2012-06-22

XPRIME-EM: Eliciting Expert Prior Information for Motif Exploration Using the Expectation-Maximization Algorithm

Wei Zhou
Brigham Young University - Provo

XPRIME-EM: Eliciting Expert Prior Information for Motif Exploration Using the Expectation-Maximization Algorithm

Wei Zhou

A selected project submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

David A. Engler, Chair
Natalie J. Blades
Scott D. Grimshaw

Department of Statistics

Brigham Young University

August 2012

ABSTRACT

XPRIME-EM: Eliciting Expert Prior Information for Motif Exploration
Using the Expectation-Maximization Algorithm

Wei Zhou
Department of Statistics, BYU
Master of Science

Understanding the possible mechanisms of gene transcription regulation is a primary challenge for current molecular biologists. Identifying transcription factor binding sites (TFBSs), also called DNA motifs, is an important step in understanding these mechanisms. Furthermore, many human diseases are attributed to mutations in TFBSs, which makes identifying those DNA motifs significant for disease treatment. Uncertainty and variations in specific nucleotides of TFBSs present difficulties for DNA motif searching. In this project, we present an algorithm, XPRIME-EM (Eliciting EXpert PRior Information for Motif Exploration using the Expectation-Maximization Algorithm), which can discover known and *de novo* (unknown) DNA motifs simultaneously from a collection of DNA sequences using a modified EM algorithm and describe the variation nature of DNA motifs using *position specific weight matrix (PWM)*. XPRIME improves the efficiency of locating and describing motifs by prevent the overlap of multiple motifs, a phenomenon termed a *phase shift*, and generates stronger motifs by considering the correlations between nucleotides at different positions within each motif. Moreover, a Bayesian formulation of the XPRIME algorithm allows for the elicitation of prior information for motifs of interest from literature and experiments into motif searching. We are the first research team to incorporate human genome-wide nucleosome occupancy information into the PWM based DNA motif searching.

Keywords: DNA motif, modified EM algorithm, human nucleosome occupancy information

ACKNOWLEDGMENTS

In the first place, I would like to express my deepest appreciation to Dr. Evan Johnson and Dr. David Engler for their amazing ideas, advice and supervision to my project and thesis. Dr. Johnson introduced me to the field of DNA motif searching, whose enthusiasm for the adventure regarding the research had lasting effects. Under his supervision, the project worked so organized. Working in his lab was a wonderful experience in my life. Without Dr. Johnson and Dr. Engler's guidance and persistent help, it would have been impossible for me to finish this thesis.

I would like to thank my committee members for their precious time and advice for my project. I also would like to thank my husband Shuai Wei and my parents for their support and encouragement for me to overcome all difficulties.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Basics of gene expression

All living things, such as plants, animals, bacteria, viruses and fungi, depend on their genetic information for inheritance. Nucleic acids are large molecules that carry all genetic information and there are two types of nucleic acids: the deoxyribonucleic acid (DNA) and the ribonucleic acid (RNA). For all complex organisms, DNA is the molecule that carries genetic information.

A gene is a segment of DNA and is defined as a fundamental heredity unit. The genome is a complete copy of the entire set of genes in an organism. DNA consists of two long polymer sequences, which are made up with four different molecules called *nucleotides*. A base attaches to the phosphate with a sugar ring to form a nucleotide. There are four different bases in DNA: A(Adenine), C(Cytosine), G(Guanine) and T(Thymine). DNA consists of two complementary strands of nucleotides which bind together for a double helix. In this structure, each individual type of nucleotide only interacts with one other type of nucleotide in the other strand, that is, A only links to T, and C only links to G. This process of linking to only one *complementary* base is called *base-pairing*.

Gene expression is the process that allows the inherited information in the genes to direct the synthesis of functional gene products (primarily proteins). *Transcription* is the first step of this process, in which DNA works as the template for creating RNA, followed by the second step in the process called *translation*, where RNA works as the template for protein synthesis. Not all genes are expressed all the time in any particular cell type, so gene expression needs to be controlled for cells to adapt to different environments, damage, diseases, etc. The mechanisms of transcription regulation control the timing of gene expres-

sion occurrence and the amount of RNA and proteins to be synthesized. This regulation may occur at different stages of the gene expression process.

1.2 Mechanisms of transcription

A transcription factor (TF) is a function protein that aids in regulating gene expression at the transcription stage. TFs bind to specific DNA sequences to activate or inhibit the recruitment of the RNA polymerase, which is the enzyme performing RNA transcription. Transcription factor binding sites (TFBSs) are DNA sequence patterns (DNA motifs) where TFs bind. TFBSs are relatively short DNA segments (5 to 20 base-pairs) and can be located on either strand of the DNA. The binding motifs for one TF are usually highly conserved within and across species, but many TFs are capable of binding to many slight variations in specific nucleotides of the TFBS.

The majority of TFBSs occur in specific regions around the genes, usually called either *promoters* or *enhancers*. Promoters can be found in both eukaryotic and prokaryotic cells, while only eukaryotes have enhancers. Promoters are usually located upstream of genes and close to transcription start sites (TSSs). Enhancers may be located either upstream or downstream of genes and they are not necessarily close to TSSs.

Understanding the possible mechanisms of gene transcription regulation is a primary challenge for current molecular biologists, and identifying TFBSs is an important step in understanding these mechanisms. DNA motifs may be important signals of gene expression regulation in cells' response to condition changes.

Several laboratory techniques have been developed to find TFBSs, such as the electrophoretic mobility shift assay (Hellman and Fried 2007) and the DNase footprinting assay (Galas and Schmitz 1978), but they are laborious and inefficient for large scale studies. Therefore, computational approaches are necessary for efficient identification of DNA motifs given a set of sequences. Over the past decade, many computational methods have been introduced and the algorithms that they use are described in detail in Section 2.

1.3 Challenges in finding TFBSs

The highly variable nature of TFBSs presents difficulties for DNA motif searching, especially when only the direct frequency counts are used. A more sophisticated approach is to utilize a *position-specific weight matrix (PWM)*, a popular way to represent the variation of nucleotides at each position of a DNA motif. A PWM is a $4 \times n$ matrix, where the first dimension (4) represents the four possible nucleotides A, C, G and T, and the second dimension (n) is determined by the length of the motif. Each element in a PWM, denoted p_{ij}, is the frequency that the ith (row number) nucleotide occurs at the jth (column number) position in the motif. Columns in a PWM are assumed to be independent of each other, and elements in each column in the PWM should add up to one. Table 1.1 below shows the PWM of the binding motif of the TF ETS1 according to information in the database TRANSFAC (Wingender 2008).

Table 1.1: The PWM of the binding motif of the TF ETS1 according to information in the database TRANSFAC (Wingender 2008).

Position	1	2	3	4	5	6	7	8
A	0.067	0.333	0.0	0.0	1.0	0.533	0.267	0.067
C	0.933	0.600	0.0	0.0	0.0	0.133	0.067	0.400
G	0.000	0.000	1.0	1.0	0.0	0.000	0.667	0.000
T	0.000	0.067	0.0	0.0	0.0	0.333	0.000	0.533

In addition to the PWM representation, a *sequence logo* can be used to graphically represent a PWM. Figure 1.1 is the sequence logo corresponding to the TF ETS1 according to the database TRANSFAC. For a DNA motif, each position of the sequence logo corresponds to a column of the PWM. Within a position, the relative height of each nucleotide represents its frequency p_{ij} in PWM. The relative height of each position to other positions represents the importance of that position in the binding site. The R package *seqLogo* (Bembom 2007) can be used to plot the sequence logo for a given PWM.

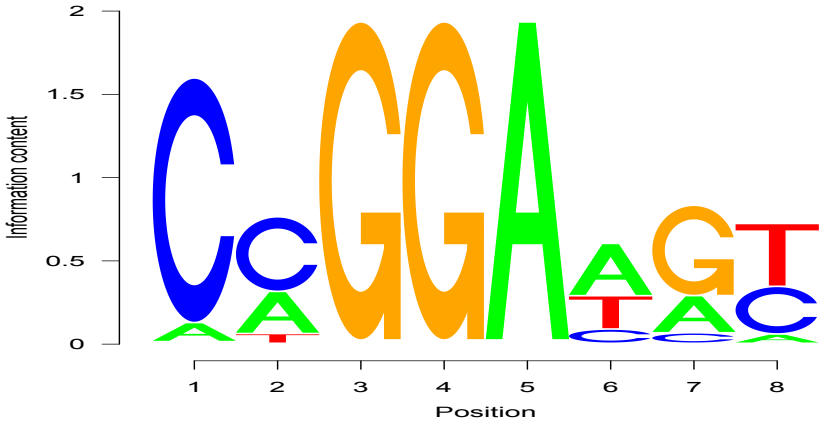

Figure 1.1: Sequence logo for ETS1 according to the TRANSFAC batabase (Wingender 2008).

Identification of DNA motifs in eukaryotes is more difficult than in prokaryotes. In eukaryotes, TFBSs are typically shorter. Motifs in enhancers can be quite variable and located far away from TSSs (up to several kilobases) (van Helden et al. 1998)

1.4 DNA packaging and transcription

Each human cell contains approximately two meters of DNA. In order for these long DNA molecules to fit in the limiting space of a cell nucleus, the DNA is tightly packaged around protein complexes called *nucleosomes*. Approximately 146 base-pairs DNA are wrapped around a histone octamer (eight proteins) to form a nucleosome. Nucleosomes are the basic repeating structural units of *chromatin*. Chromatins coil around themselves to be more condensed to form *chromosomes*. Figure 1.2 shows the basic structure of nucleosomes, in which the core indicates the wrapped DNA and the linker is the unwrapped DNA (Kornberg and Lorch 1999). One difficulty that is presented to researchers involved in DNA motif searching is to find functional occurrences of DNA motifs, because in some cases not all DNA motifs are bound by TFs *in vivo*. For example, it has been shown that functional TFBSs are usually located in nucleosome depletion regions in Yeast genome *in vivo* (Lee et al. 2004). In humans, the dynamic regulation of nucleosome positioning along the DNA

plays important roles in gene expression regulation (Schones et al. 2008). Both *in vitro* and *in vivo* packaging promoters in nucleosomes prevent the initiation of transcription (Knezetic and Luse 1986) implying that histones, in general, are gene expression inhibitors. The activation of human CD4+ T cells also induces the reorganization of nucleosomes (Schones et al. 2008), therefore, the nucleosome positioning information may be valuable for detecting functional DNA motifs in a given sequence set.

