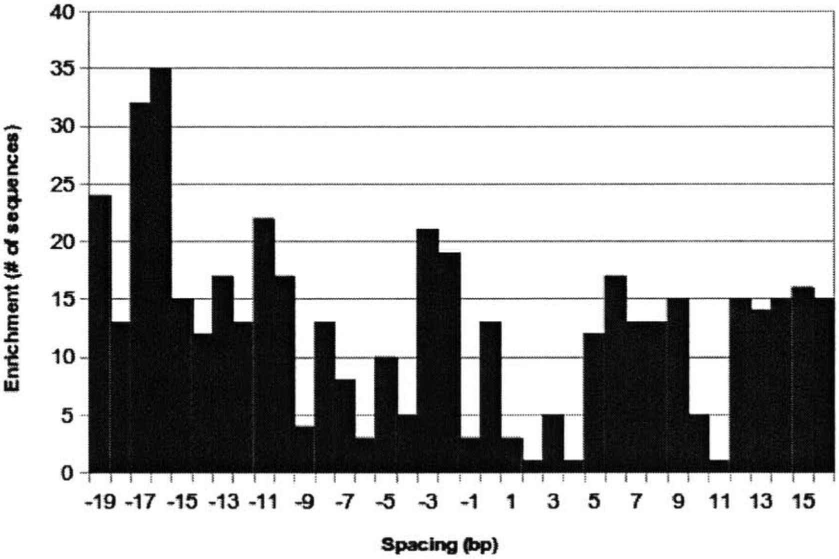


Figure 3-6: Distance distribution, calculated by SpaMo, between the -10 motif (at displacement 0bp) and -35 motifs (at displacement -17) for a class of prokaryotic promoters

Chapter 4

Conclusion

In this thesis, we considered the tasking of learning more complex sequence representations of DNA, rather than simple motifs. We discussed the importance of these representations and outlined a method that can recover previously known spacings between motifs in prokaryotic promoter datasets and perform better than other methods at this task.

4.1 Future Work

There are several avenues for future work

- First, we adopted a simple representation of a grammar as two spaced motifs. More complex representations would likely allow for fewer matches in a set of DNA sequences, resulting in greater specificity of discovered grammars.

- Second, the current framework allows for a sequence to be generated from background or one out of a set of grammars. This could be augmented to allow classes of grammars. For example, we might expect that the binding context of a factor might differ when it is bound in an enhancer region versus a promoter region. A model that added an extra level to capture these biological notions would allow for greater interpretability and also limit weak assignments of a sequence to a grammar.

- Adding additional informative priors could aid in discovery. Priors have been used with other motif finders and shown to improve motif discovery results and similar priors over the subsequences could aid in our problem setting [30].
- A different formulation may also improve performance. One limitation of the current model is that if the number of dyads grows too large there is a corresponding increase in the number of mixture components, which, in general, reduces the performance of mixture models, due to overfitting.
- This algorithm is reminiscent of the MEME two component mixture algorithm, which has been applied to proteins. With minor extension, this algorithm could also be applied to proteins, which have also been shown to have domains with specific spacing constraints.

Appendix A

Derivation of EM algorithm

In this section, we provide some intuition and a derivation for the algorithm in Chapter 2. Refer to Table 2.1 for notation. For the model presented, the likelihood of a particular sequence is given by:

$$P(W_n, Z_n;\ \pi, \lambda, \theta) = \pi_{BG} P(W_n \mid B_n = 1;\ \theta_{BG}) + (1 - \pi_{BG}) \sum_k P(W_n \mid Z_n = k;\ \theta_k, \lambda_k) \tag{A.1}$$

Therefore, for the entire dataset, the complete-data log-likelihood is

$$P(W, Z;\ \pi, \lambda, \theta) = \sum_n \log P(W_n, Z_n;\ \pi, \lambda, \theta) \tag{A.2}$$

where we have introduced a latent variable Z_n for each subsequence to indicate the hidden component that is responsible for generating the sequence and B_n which is a binary variable that indicates whether the sequence was generated by the background model.

Optimization [??] of the observed-data log-likelihood is not tractable and as a result, we instead introduce latent variables and iteratively optimize the complete-data log-likelihood [A.2], since we do not know which latent component generated each observed subsequence. This EM algorithm is used to estimate the parameters of the model by fitting the data with the model using maximum likelihood estimation.

The derivation below follows some of the conventions presented in derivations of probabilistic latent semantic analysis [28] [21].

A.1 M-step

We first derive the maximization step of EM, which assumes that we have computed responsibilities, or expectations of the latent variables, in the E-step and can now perform optimization of the complete data log-likelihood. Here, we focus on deriving the motif updates, as the mixing parameter updates follow the procedure that is typical of mixture models.

We update parameters for each dyad component independently. We further decompose this function into learning two individual motifs based on the profile of the current dyad we are learning. Specifically, we know the lengths of the two motifs and the spacing parameter composing the dyad, which allows us to learn the motifs independently. As a result, in the following steps, we consider updates for only one motif of dyad k. The updates for the other motif are analogous.