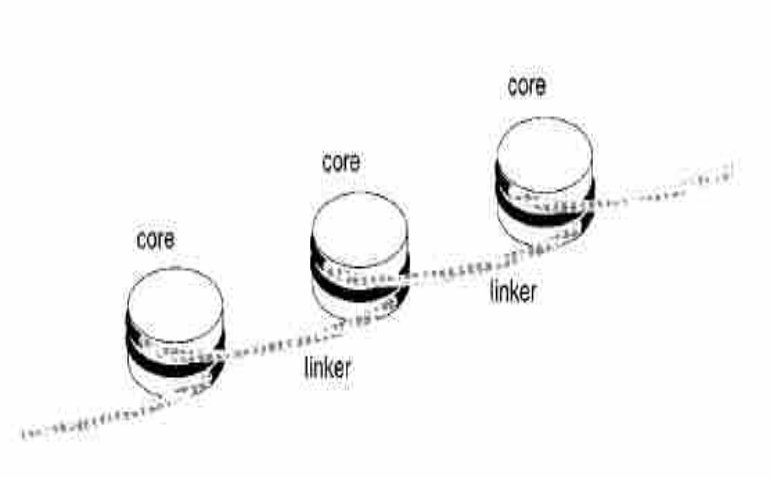

Figure 1.2: Schematic of Nucleosome Core Particle and Linker. Cores are the DNA wrapped around a histone octamer and the linkers are the unwrapped DNA (Kornberg and Lorch 1999).

1.5 Project goals

The goal is to develop an algorithm to identify the DNA motifs in a set of unaligned DNA sequences, that is to locate the starting positions of the DNA motifs in sequences and describe those motifs. We present an algorithm, XPRIME (Eliciting EXpert PRior Information for Motif Exploration), which can discover known and *de novo* (unknown) DNA motifs simultaneously from a collection of DNA sequences and describes DNA motifs using PWMs.

Previous work has defined the basic model for XPRIME (Poulsen 2009), although we will revisit the model details in the following sections. Poulsen used Gibbs sampler to estimate parameters in the model, which produced a posterior distribution of the PWMs.

However, these posteriors were fairly symmetric and the computation is time-consuming on the order of $10,000$ posterior draws.

In this report, we will describe an improved version of XPRIME over the Poulsen approach. A modified expectation-maximization (EM) algorithm is utilized in XPRIME. The observed data in the model are the given DNA sequences and the unobserved data are the locations of motifs of interest. Elements of the PWMs for motifs of interest are parameters to be updated. XPRIME uses the EM algorithm to update the parameters (PWMs) in the model using DNA sequence data (observed data) while imputing or integrating over the unknown motif locations (missing data). XPRIME improves the efficiency of locating and describing shared motifs by not allowing multiple motifs to overlap each other, a phenomenon termed a *phase shift*. Stronger motifs are expected to be generated by considering the correlations between nucleotides at different positions within each motif in XPRIME. The EM algorithm is a modified EM algorithm because some extra steps, such as phase shifting, are added between the E-step and M-step.

Moreover, a Bayesian formulation of the XPRIME algorithm allows for the elicitation of prior information for motifs of interest from literature and experiments into motif searching, which increases the efficiency of motif searching and makes the motif searching results more accurate. TRANSFAC is a database containing information of eukaryotic TFBSs (Wingender 2008). All pieces of information in TRANSFAC are obtained from available literature as well as *in vitro* experiments (Wingender et al. 1996). Because TFs may act differently *in vitro* from *in vivo*, the information from TRANSFAC of DNA motifs may be not true for motifs identified from *in vivo* environments, but this kind of information serves as the expert prior information of DNA motifs in XPRIME.

Databases, such as TRANSFAC and JASPAR (Sandelin et al. 2004), provide information that is used by most methods as expert knowledge to fix parameters in DNA motif searching. Besides these databases, more and more kinds of informative priors that improve motif detection are being identified. For example, informative priors based on structural

classes of TFs have been shown to improve the motif searching (Narlikar et al. 2006). The same group also has shown that incorporating the nucleosome occupancy information in yeast into motif discovery improves DNA motif discovery (Narlikar et al. 2007). Also, the genome-wide nucleosome positioning data for human active CD4+ T cells are becoming available (Valouev et al. 2011). Incorporating these nucleosome occupancy information may improve the performance of XPRIME.

Compared to the Poulsen version of XPRIME, in which Gibbs sampling is used to estimate parameters, the current XPRIME improves the efficiency of locating and describing shared motifs by not allowing multiple motifs to overlap each other, a phenomenon termed a *phase shift*. The improved XPRIME generates stronger motifs by considering the correlations between nucleotides at different positions within each motif. Moreover, the EM algorithm converges in 20-50 iterations and produces roughly the same results at up to a 550 fold improvement in the computational time of the Gibbs sampling procedure.

CHAPTER 2

LITERATURE REVIEW

The development and applications of algorithms for DNA motif finding have been motivated by two challenges: first, how to represent the known TF binding sites so that the representations can be efficiently used in searching for new sequences, and second, given a batch of sequences, how to identify known or *de novo* DNA motifs. In the past decade, many DNA motif finding algorithms have been developed, and these methods can be classified into two main groups based on their approaches that are used. The first group of methods consists of word-based methods that primarily perform regular word enumeration, whereas the second group of algorithms use probabilistic sequence models in which maximum-likelihood or Bayesian inference is used to estimate parameters (e.g. PWM). As discussed in the previous chapter, PWM is a popular way to represent DNA motifs in probabilistic approaches.

Word-based enumeration methods perform well when searching for short identical motifs. However, variation is common in most of DNA motifs in complex organisms. Probabilistic sequence models are more sensitive to variation due to their PWM parameterization, and can also improve performance when searching for longer motifs (Das and Dai 2007).

2.1 Word-Based algorithms

van Helden et al. presented a simple and fast word-based method for identifying TFBSs within a list of coregulated genes (van Helden et al. 1998). This method is based on detecting over-represented oligonucleotides in the given coregulated sequences. At first, the expected oligonucleotide frequency $F_{nc}b$ for each possible oligonucleotide (b) is observed through all non-coding segments in the genome, e.g. 800bp upstream regions in the yeast genome. Then the oligonucleotide-specific expected frequencies (F_eb) are estimated by $F_eb = F_{nc}b$. If the coregulated sequence set contains s sequences and the sequence lengths are denoted by L_i,

for i from 1 to s, the total number of the possible occurrences of each oligonucleotide with length w is $T = 2 \times \sum_{i=1}^{s} (L_i - w + 1)$. The constant 2 indicates that the occurrences are counted in both DNA strands. The number of expected occurrences of each oligonucleotide is calculated by multiplying the oligonucleotide-specific expected frequency with the total number of possible occurrences, which is

$$E(occ[b]) = F_e b \times 2 \times \sum_{i=1}^{s} (L_i - w + 1) = F_e b \times T \tag{2.1}$$

The binomial formula is used to calculate the possibility that each oligonucleotide is observed to occur no less than n times in the given coregulated sequence set:

$$P(occ[b] \geq n) = \sum_{j=n}^{T} P(occ[b] = j) = \sum_{j=n}^{T} \binom{T}{j} \times (F_e b)^j \times (1 - F_e b)^{(T-j)} \tag{2.2}$$

The significance coefficient, $sig = -log_{10}[P(occ[b] \geq n) \times D]$, is used to detect the true over-represented oligonucleotides. $\frac{1}{D}$ is a threshold chosen depending on the length of the oligonucleotides. For example, the criterion, $sig \geq 0$, helps detect every oligonucleotide with possibility that its occurrences are no less than n times in the coregulated sequences is lower than $\frac{1}{D}$.

This method requires calibration of the uneven oligonucleotide representation in the genome with a set of reference sequences, for example, all of the non-coding regions in the genome. It is efficient in identifying the known and unknown motifs that are over-represented in a set of coregulated sequences. However, it has shortcomings: its range of detection is limited to relatively short motifs, no variations are allowed within an oligonucleotide, and it is difficult to detect the spaced dyad motifs. The last shortcoming was overcome by van Helden et al. in the improved version of the algorithm (van Helden et al. 2000).