Let $W^k[i]$ denote the substrings corresponding to motif i for the k^{th} dyad. There are N substrings in this set. Let a_{nj} be the character at the j^{th} of the n^{th} substring from this set. We can evaluate the log-likelihood of the set of sequences, $W^k[i]$, according to the k^{th} dyad model: (where $c \in \{A, C, G, T\}$, p_{cj} is the probability of character c in position j according to motif i for the k^{th} dyad, and $\mathbb{1}$ is the indicator function which returns 1 only if c matches a_{nj})

$$\sum_n \sum_j \sum_c \mathbb{1}[a_{nj} = c]\mathbb{1}[B_n = 1] \log[\pi_{BG} P(a_{nj} \mid \theta_{BG})] + \\ \sum_n \sum_j \sum_c \mathbb{1}[a_{nj} = c]\mathbb{1}[B_n \neq 1]\mathbb{1}[Z_n = k] \log[(1 - \pi_{BG})p_{cj}] \quad \text{(A.3)}$$

We then take expectations of the latent variables and simplify to get:

$$\sum_n \sum_j \sum_c \mathbb{1}[a_{nj} = c]\mathbb{E}[\mathbb{1}[B_n = 1]] \log[\pi_{BG} P(a_{nj} \mid \theta_{BG})] +$$

$$\sum_n \sum_j \sum_c \mathbb{1}[a_{nj} = c]\mathbb{E}[\mathbb{1}[B_n \neq 1]\mathbb{1}[Z_n = k]] \log[(1 - \pi_{BG}) p_{cj}] \quad \text{(A.4)}$$

Our goal is to determine the parameters p_{cj}. Therefore, we include Lagrange multiplier constraints and differentiate with respect to p_{cj} for a particular c and j.

$$\frac{\sum_c \mathbb{1}[a_{nj} = c]\mathbb{E}[\mathbb{1}[B_n \neq 1]\mathbb{1}[Z_n = k]]}{p_{cj}} + C \quad \text{(A.5)}$$

where C is a constant that results from differentiating the Lagrange terms.

Setting the derivative equal to zero and solving for p_{cj} gives:

$$p_{cj} \propto \sum_c \mathbb{1}[a_{nj} = c]\mathbb{E}[\mathbb{1}[B_n \neq 1]\mathbb{1}[Z_n = k]] \quad \text{(A.6)}$$

After normalizing and recognizing that the expectations in the expressions correspond to the expectations computed in the E-step, we get the expression for p_{cj} as presented in Algorithm 2.

$$p_{cj} = \frac{\sum_{n=1}^{N} \mathbb{1}[a_{nj} = c]\gamma[Z_{nk}](1 - \gamma[B_n])}{\sum_{n=1}^{N} \sum_{c' \in \mathbb{A}} \mathbb{1}[a_{nj} = c']\gamma[Z_{nk}](1 - \gamma[B_n])} \quad \text{(A.7)}$$

A.2 E-step

Using the initialized parameters of the model or the parameters estimated in the preceding M-step, we update our expectations for the latent variables using Bayes' Rule.

We first compute the responsibility of a sequence having been generated by background:

$$\begin{aligned} P(B_n = 1 \mid W_n;\ \theta) &= \frac{P(W_n \mid B_n = 1;\ \theta)P(B_n = 1)}{P(W_n)} \\ &= \frac{P(W_n \mid B_n = 1;\ \theta_{BG})\pi_{BG}}{\pi_{BG}P(W_n \mid B_n = 1;\ \theta_{BG}) + (1 - \pi_{BG})\sum_k \lambda_k P(W_n \mid Z_n = k;\ \theta)} \\ &= \gamma[B_n] = \mathbb{E}[\mathbb{1}[B_n = 1]] \end{aligned} \quad (A.8)$$

The result we get can be interpreted as the parameter of a binomial distribution over assignment of a sequence to background or non-background.

We then compute relative responsibilities for each of the dyad models:

$$\begin{aligned} P(Z_n = k \mid Z_n \neq B, W_n) &= \frac{P(W_n \mid Z_n = k)P(Z_n = k)}{P(W_n)} \\ &= \frac{P(W_n \mid Z_n = k)\lambda_k}{\sum_k P(W_n \mid Z_n = k;\ \theta, \lambda)} \\ &= \gamma[Z_{nk}] = \mathbb{E}[\mathbb{1}[Z_n = k]] \end{aligned} \quad (A.9)$$

The results we get for all k models can be interpreted as the parameters of a multinomial distribution over the dyads.

Bibliography

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

[2] T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Technical Report CS94-351, University of California at San Diego, 1994.

[3] A. S. Bais, N. Kaminski, and P. V. Benos. Finding subtypes of transcription factor motif pairs with distinct regulatory roles. *Nucleic Acids Research*, doi:10.1093:1–13, 2011.

[4] C. Bishop. *Pattern recognition and machine learning.* Springer, 2006.