With development of advanced sequencing technologies, genome-wide mRNA expression data for several organisms are becoming available. In 2000, Bussemaker et al. proposed a Probabilistic Segmentation Model for detecting TFBSs (Bussemaker et al. 2000). Compared to the model developed by van Helden et al. in 1998, this model does not require any separate set of reference data to define probabilities, and it considers DNA sequences as an

unknown language with four letters (A, C, G and T). As a dictionary-based sequence model, it decomposes DNA sequence into the most probable "dictionary" of motifs or words. The words are oligonucleotides with various lengths and each word α has an associated possibility p_α. These probabilities are normalized and the sum of them is equal to one. Given a DNA sequence S, building a dictionary from S starts from detecting the frequencies of individual letters and over-represented pairs. Each possible pair (α, β) can be tested for over-representation using a Z-score.

$$Z_{\alpha\beta} = \frac{\langle N_{\alpha\beta} \rangle - N_{av} p_\alpha p_\beta}{\sqrt{N_{av} p_\alpha p_\beta}}, \tag{2.3}$$

where $\langle N_{\alpha\beta} \rangle$ is the predicted value of $N_{av} p_\alpha p_\beta$ in the model, and $N_{av} = L/\langle l \rangle$ is the average number of words in a partition, with $\langle l \rangle = \sum_\alpha l_\alpha p_\alpha$. Pairs with Z-scores above a specific threshold will be added to the dictionary, after which their associated probabilities are calculated using the maximum-likelihood procedure. The longer fragments will be tested and added in the same way. That is to say, a dictionary will be built by beginning with the four bases and ending when no pairs with Z-scores above the threshold can be found.

The word-based algorithms developed by Tompa in 1999 addressed the problem that no variations are allowed within an oligonucleotide in previous word-based algorithms (Tompa 1999). This approach considers the absolute number of occurrences of the motif and the distribution of the background genome. Later in 2000, Shiha and Tompa incorporated the transition matrix for an order m Markov chain that is constructed from the entire sequence set by assuming that the occurrences of a motif are not independent, but depend on previous occurrences (Sinha and Tompa 2000). This model also uses a Z-score to measure the statistical significance for each motif. Given a set of random DNA sequences X and a motif s, let the random variable be the number of occurrences of the motif s in X and let $E(X_s)$ and $\sigma(X_s)$ be its mean and standard deviation. The Z-score of s is represented as

$$Z_s = \frac{N_s - E(X_s)}{\sigma(X_s)}. \tag{2.4}$$

2.2 PWM UPDATING METHODS

Lawrence and Reilly applied the expectation maximization (EM) method on the identification of protein motifs, which can also be applied for DNA motifs (Lawrence and Reilly 1990). The unknown locations of the motif in a given set of sequences are treated as the missing data by the EM algorithm. p_{ik} represents the probability that the shared motif starts at position k in sequence i, given the input of a set of N unaligned sequences and the width of the shared motif, W. f_{mj} refers to the probability that the nucleotide is m ($m \in M = \{A, C, G, T\}$) at position j in the shared motif ($1 \leq j \leq W$) and is the element in row m and column j of the PWM matrix for the motif ($m \in M = \{A, C, G, T\}$ is corresponding to row $1, 2, 3, 4$). Let L be the length of sequences (all sequences are assumed to be of the same length) and q_m be the frequency of the nucleotide m at all positions of the sequences other than the regions of the motif. The log of the likelihood function of the model given the sequences is

$$log(likelihood) = N \sum_{j=1}^{W} \sum_{m \in M} f_{mj} log(f_{mj}) + N(L - J) \sum_{m \in M} q_m log(q_m). \tag{2.5}$$

The EM algorithm starts with an estimate of f generated randomly or specified by the user, and then alternatively estimates f and p_{ik} until f changes very little from iteration to iteration. In the E step, the expectation of the log of the likelihood and the distribution of p_{ik} are estimated given the current estimates of f, and in the M step, f is estimated by maximizing the expectation of the log of the likelihood.

This algorithm has several limitations: how to choose a starting value for f and p_{ik} is not demonstrated clearly, the assumption that each sequence contains exactly one motif may not be appropriate for all of the sequences and may bring inaccuracy to the characterization of the motif. Furthermore, only one shared motif can be found each time.

The algorithm Multiple EM for Motif Elicitation (MEME), developed by Bailey and Elkan, is an extension of the EM algorithm (Bailey and Elkan 1993). MEME has overcome the limitations of the original EM algorithm. First, the EM algorithm is not guaranteed

to converge in the global maximum because of the random choice of the starting point. To solve this problem, MEME uses all subsequences of the given motif length W in the input sequences as starting points to make sure that the actual occurrences of the shared motif are always used as a staring point and make the EM converge to the global optimum. Second, MEME allows each sequence to have zero, one or several appearances of the shared motif by letting the user set the number of occurrences of motifs. Third, by probabilistically erasing appearances of a motif after they are found and continuing the searching for another shared motif, MEME is able to find more than one shared motif each time.

Both EM and MEME are two component mixture (TCM) models. Let N be the number of sequences in the given sequence set, L be the length of each sequence (all sequences are assumed to be of the same length), W be the width of the motif, $X_i = \{X_{i,k}\}_{k=1}^{L}$ represent all nucleotides in sequence i, and $X_{i,k} \in \{A, C, G, T\}$ denote the nucleotide at position k of sequence i. The observed data are represented by N i.i.d. random variables $\{X_1, \ldots, X_N\}$. Each position in a motif of interest (or, equivalently, each column in the PWM) is assumed to have an independent multinomial distribution. $Z_{i,j}$ is an indicator variable to indicate if the motif starts at the jth position in the sequence X_i. λ is the probability of $Z_{i,j} = 1$. θ_0 represents the parameters of the motif model, and θ_1 represents the parameters of the background model. $\tilde{X}_{i,j}$ represents the subsequence of width W starting at position j in sequence X_i. TCM models assume that all $\tilde{X}_{i,j}$ are independent with each other. Although the overlapping $\tilde{X}_{i,j}$ do not seem to follow this independent assumption, Bembom et al. showed that the results with and without this assumption in MEME are comparable (Bembom et al. 2007). Under the independent assumption, the likelihood of the subsequence $\tilde{X}_{i,j}$ conditional on the variable $Z_{i,j}$ is given by

$$Pr(\tilde{X}_{i,j}|Z_{i,j} = 1, \theta_1) = \prod_{k=1}^{W}\prod_{j=1}^{4} \theta_{kj}^{I(Xi,j+k-1=j)} \tag{2.6}$$

$$Pr(\tilde{X}_{i,j}|Z_{i,j} = 0, \theta_0) = \prod_{k=1}^{W}\prod_{j=1}^{4} \theta_{0j}^{I(Xi,j+k-1=j)}. \tag{2.7}$$

In 1994, Bailey and Elkan introduced an advanced version of MEME (MEME+), which is also a TCM model and uses the EM algorithm to estimate the number of occurrences of motifs and update the parameters for the model (Bailey and Elkan 1994). The MEME+ model has two components: one component is for the motif and the other component is for the background noise. Users do not need to know in advance how many times the motif occurs in the sequences in MEME+. Let $X = (X_1, X_2, \ldots, X_n)$ denote the input set of n sequences, where each sequence has length L. W refers the width of the motif of interest, and $m = L - W + 1$ is the number of possible starting positions for a motif with length W in each sequence. $Z_{i,j}$ is an indicator variable to indicate whether or not the motif starts at the jth position in the sequence X_i. Let λ denote the probability of $Z_{i,j} = 1$. θ_0 represents the PWM of the motif of interest, θ_1 represents the background PWM. The log of the joint likelihood for the model in MEME+ is given below,

$$\begin{aligned} Log(likelihood) &= LogPr(X, Z|\theta, \lambda) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}[(1 - Z_{i,j})LogPr(X_{i,j}|\theta_0) \\ &+ Z_{i,j}LogPr(X_{i,j}|\theta_1) + (1 - Z_{i,j})Log(1 - \lambda) \\ &+ Z_{i,j}Log\lambda]. \end{aligned} \tag{2.8}$$

The input required by MEME+ consists of a set of sequences (X) and a number specifying the width of the motifs of interest (W). MEME+ returns a model of each motif and a threshold t. t is calculated as $log(\frac{1-\lambda}{\lambda})$ for each motif. t and the model for each found motif can be used as a Bayes-optimal classifier for searching for occurrences of the motif in other sets of sequences.

With the recent technological innovation, CHIP-chip (chromatin immunoprecipitation coupled with microarray analysis), researchers are able to identify regions of a given genome that contain specific TFBSs. These results are valuable for detecting DNA motifs for transcription factors. DNA regions with high CHIP-chip scores are more likely to contain the motifs of interest. Based on the TCM models, Shim and Keles in 2008 introduced a condi-

tional TCM model, called CTCM, by incorporating the CHIP-chip information in the DNA motif searching (Shim and Keles 2008). Let $T = (T_1, \ldots, T_N)$, where $T_i = (T_{i,1}, \ldots, T_{i,L})$, denote the CHIP-chip scores for each base pair in each sequence of the input sequence set. They assume that $\tilde{X}_{i,j} \perp T_{i,j} | Z_{i,j}$, that is, the sequence data are independent of the CHIP-chip data conditional on the motif occurrence and location random variables. The model is written as

$$Pr(\tilde{X}_{i,j}|T_{i,j}, \Theta) = \sum_{z=0}^{1} Pr(\tilde{X}_{i,j}|Z_{i,j} = z, \theta_0, \theta_1) Pr(Z_{i,j} = z|T_{i,j}, \theta_f), \tag{2.9}$$

where θ_f is the parameters of the conditional distribution of Z given T. Three alternative models are considered to model the $Pr(Z_{i,j} = z|T_{i,j}, \theta_f)$: the beta prior on λ, the logistic regression model, and the piecewise constant model.

One EM approach to align a set of DNA sequences for detecting the shared DNA motif is in the form of hidden Markov Models (HMM) (Baldi et al. 1994). This approach allows a gap between any two nucleotides in DNA sequences. A set of N DNA sequences can be seen as a set of different utterances of the same word that are generated by a common underlying HMM with a left-right architecture (motif). This HMM model is defined by a set of states S (main state, delete state and insert state), an alphabet of four letters (A, C, G, T), a probability transition matrix $T = (t_{ij})$ and a probability emission matrix $E = (e_{i\alpha})$. When a system is in state i, it has a probability t_{ij} to move to state j and a probability $e_{i\alpha}$ to emit symbol α. In the case of DNA motif searching, the main and insert states always emit a letter of the alphabet, while the delete states are mute. The linear main states make up the backbone of the HMM model. This approach needs a set of training sequences to modify the parameters iteratively using the product of the likelihood of the sequences. For each sequence, the corresponding most probable path is computed through iterations. Aligning these paths using the maximum likelihood estimator identifies new motifs.

Another important probabilistic method Gibbs sampling, a Markov Chain Monte Carlo (MCMC) approach, has been used in several motif searching algorithms. As in the EM method, Gibbs sampling at each step only depend on the results of the previous step.

Also, in contrast to the selection of next step in the EM method, the way to select the next step in Gibbs sampling is based on random sampling and it is not deterministic (Das and Dai 2007).

In 1993, Lawerence et al. developed a new algorithm for local multiple alignment to search for shared motifs in multiple protein or nucleic acids sequences (Lawrence et al. 1993). This algorithm utilizes a Gibbs sampling strategy and does not require any prior information on the motifs of interest. It is called the "site sampler" for its assumption that every sequence at least contains one instance of a motif (Das and Dai 2007). Gibbs sampler is able to search for several motifs simultaneously. Following is an example of searching for one motif at one time. Given a set of N sequences $S_1, S_2, \ldots, S_N$, this algorithm finds an optimized local alignment model for these N sequences in N-linear time. Within each sequence, the algorithm searches for mutually similar segments of specified width. Two data structures are used: one uses the PWM to describe the motif, where the element in PWM $q_{i,j}$ (i is from 1 to W and j is A, C, G, T) represents the frequency of nucleotide j at position i. Also, "background frequencies" p_j (j is A, C, G, T) represent the analogous probability that the nucleotide j occurs in the site where the motif does not occur. The objective is to identify the most probable motif, with which the alignment is located to maximize the ratio of the corresponding motif probability to background probability.

This algorithm starts with randomly choosing a starting sequence out of the N sequences (assuming that sequence z is chosen) and then iteratively proceeds. Each iteration consists of two steps. First, a predictive update step is conducted, in which the $q_{i,j}$ in PWM and "background frequencies" p_j are calculated based on all sequences excluding z. Second, a sampling step is conducted where every possible subsequence x with width W in sequence z is considered as a possible motif. For each x, let Q_x denote the probability to generate x based on $q_{i,j}$ from last step and let P_x denote the probability to generate x based on the p_j from last step. The weight $A_x = Q_x/P_x$ is assigned to each x and within these weighted segments, another position a_z is then randomly chosen. Basically, this algorithm selects a

set of positions a_k, for k from 1 to N, in the set of sequences such that the product of the corresponding set of A_x is maximized. Equivalently, the sum of the log of A_x is maximized, which is denoted as F. F is given by

$$F = \sum_{i=1}^{W} \sum_{j=A,C,G,T} c_{i,j} log \frac{q_{i,j}}{p_j}, \tag{2.10}$$

where $c_{i,j}$ represents the count of nucleotide j at each position i in the $N-1$ sequences (all sequences excluding the sequence z). One defect of this algorithm may be that it may fall into a non-optimal local maximization when randomly chosen positions happen to be near each other in different iterations. In order to solve this problem, a "phase shift" step may be inserted after every maximization step. Moreover, instead of requiring the input of a specific motif width, a superior criterion which is based on the incomplete-data log-probability ratio may be used to choose motif width.

The statistical background of the Gibbs sampling strategy in motif searching has been well presented by Liu et al. in 1995. Given a set of N DNA sequences with length L, the goal is to identify the most probable motif with width W and the alignment pattern. Firstly, it is assumed that each sequence contains a single copy of the motif. The motif segments with length W from input sequence are assumed to be independent observations from a product-multinomial model (called a motif). This model describes nucleotide frequencies for each position j within the motif and consists of $4 \times W$ parameters denoted by $\Theta = \{\theta_{i,j}\}$, for each i representing A, C, G, T and j from 1 to W. The background parameters $\theta_{i,0}$ describe the nucleotide frequencies in the non-motif region.

Let random vectors $R_n = (r_{n1}, \ldots, r_{nW})$, for n =1, ..., N, where the r_{nj} is the corresponding random variable for the observation at the jth position within the motif region in the nth sequence. The likelihood function can then be written as

$$\pi(R_1, \cdots, R_N|\Theta) \propto \prod_{j=1}^{W} \theta_j^{h(R_j)}, \tag{2.11}$$

where $h(R_j)$ is the sufficient statistic of θ_j for all j. It is known that the conjugate prior for a product multinomial distribution is a product Dirichlet distribution. Thus, if the prior

distribution of Θ is a product Dirichlet distribution PD(B), the posterior distribution of Θ will be $PD(B + H)$, where $H = (h(R_{\cdot 1}, \ldots, R_{\cdot W}))$.

In order to identify the alignment pattens (i.e., the positions of motif copies in each input sequence), the starting positions for the motif copies are incorporated as missing data in the model. Let $\{A\} = \{(n, a_n + j - 1) : n = 1, \ldots, N, j = 1, \ldots, W\}$ denote the set of indices occupies by the copies of the motif in input sequences and let $\{A\}^c$ denote the set of indices not occupied by the copies of the motif in the input sequences. The complete-data likelihood can be written as

$$\pi(R_1, \cdots, R_N, A|\Theta) \propto \theta_0^{h(R\{A\}^c)} \prod_{j=1}^{W} \theta_j^{h(R_{A(j)})}. \tag{2.12}$$

Based on this model, the parameter vectors θ_0 and θ are integrated out to obtain a predictive update under the Gibbs sampler. Some modifications can be made for allowing multiple copies of a motif within each input sequence and for searching for multiple motifs in each run of the algorithm.

By choosing the Gibbs sampling strategy as the starting point, Roth et al. developed an algorithm to discover the recurring motifs in unaligned sequences, called "AlignACE", short for *Aligns Nucleic Acid Conserved Elements* (Roth et al. 1998). A motif is defined as the characteristic base-frequency patterns of the most information-rich columns of a set of aligned sites in this algorithm. A series of motifs that are overrepresented in the input set of DNA sequences is returned in the form of weight matrices. An alignment score for each resulting motif, which is a measure of "goodness" of sequence alignment, is then calculated using Berg and von Hippel. A threshold is set for this score and motifs with scores exceeding the threshold are considered. In order to measure the fraction of ORFs (Open Reading Frames) in the genome with matching upstream sites, an occurrence score is calculated for each resulting motif. Motifs with a score lower than 1% are selected. This criterion ensures that motifs occur infrequently among upstream regions.

Compared with the Gibbs sampling strategy, AlignACE is different in several aspects (Das and Dai 2007). First, AlignACE uses the fixed base frequencies for background re-

gions according to the source genome. Second, AlignACE searches for motifs in both DNA strands. Third, In AlignACE, multiple motifs are found one by one by masking iteratively, not simultaneously. Finally, MAP (maximum a priori log-likelihood) is used in AlignACE to measure the degree of the over-represented motifs. The shortcoming of MAP is that it cannot distinguish between the true over-represented motifs from some motifs occurring ubiquitously in a genome, such as A-rich motifs in yeast.

BioProspector is a DNA motif searching method also using the Gibbs sampling strategy and makes several improvements (Liu et al. 2001). Since in some cases in DNA, a particular nucleotide may affect the presence of the nucleotide in its neighboring positions, BioProspector uses zero to third-order Markov background models to score segments. For example, from a third-order Markov background model, the probability of generating segment ATGTA is calculated as:

$$\begin{aligned} P^3_{ATCTA} = P(A) \times P(T|\text{previous base is } A)P(G|\text{previous bases are } AT)\times \\ P(T|\text{previous 3 bases are } ATG) \times P(A|\text{previous 2 bases are } TGT) \end{aligned} \tag{2.13}$$

Parameters of the third-order Markov background model are either given by the user or estimated from a specific sequence file. Moreover, BioProspector considers the cases in which simultaneous and proximal binding of two transcription factors or binding of a homodimer may be required in the transcription initiation. Two probability matrices may be used to capture two blocks with their gap range specified. In this way, BioProspector is able to search for spaced dyad motifs and palindromic motifs. To score a motif, BioProspector uses the following formula:

$$MotifScore = \#seg \times exp[\sum_{\text{all positions}} \sum_{\text{all nucleotides}} q_{i,j} \times log(q_{i,j}/p_j)]/W, \tag{2.14}$$

where the definitions of $q_{i,j}, p_j$ and W are the same as those in the original Gibbs sampling strategy above. The statistical significance of the motif score is estimated by Monte Carlo simulations.

The original Gibbs sampling strategy assumes that each sequence in the input set of sequences contains a single copy of the motif and only allows for sequences to contain

more than zero copies or multiple copies of the motif. BioProspector solves this problem by using two score thresholds, a high threshold T_H and a low threshold T_L, when sampling new alignments. All the non-overlapping segments of a sequence with scores higher than T_H are added to the motif, and all the non-overlapping segments of a sequence with scores lower than T_L are removed. Segments with scores between T_L and T_H are chosen with probability proportional to $A_x - T_L$, where $A_x = Q_x/P_x$, Q_x and P_x are defined as in the original Gibbs sampling strategy. This criterion can help the program converge more quickly.