[5] A. D. Smith et al. Mining ChIP-chip data for transcription factor and cofactor binding sites. *Bioinformatics*, 21:i403–i412, 2005.

[6] C. Cheng et al. Understanding transcriptional regulation by integrative analysis of transcription factor binding data. *Genome Research*, 22:1658–1667, 2012.

[7] Huerta AM et al. Selection for unequal densities of 70 promoter-like signals in different regions of large bacterial genomes. *PlOS Genetics*, 2(11):e185, 2006.

[8] J. Wang et al. Sequence features and chromatin structure around the genomic regions bound by 119 human transcription factors. *Genome Research*, 22:1798–1812, 2012.

[9] M. C. Firth et al. Discovering Sequence Motifs with Arbitrary Insertions and Deletions. *PloS Comp. Biol.*, 4(5): e1000071. doi:10.1371/journal.pcbi.1000071, 2008.

[10] M. Xu et al. comotif: A mixture framework for identifying transcription factor and a co-regulator motif in chip-seq data. *Bioinformatics*, 2011.

[11] S. Neph et al. An expansive human regulatory lexicon ecoded in transcription factor footprints. *Nature*, 489:83–90, 2012.

[12] S. Sinha et al. A probabilistic method to detect regulatory modules. *Bioinformatics*, 19:292–301, 2003.

[13] T. Lin et al. BayCis: A Bayesian Hierarchical HMM for Cis-Regulatory Module Decoding in Metazoan Genomes. *RECOMB*, pages 66–81, 2008.

[14] T. Whittington et al. Inferring transcription factor complexes from chip-seq data. *Nucleic Acids Research*, 39 (15):e98, 2011.

[15] P. J. Farnham. Insights from genomic profiling of transcription factors. *Nat. Rev. Genet.*, 10:605–616, September 2009.

[16] M. C. Frith, M. C. Li, and C. Weng. Cluster-Buster: finding dense clusters of motifs in DNA sequences. *Nucleic Acids Research*, 31:3666–3668, 2003.

[17] Leo Gordon, Alexey Ya. Chervonenki, Alex J. Gammerman, Ilham A. Shahmuradov, and Victor V. Solovyev. Sequence alignment kernel for recognition of promoter regions. *Bioinformatics*, DOI: 10.1093/bioinformatics/btg265:1964–1971, 2003.

[18] Y. Guo, S. Mahony, and D. K. Gifford. High resolution genome wide binding event finding and motif discovery reveals transcription factor spatial binding constraints. *PloS Comp. Bio.*, 8(8):e1002638, August 2012.

[19] M. Gupta and J. S. Liu. De novo cis-regulatory module elicitation for eukaryotic genomes. *PNAS*, 102:7079–7084, 2005.

[20] R. C. Hardison and J. Taylor. Genomic approaches towards finding cis-regulatory modules in animals. *Nat. Rev. Genet.*, 13:469–483, July 2012.

[21] Hong L. A tutorial on probabilistic latent semantic analysis. Technical report, Lehigh university, 2010.

[22] B. Lenhard, A. Sandelin, and P. Carninci. Metazoan promoters: emerging characteristics and insights into transcriptional regulation. *Nat. Rev. Genet.*, doi:10,1038/nrg3163:1–14, April 2012.

[23] X Liu, D.L. Brutlag, and J.S. Liu. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac. Symp. Ciocomput.*, pages 127–138, 2001.

[24] S. Malik and R.G. Roeder. The metazoan Mediator co-activator complex as an integrative hub for transcriptional regulation. *Nat. Rev. Genetics*, doi:10.1038/nrg2901, 2010.

[25] K. Noto and M. Craven. Learning probabilistic models of cis-regulatory modules that represent logical and spatial aspects. *Bioinformatics*, doi:10.1093:e156–e162, 2006.

[26] C. Ong and V. G. Corces. Enhancer function: new insights into the regulation of tissue-specific gene-expression. *Nat. Rev. Genet.*, 12:283–293, April 2011.

[27] ENCODE project consortium. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489:57–74, 2012.

[28] Mei Q and Zhai C.X. A note on em algorithm for probabilistic latent semantic analysis. Technical report, University of Illinois at Urbana-Champaign.

[29] Gill Bejerano Alberto Santos-Zavaleta Ruti Hershberg and Hanah Margalit. PromEC: An updated database of Escherichia coli mRNA promoters with experimentally identified transcriptional start sites. *Nucleic Acids Research*, 29:277–00, 2001.

[30] M Boden T Whittington T.L. Bailey and P Machanick. The value of position-specific priors in motif discovery using MEME. *BMC Bioinformatics*, 11:179, 2010.

[31] J. O. Yanez-Cuna et al. Uncovering cis-regulatory sequence requirements for context-specific transcription factor binding. *Genome Research*, doi/10.1101/gr.132811.111, 2012.

[32] Q Zhou and W. H. Wong. CisModule: De novo discovery of cis-regulatory modules by hierarchical mixture modeling. *PNAS*, 101:12114–12119, 2004.