Motif Discovery Scan (MDscan) is another computational method developed by Liu et al. to search for DNA motifs (Liu et al. 2002). MDscan combines the advantages of word enumeration and PWM updating approaches and incorporates the chromatin immunoprecipitation array (CHIP-array) ranking information to make searches faster and increase the success rates. CHIP-array experiments can help select probable protein-DNA interaction loci within 1-2 kilobase resolution. Each TF binding DNA fragment selected by CHIP-array is enriched in the experiment, and the ones having high CHIP-array enrichment are more likely to represent multiple DNA motifs. Given a set of DNA sequences selected by CHIP-array experiments, they are listed by their enrichment from highest to lowest. The goal is to find DNA motifs of width W. At first, MDscan uses a word-enumeration strategy to search for oligomers of width W (W-mers) that are abundant in the top sequences (top 3-20 sequences) with at least m base pairs matching the candidate motif, where m is specified by the user. If this step can generate a reasonable range of the total number of motif elements, MDscan then will use a Bayesian scoring function to evaluate and refine the PWM:

$$\frac{x_m}{W} \times \Big[\sum_{i=1}^{W} \sum_{j=A}^{T} p_{i,j} Log(p_{ij}) - \frac{1}{x_m} \sum_{\text{all segments}} Log(p_0(s)) - Log(\frac{\text{expected based}}{\text{site}}) \Big], \qquad (2.15)$$

where x_m is the number of m-matches aligned in the motif. p_{ij} is the frequency of nucleotide j at position i of the PWM, and $p_0(s)$ is the probability of generating the m-matches from the third-order Markov background model. If the expected number of sites in the top sequences

is not known, the scoring function will be

$$\frac{Log(x_m)}{W} \times \big[\sum_{i=1}^{W}\sum_{j=A}^{T} p_{i,j} Log(p_{ij}) - \frac{1}{x_m} \sum_{\text{all segments}} Log(p_0(s))\big]. \tag{2.16}$$

The top 10-50 candidate motifs with the highest scores are used by MDscan to update the PWM iteratively. Compared to the original word enumeration approaches, MDscan brings flexibility of base substitutions in PWM. MDscan also performs better than several PWM updating approaches when dealing with large datasets by increasing the running speed and avoiding serious local-maximum problems.

MDscan is based on the assumption that in response to a given biological condition, the effect of a TFBS is strongest among genes with the most dramatic increase or decrease in mRNA expression. An alternative approach to discover DNA motifs named Motif Regressor is also based on this assumption (Conlon et al. 2003). Motif Regressor uses MDscan to find motifs from the most induced and repressed genes, and then verifies each candidate motif by associating each gene's upstream sequence motif-matching score with its downstream expression measure. Let S_{mg} denote the score for the matching of the upstream sequence of gene g and a motif m and let Y_g denote the log-expression value of gene g. For each motif reported from MDscan, a simple linear regression model is fit:

$$Y_g = \alpha + \beta_m S_{mg} + \epsilon_g, \tag{2.17}$$

where ϵ_g is the error term for gene g. A non-zero β_m suggests that motif m is correlated with the gene expression of gene g. Motif candidates with an insignificant β_m are then removed. For remaining motif candidates, each β_m is retained and used to fit a multiple regression model performed by the stepwise regression:

$$Y_g = \alpha + \sum_{m=1}^{M} \beta_m S_{mg} + \epsilon_g. \tag{2.18}$$

The stepwise regression starts with the intercept term, at each step adds the motif that gives the largest reduction in residual error and ends when no motif can be added with a significant coefficient.

Several algorithms have been developed to identify the cis-regulatory modules (CRMs) based on the existing DNA motif searching algorithms. CRMs refer to a combination of transcription factors with corresponding binding sites forming homotypic or heterotypic clusters. Most eukaryotic genes are regulated by CRMs. EMCMODULE is a method for inferring the CRM responsible for a set of co-regulated genes by using a Hidden Markov Model(Gupta and Liu 2005). EMCMODULE starts with a set of motifs found by existing algorithms and databases and then selects motifs that are like members of CRMs and updates the parameters iteratively. By assuming that there are K motifs in the module of interest, EMCMODULE models the dependencies of these motifs on each other by a $K \times K$ transition matrix, V. A truncated geometric distribution, d_{ij}, is used to model the distance between the K motifs. A hidden Markov model is used to represent the CRM of interest. The evolutionary Monte Carlo (EMC) method is employed to screen the motif candidates and a forward-backward recursion method is performed to locate the motifs.

CHAPTER 3

METHODS

3.1 Previously developed model

We assume that we are given a set of S DNA sequences. Let X_s denote the sth sequence, for s from 1 to S and let L_s denote the length of the sth sequence. $X_n = \{X_{n,s}\}_{n=1}^{L_s}$ represents all nucleotides in sequence s and $X_{n,s} \in \{A, C, G, T\}$ denotes the nucleotide at position n of sequence s. The observed data are i.i.d random variables $X_1, \ldots, X_S$. The number of motifs of interest m and the width of motifs W are specified by the user. It is assumed that all motifs are of the same length.

Motifs are represented by PWMs. Recall that a PWM is a $4 \times W$ matrix. Every element in PWM p_{ij} is the frequency that jth (column number) position in the motif is the ith (row number) nucleotide. Columns in PWM are assumed to be independent to each other, and the elements in each column should sum to one. Following Lawerence et al. (Lawrence et al. 1993), we assume that each position in a motif has a multinomial distribution with $k = 4$. Assuming that the positions on the PWM are independent of each other, the joint distribution of the positions follow a product multinomial distribution. The remaining positions or sequences not occupied by motifs called *background* sequences. The background can also be seen as a randomly generated motif with length W.

For each segment of length W, called a W-mer, the likelihood score is written as a product multinomial distribution. Let $X_{n,s}$ denote the subsequence of width W starting at position n in sequence X_s, where n is from 1 to $L_s - W + 1$. Let Δ_d, for d from 1 to $m + 1$, be a indicator variable, where $\Delta_d = 1$ denotes that the W-mer belongs to the dth motif and $\Delta_d = 0$ denotes that the W-mer does not belong to the dth motif. Each W-mer can only belong to one motif, that is, only one Δ can be equal to one.

The likelihood score for the W-mer $X_{n,s}$ given $X_{n,s}$ belongs to the dth motif can be written as

$$f_d(X_{n,s}) = f(X_{n,s}|\Delta_d = 1)_{score} = \prod_{i=1}^{W} \prod_{j\in\{A,C,G,T\}} p_{dij}^{I(X_{(n+i-1),s}=j)}, \tag{3.1}$$

where p_{dij} represents the (i, j) element of the PWM for the dth motif.

For the likelihood function of all W-mers in the given DNA sequences $\{X_{n,s}\}$, the parameters can be represented by

$$\Theta = ([p_{dij}], r_1, \ldots, r_{m+1}), \tag{3.2}$$

where $[p_{dij}]$ represents the PWM for the motif d and r_d, for d from 1 to $m + 1$, represents the ratio of the number of W-mers that belong to motif d to the total number of all W-mers, $\sum_{i=1}^{m+1} r_i = 1$. W-mers in the data set may have overlaps. For example, one W-mer and the preceding W-mer have $W - 1$ nucleotides overlapping. We may assume all W-mer are independent to each other, because Bembom et al. showed that the motif searching results in MEME (Bailey and Elkan 1994) with and without the independence assumption are comparable (Bembom et al. 2007).

All W-mers in the given DNA sequences $\{X_{n,s}\}$ are the observed data while the unknown motif locations $\{\Delta_{d,n,s}\}$ are the unobserved hidden data. Let $z = (\{X_{n,s}\}, \{\Delta_{d,n,s}\})$ represent the complete data. With the assumption that all W-mer are independent to each other, the complete data likelihood can be written as the products of the likelihood score for all W-mer:

$$L(\Theta|z) = Pr(z|\Theta) = \prod_{s=1}^{S} \prod_{n=1}^{L_s+W-1} \prod_{d=1}^{m+1} C(X_{n,s})[r_d f_d(X_{n,s})]^{\Delta_{d,n,s}}. \tag{3.3}$$

Priors

For a class of likelihood functions $\pi(x|\Theta)$, if the prior distribution $\pi(\Theta)$ is in the same family as the posterior distribution $\pi(\Theta|x)$, the prior distribution $\pi(\Theta)$ is the conjugate prior for the likelihood function family. The Dirichlet distribution, which is an extension

of the beta distribution, is the conjugate prior for the multinomial distribution, and follows that the product Dirichlet distribution is the conjugate prior for the product multinomial distribution. According to Bayes Theorem, the posterior distribution, $\pi(\Theta|x)$, is proportional to the product of the prior distribution and the the likelihood of the data. More specifically,

$$\pi(\Theta|x) \propto \pi(\Theta)\pi(x|\Theta). \tag{3.4}$$

As (3.1) shows, the $f_d(X_{n,s})$ (d is from 1 to m) is the product multinomial distribution, so using a product Dirichlet distribution as the prior distribution for $f_d(X_{n,s})$ (d is from 1 to m) results in a product Dirichlet posterior distribution.

In order to search for *de novo* motifs in the given sequences, a PWM prior will be generated from a random product Dirichlet distribution. When searching for motifs with known PWMs, the PWM priors incorporate expert information of the motifs, (e.g., information from the Databases, such as TRANSFAC and JASPAR (Sandelin et al. 2004)).

The parameters $\{r_i\}$, for i from 1 to $m+1$, represent the proportions of the numbers of W-mers DNA that belong to each motif to the total number of all W-mers in the data set, and $\sum_{i=1}^{m+1} r_i = 1$. By this definition, the distribution of $\{r_i\}$ can be a multinomial distribution. Therefore, a Dirichlet prior for $\{r_i\}$ is chosen for the multinomial distribution. The parameters for the Dirichlet prior are chosen by the user. If the user has expert knowledge about the parameters, the parameters can be set to reflect the information. By default, the parameters are set such that $r_1 = r_2 = \ldots = r_{m+1}$.

Gibbs sampling

Gibbs sampling was used by Poulsen to update all of the parameters in the model (Poulsen 2009). We call this program XPRIME-GIBBS in this project. The complete posterior distribution are proportional to the product of the complete likelihood function and the priors:

$$\pi(\Theta|x) \propto L(\Theta|z)\pi(r) \prod_{d=1}^{m+1} \pi(\{p_{dij}\}) \tag{3.5}$$

The complete conditional posterior distribution of the PWM for the dth motif can be expressed as

$$[\{p_{dij}\}|\Delta_{d,n,s}] = \prod_{s=1}^{S} \prod_{n=1}^{L_s+W-1} [f_d(X_{n,s})]^{\Delta_{d,n,s}} \pi(\{p_{dij}\}), \tag{3.6}$$

where if the prior $\pi(\{p_{dij}\})$ follows the product Dirichlet distribution with parameters $\{\alpha_{Dij}\}$, the complete conditional posterior $[\{p_{dij}\}|\Delta_{d,n,s}]$ will the product Dirichlet distribution with parameters $\sum_{s=1}^{S} \sum_{n=1}^{L_s+W-1} \Delta_{d,n,s} I(X_{(n+i-1),s} = j) + \{\alpha_{Dij}\}$.

The complete conditional posterior distribution of r can be expressed as

$$[r|\{\Delta_{d,n,s}\}] = \prod_{s=1}^{S} \prod_{n=1}^{L_s+W-1} \prod_{d=1}^{m+1} [r_d f_d(X_{n,s})]^{\Delta_{d,n,s}} \pi(r), \tag{3.7}$$

where if the prior $\pi(r)$ follows the product Dirichlet distribution with parameters $\{\alpha_d\}$, the complete conditional posterior $[r|\{\Delta_{d,n,s}\}]$ are the product Dirichlet distribution with parameters $\sum_{s=1}^{S} \sum_{n=1}^{L_s+W-1} \Delta_{d,n,s} + \{\alpha_d\}$.

Notice that the complete conditional posterior distributions depend on the unknown motif locations $\{\Delta_{d,n,s}\}$. $\{\Delta_{d,n,s}\}$ are assumed to follow the multinomial distribution with parameters $p_\Delta \propto \{r_i\} \times f(X_{n,s})$. The Gibbs sampling procedure has the followings steps:
(1) Draw Δs from a multinomial distribution with parameters $p_\Delta \propto \{r_i\} \times f(X_{n,s})$.

(2) Draw r from a Dirichlet distribution with parameters $\sum_{s=1}^{S} \sum_{n=1}^{L_s+W-1} \Delta_{d,n,s} + \{\alpha_d\}$

(3) Draw $\{p_{dij}\}$ from a Dirichlet distribution with parameters $\sum_{s=1}^{S} \sum_{n=1}^{L_s+W-1} \Delta_{d,n,s} I(X_{(n+i-1),s} = j) + \{\alpha_{Dij}\}$

(4) Repeat 1 through 3 steps for N iterations, where N denotes the number of iterations set by the user.

3.2 EM ALGORITHM

The expectation maximization (EM) algorithm is a general method for updating the parameters of a model when there are missing values for unobserved random variables. In this report, the EM algorithm is used to update the parameters (PWM) in the model using all W-mers in the given DNA sequences $\{X_{n,s}\}$ (observed data) while imputing or integrating over the unknown motif locations $\{\Delta_{d,n,s}\}$ (unobserved hidden data). $z = (\{X_{n,s}\}, \{\Delta_{d,n,s}\})$ is the complete data. The formal optimization problem is, given a fixed $\{X_{n,s}\}$, to find the value of Θ which maximizes the marginal probability

$$Pr(\{X_{n,s}\}|\Theta) = \sum_{\Delta_{d,n,s}} Pr(\{X_{n,s}\}, \{\Delta_{d,n,s}\}|\Theta) = \sum_{\Delta_{d,n,s}} Pr(z|\Theta) \tag{3.8}$$

The EM algorithm maximizes $Pr(\{X_{n,s}\}|\Theta)$ with respect to Θ by iteratively computing a sequence $\{\Theta_p\}$ which converges to a optimal value $\hat{\Theta}$. In the pth iteration, in the expectation E step, the expected value of $\{\Delta_{d,n,s}\}$ given $\{X_{n,s}\}$ and $\{\Theta_p\}$ is calculated:

$$\{\hat{\Delta}_{d,n,s}\} = E(\{\Delta_{d,n,s}\}|\{X_{n,s}\}, \{\Theta_p\}) \tag{3.9}$$

Take a single W-mer $X_{n,s}$ as an example,

$$\begin{aligned} \hat{\Delta}_{d,n,s} &= E(\Delta_{d,n,s}|X_{n,s}, \Theta_p) \\ &= 1 \times Pr(\Delta_{d,n,s} = 1|X_{n,s}, \Theta_p) \end{aligned} \tag{3.10}$$

Then in the maximization M step, the $\{\hat{\Delta}_{d,n,s}\}$ is used to improve the estimate of Θ, and Θ_{p+1} is chosen to maximize the joint probability of $\{X_{n,s}\}$ and $\{\hat{\Delta}_{d,n,s}\}$:

$$\{\Theta|Pr(\{X_{n,s}\}, \{\hat{\Delta}_{d,n,s}\}|\Theta)\text{is maximized}\} \tag{3.11}$$

This step is to maximize the posterior distribution of $f_d(X_{n,s})$ (d is from 1 to m), which is a product Dirichlet distribution.

The EM algorithm in XPRIME has the following steps:

(1)In the pth iteration, for $p = 1$, $\Theta^{(1)} = ([p_{dij}]^{(1)}, r_1^{(1)}, \ldots, r_{m+1}^{(1)})$, where $[p_{dij}]^{(1)}$ are drawn from a product Dirichlet distribution with parameters specified by the user and by

default the parameters are $\underbrace{(1, 1, \ldots, 1)}_{4\times n}$. By default, $\{r_i\}$ are set to be $r_1 = r_2 = \ldots = r_{m+1}$. In the *pth* iteration, for $p > 1$, $\Theta^{(p)}$ is calculated from the previous iteration.

(2)$\hat{\Delta}_{d,n,s}$ is calculated as (3.10) shows given $X_{n,s}$ and $\Theta_{(p)}$, where we assume $\Delta_{d,n,s}$ has a multinomial distribution with parameter $p_\Delta \propto \{r_i\} \times f(X_{n,s})$.

(3) $\Theta_{(p+1)}$ is the maximum likelihood estimator (MLE) which maximizes its posterior distribution given the $\hat{\Delta_{d,n,s}}$ and $\{X_{n,s}\}$. It is known that the posterior distribution of Θ is a product Dirichlet distribution, thus, it is easier to get the MLE.

Iterations end when the difference of Θs from two preceding iterations is small enough.

3.3 Modified EM algorithm

The EM algorithm described above is modified and used in XPRIME. Some extra steps are added between the E step and M step. First, in order to avoid that the starting positions of different motifs of interest are located too close to each other in a sequence, the "Phase Shift" function is used to make sure that only one motif is located within a small region of a sequence. Second, within a motif, "correlation" between two different positions is incorporated to the calculation of the likelihood scores for W-mers. In this way, the within motif dependence for the TFBSs are considered in the motif searching. For a single DNA motif, the strong motif form will be stronger and the less frequent motif form will be attenuated or even removed. Furthermore, valuable prior information can be incorporated into motif searching, such as the nucleosome positioning data for given sequences.

Below is the pseudo-code description of XPRIME with the modified EM algorithm, and the distinct motifs are searched in parallel.

•*Input* :

dataset of S sequences

NMOTIF (number of motifs of interest to be searched for)

W (width of motifs of interests to be searched for)

PWMs (prior PWMs for motifs of interest and the motif for background)

ITERS (number of iterations to run XPRIME)

•*Algorithm* : *All S sequences are combined to be one big sequence and are read in as a matrix.*

for iteraction=1 to ITERS {

E step

Phase Shift

Likelihood scores for each W-mer is scaled by the correlation factors and/or nucleosome positioning scores.

M step

}

3.4 Algorithm implementation

Search Both DNA Strands

XPRIME can search for TFBSs on both DNA strands. More specifically, according to the manner of the base-pairing, "CCTA" is the reverse complement of "GGAT". If a TF binds to "GGAT" on one strand, it can also bind to "CCTA" on the same strand, because binding to "CCTA" on the strand means it binds to "GGAT" on the other strand. So "CCTA" is also considered when "GGAT" is the TFBS of interest. XPRIME created the PWM corresponding to the reverse complement when given a PWM. For example, if the following PWM for ETS1 is given,

Table 3.1: The PWM of the binding motif of the TF ETS1.

Position	1	2	3	4	5	6	7	8
A	0.067	0.333	0.0	0.0	1.0	0.533	0.267	0.067
C	0.933	0.600	0.0	0.0	0.0	0.133	0.067	0.400
G	0.000	0.000	1.0	1.0	0.0	0.000	0.667	0.000
T	0.000	0.067	0.0	0.0	0.0	0.333	0.000	0.533

the reverse complement will be

Table 3.2: The PWM of the binding motif of the reverse complement TF ETS1.

Position	1	2	3	4	5	6	7	8
A	0.533	0.000	0.333	0.0	0.0	0.0	0.067	0.000
C	0.000	0.667	0.000	0.0	1.0	1.0	0.000	0.000
G	0.400	0.067	0.133	0.0	0.0	0.0	0.600	0.933
T	0.067	0.267	0.533	1.0	0.0	0.0	0.333	0.067

Both the motif corresponding to the given PWM and the one corresponding to the reverse complement PWM are searched in the given data set, and they will be counted as the same motif.

Phase Shift

Most eukaryotic genes are not regulated by a single TF but by multiple TFs that bind to distinct TFBSs, which are called cis-regulatory modules (CRMs) (Gupta and Liu 2005). These TFs regulate the transcription of the gene in combination. For multiple motifs or multiple copies of a motif found in DNA sequences, they can not be too "overlapping" to each other. Thus, after the starting positions of a motif are located by expectation in the E step of XPRIME, we need to make sure that in the region around the motif, no more than one motif is found. The likelihood scores for all of the W-mers associated with all motifs are calculated following (3.1), and then the "Phase Shift" function is used to check all of the likelihood scores within a DNA region and to turn the all of the likelihood scores other than the highest score to be zero. The width of "Phase Shifted" DNA is specified by the user. By default, the "Phase Shifted" width is W. The "Phase Shifted" function only applies to the likelihood scores of W-mers given the W-mers belong to the motifs of interest not the background motifs.

Table 3.3 shows an example to illustrate the the procedure of "Phase Shift". The first column shows a region of a DNA sequence and given this sequence, three motifs plus one background motif with $W = 4$ are going to be searched by XPRIME. Each nucleotide represents a starting position of a 4-mer. For example, the first score on the left top is

0.00533, which is the likelihood score for the 4-mer "ACGG" starting with "A", given the 4-mer belongs to the motif 1. The dot circle in the first table contains the first three 4-mers and in the second table it can be seen that only the highest score is left with other scores turned to be zero. And then Phase Shift continues to check the next three 4-mers in the same manner.

Although"Phase Shift" function may help to improve the efficiency of XPRIME to search for motifs with less noise, the way that "Phase Shift" works is biased. If a score in the last row in a W bp frame is kept and then a score in the first row in the proceeding W bp frame is kept, two motifs that are left still overlap each other with $W-1$ nucleotides. It is thought that for W bp motifs, if they overlap, it is better for them to overlap at no more than $[W/2]$ nucleotides. In the case of the example in Table 3.3, the possibility that there are still motifs left by "Phase Shift" overlapping with each other at 2 nucleotides is $\frac{1}{4} \times \frac{1}{4} = \frac{1}{16}$, which is relatively small. And in reality, TFBSs are usually 5bp to 20bp, so the error probabilities of "Phase Shift" are even smaller when running with real datasets.

Table 3.3: "Phase Shift" the likelihood scores of W-mers to avoid that motifs are overlapping too much with each other.

SP \ Motif	1	2	3	background
A	0.00533	0.000	0.0033	0.0331
C	0.010	0.00667	0.0025	0.0212
G	0.0040	0.0067	0.0023	0.0010
G	0.067	0.0267	0.0052	0.042
T	0.0017	0.0267	0.0052	0.032
T	0.0027	0.0137	0.0031	0.01
A	0.003	0.267	0.0022	0.01
T	0.067	0.0267	0.0052	0.042
C	0.0017	0.0267	0.0033	0.032
A	0.0027	0.267	0.0031	0.01
T	0.003	0.267	0.0022	0.01

SP \ Motif	1	2	3	background
A	0	0	0	0
C	0	0	0	0
G	0	0	0	0
G	0.067	0	0	0
T	0.0017	0.0267	0.0052	0.032
T	0.0027	0.0137	0.0031	0.01
A	0.003	0.267	0.0022	0.01
T	0.067	0.0267	0.0052	0.042
C	0.0017	0.0267	0.0033	0.032
A	0.0027	0.267	0.0031	0.01
T	0.003	0.267	0.0022	0.01

SP \ Motif	1	2	3	background
A	0	0	0	0
C	0	0	0	0
G	0	0	0	0
G	0.067	0	0	0
T	0	0	0	0
T	0	0	0	0
A	0	0.267	0	0
T	0	0	0	0
C	0.0017	0.0267	0.0033	0.032
A	0.0027	0.267	0.0031	0.01
T	0.003	0.267	0.0022	0.01

Scaling the Likelihood Score for Each W-mer by the Motif Correlation Factor

PWMs and the corresponding sequence logos are popularly used by DNA motif searching methods to represent the found motifs in the input sequence sets. This way to represent resulted motifs has a limitation. Here we use an example to describe this limitation. Suppose that Figure 3.1 shows the sequence logo for the found motif and there are four different copy forms for this motif, which shows in Figure 3.2. If the four forms spread in the given sequences evenly, which means the proportion of the number of each form to the total number of the motif is $\frac{1}{4}$, it can be found that the combination "AG" at the position 5 and 6 is twice as frequent as the "AT" and "CT", which probably indicates given "A" at the position 5, it is more likely to find "G" and "T" than "A" and "C" at the position 6. Neither the sequence logo nor PWM can present structure.

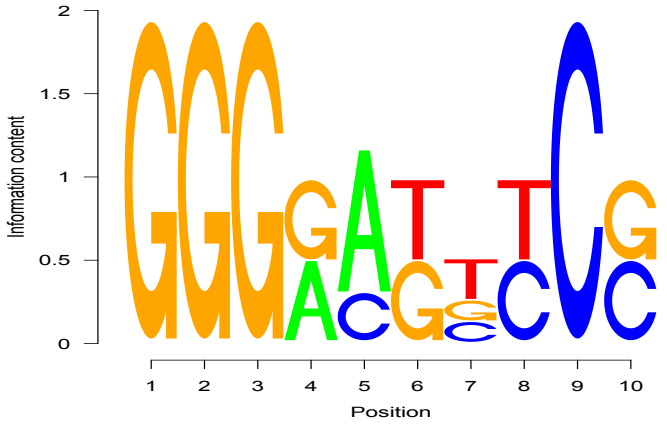

Figure 3.1: SeqLogo for the posterior motif.

GGGG **AT** TCCC
GGGA **AG** CTCG
GGGG **CT** TCCC
GGGA **AG** GTCG

Figure 3.2: Four W-mers belong to the found motif and they spread in the sequences evenly.

In some cases, researchers would like to identify stronger motifs and the forms with the combination "AG" needs to be weighted more than the ones with combinations "AT" and "CT". The “correlation factor” can be used. First, all of the W-mers that have non-zero likelihood scores given the motif of interest are collected and different nucleotide combinations at any two different positions within the motif are considered. There are $(1+2+\ldots+(W-1))$ sets of two different positions and for each two positions, there are 16 different nucleotide combinations: “AA”, “AC”, “AG”,“AT”,“CA”,“CC”, “CG”,“CT”, “GA”,“GC”, "“GG”, “GT”, “TA”,“TC”, “TG” and “TT”.

At first, $(1 + 2 + \ldots + (W - 1))4 \times 4$ matrices are generated. Table 3.4 shows an example of a matrix for position 2 and 3. The column names are nucleotides at position 3 and the row names are nucleotides at position 2. For example, for "“AA” at position 2 and 3, we pick up all collected W-mers that have “AA” at position 2 and 3 and get the sum of the likelihood scores of these W-mers, which is 13.067. This number is the $(1, 1)$ in the matrix for position 2 and 3. Elements are the sum of likelihood scores calculated as described above.

Table 3.4: The matrix for position 2 and position 3.

Position 2&3	A	C	G	T
A	13.067	17.333	11.533	10.267
C	12.933	14.600	8.067	9.400
G	15.000	12.000	12.667	11.533
T	16.000	9.067	7.333	12.634

Secondly, each matrix is normalized by dividing elements by the sum of all elements in the matrix. And all matrices have the same mean and then they are ranked by their variance. A matrix with higher variance represents two positions where the frequencies of different nucleotide combinations are very different. T matrices with highest variances are picked up. If W is odd, $T = (W - 1)/2$ and if W is even, $T = W/2$. When picking out the T matrices, we need to make sure that two different pairs are not allowed to have common positions

Finally, the T matrices are used to scale the likelihood scores of all W-mers with non-zero scores given the motif after E step before M step in every iteration in XPRIME. The likelihood score of each W-mer is multiplied by T numbers extracted from the T matrices. The T numbers correspond to the combinations of nucleotides of the W-mer at the positions of T matrices. For a W-mer with "AC" at its position 2 and 3, its likelihood score is then multiplied by 11.533.

Therefore, more frequent motif forms are more weighted and less frequent motif forms are less weighted. The rescaling enables the user to get stronger motifs and understate the other less frequent W-mers.

Incorporating the nucleosome positioning scores for the input sequences in XPRIME

Recall that in the Introduction nucleosome positioning regulating transcription and the presence of nucleosomes inhibiting the binding of TF to the DNA region was discussed. Valouev et al., have published the genome-wide nucleosome positioning data for human active CD4+ T cells, which contain 342 million reads from high-throughput SOLiD sequencing (Valouev et al. 2011). GNUMAP (Genomic Next-generation Universal MAPper) may be used to map these reads to any human sequences(Clement et al. 2010). We have mapped these reads to the ETS1 bound sequences, which were obtained from the Graves lab. From the mapping result, the numbers of reads, which were mapped to ETS1 bound regions, have been counted. These counted numbers are plotted against the relative position of the reads to the ETS1 bound regions (-550bp to 550bp). That is to say, in the plots in Figure 3.3, x-axis represents the relative positions of nucleosome reads to the ETS1 bound regions (distances from the regions) and y-axis represents the counts of those reads. It could be seen that around both enhancer and promoter ETS1 motif regions, the nucleosome counts are low, and the nucleosome counts increases with the distance from the motifs increasing .

By using the FASTQ sequence file, the improved XPRIME is able to incorporate the nucleosome positioning scores into the motif searching under the assumption that it is more

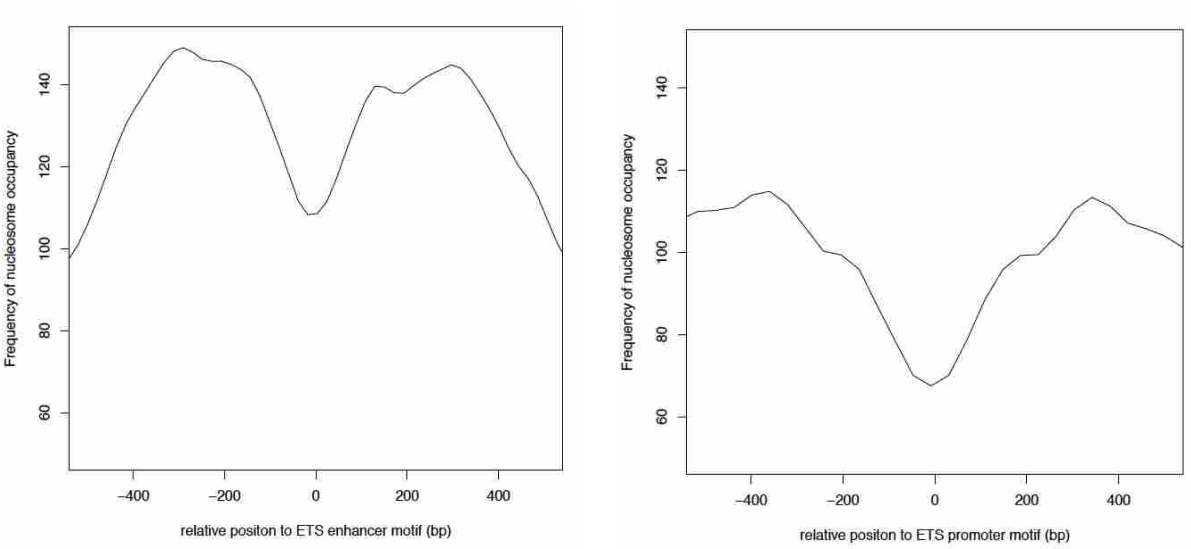

Figure 3.3: Nucleosome positioning around the ETS1 motifs

possible that TFBSs appear in the nucleosome free regions than in the nucleosome occupied regions.

The FASTA sequence file format is the most popular input format for DNA motif searching tools, including XPRIME. Each sequence in FASTA file contains two lines, the first line starts with "$>$" and is the title line and the second line provides the DNA sequence line. The FASTQ file format emerges with the development of DNA sequencing technologies, which now is a common format for data exchange between results from different DNA sequencing methods. FASTQ file is an extension of FASTA file and it stores a numerical sequencing quality score for each nucleotide in a sequence.

Figure 3.4 presents a sequence extracted from a FASTQ file (Cock et al. 2010). Each sequence in FASTQ files contains four lines: the first line begins with a "@" title line, which is a free format field with no length limit to provide the ID, length or other other comments about the sequence. The second line is the sequence line. The third line starts with a "+". Usually this line just repeats the information in the first line and is also a free format filed. The last line is the quality line and stores sequencing quality score per base of the sequence with ASCII printable characters.

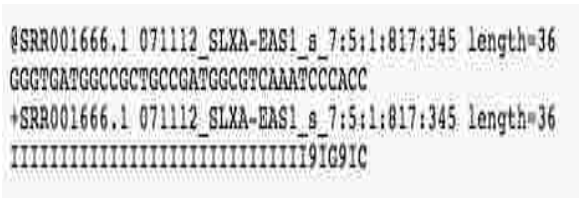

Figure 3.4: A sequence in a FASTQ file (Cock et al. 2010).

ASCII is the abbreviation for American Standard Code for Information Interchange, which is based on the ordering of English alphabets and is used to represent the sequencing quality scores. A single character in ASCII is simpler than a double digit number and it is more space efficient than digits. Since ASCII 32 is the space character, which is not so human readable, ASCII 33 to 126 are usually used to represent sequencing quality scores 0 to 93. People usually call these numerical scores PHRED quality scores, since PHRED software firstly reads DNA sequencing trace files and assigns this numerical scores to each base. PHRED quality scores eventually can represent the estimated probability of sequencing error P_e for each nucleotide in a sequence by the equation below,

$$Q_{PHRED} = -10 \times log_{10}(P_e) \tag{3.12}$$

There are three variants that use ASCII to represent the sequencing quality scores in FASTQ file. They are used by three different DNA sequencing groups: "fastq-sanger" used by Sanger institute, "fastq-solexa" introduced by Solexa. Inc and "fastq-illumina" used by Iluumina. In "fastq-sanger", the PHRED scores 0 to 93 are represented as 33 to 126 by ACSII and the scores in "fastq-sanger" minus 33 equal the PHRED scores. In "fastq-illumina" the PHRED scores 0 to 62 are represented as 64 to 126 by ACSII and the scores in "fastq-illumina" minus 64 equal the PHRED scores. Thus, "fastq-sanger" includes larger range of PHRED scores than "fastq-illumina". "fastq-solexa" has its own conversion way to convert its ASCII scores to PHRED scores : $Q_{PHRED} = -10 \times log_{10}(10^{Q_{SOLEXA}//10} + 1)$ and it uses ASCII 59 to 126 to represent the PHRED scores -5 to 62.

The FASTQ sequence file is used to incorporate the nucleosome positioning data into the motif searching. The nucleosome count at each position of the given sequences is

proportionally rescaled to be between 33 and 126 and then can be used as Q_{NUCLEO} as in (3.10), which is analogous to (3.9), and P_n is analogously defined as the estimated possibility of nucleosome-free.

$$Q_{NUCLEO} = -10 \times log_{10}(P_n) \tag{3.13}$$

We assume that it is more likely that motifs of interest are located and found in the sequence regions with lower nucleosome positioning counts (higher P_n) and the background motif is located and found in sequence regions with higher nucleosome positioning counts (lower P_n). In order to incorporate the nucleosome positioning information in motif searching, the likelihood score for a W-mer is multiplied by a factor that can reflect the possibility of finding motifs in the W-mer. For the motifs of interest, the factor is calculated as $\prod_{i=1}^{W} \frac{P_{ni}}{1-P_{ni}}$ and for the background motif, the factor is calculated as $\prod_{i=1}^{W} \frac{1-P_{ni}}{P_{ni}}$, where P_{ni} is the estimated possibility of nucleosome-free for the nucleotide at ith position within the W-mer. We do not use the $\prod_{i=1}^{W} P_{ni}$ or $\prod_{i=1}^{W}(1-P_{ni})$ because in this way factor may be very small to make the likelihood score very low and bring inaccuracy in computation.

Therefore, the likelihood score for the W-mer $X_{n,s}$ given $X_{n,s}$ belongs to the dth motif with the nucleosome positioning data incorporated can be rescaled as

$$f(X_{n,s}|\Delta_d = 1)_{score} = \begin{cases} \prod_{i=1}^{W}[\prod_{j\in\{A,C,G,T\}} p_{dij}^{I(X_{(n+i-1),s}=j)} P_{ni}/(1-P_{ni})] & \text{if } 1 \le d \le m \\ \prod_{i=1}^{W}[\prod_{j\in\{A,C,G,T\}} p_{dij}^{I(X_{(n+i-1),s}=j)} (1-P_{ni})/P_{ni}] & \text{if } d = m+1 \end{cases}$$

3.5 Simulation study

In this study, DNA sequences will be generated at random according to the nucleotide ratio A : C : G : T = 1 : 1 : 1 : 1. Each sequence set contains 100 sequences and each sequence is 550bp. The nucleosome occupancy scores will be assigned to all positions of those sequences. Three DNA motifs of 8bp in length from TRANSFAC (Wingender 2008), called ETS, TAL and FTZ will be planted at random in these simulated sequences (with replacement). The

overwritten of the planted motifs will be avoided. The sequence logos for these motifs are shown in Figure 3.5, 3.6, and 3.7. The TAL motifs will be only planted at random in the nucleosome occupied regions. The FTZ motifs will be only planted in the nucleosome free regions. And the ETS motifs will be planted randomly over the entire sequence regions. Following are details of this study.

1. *Motif planting.* The different numbers of occurrences of each motif will be chosen. Table 3.5 shows seven different combinations of motif occurrences that will be utilized in the simulated sequence set. 100 times of occurrences would be the medium-level occurrence, which suggests that on average each sequence contains one occurrence of the motif of interest. 200 times of occurrences would be the high-level occurrence and 50 times of occurrences would be the low-level occurrence. Because XPRIME is able to search for both DNA strands, all occurrences will be planted in the same orientation.

Table 3.5: Seven different combinations of motif occurrences.

	ETS	TAL	FTZ
1	100	200	100
2	100	100	100
3	100	50	100
4	100	100	50
5	100	100	200
6	200	100	100
7	50	100	100

2. *Programs running.* XPRIME using EM algorithm (called XPRIME-EM), XPRIME using Gibbs Sampling procedure (Poulsen 2009), which is called XPRIME-GIBBS, and MEME (Bailey and Elkan 1994) will be run with the seven different sequence sets described above. The length of the motif of interest will be set to be 8bp. This process will be repeated five times to obtain five samples. We will randomly generate the DNA sequences each time.

3. *Performance comparing.* To evaluate the performance of the motif searching programs, the *precision* of each program for each motif of interest will be measured, which is defined as $\frac{tp}{tp+fp}$ (Bailey and Elkan 1994). tp is the number of the corrected classified positives ("true

positives"), which is the number of positions where occurrences of the known and reported motifs overlap. fp is the number of the non-occurrences classified as occurrences ("false positives"), which is the number of positions where occurrences of the reported motifs do not overlap with the known motifs (Sinha and Tompa 2003). The *sensitivity* of each program for each motif of interest will also be measured, which is defined as $\frac{tp}{tp+fn}$ (Chen et al. 2007). fn is the number of the occurrences classified as non-occurrences ("false negatives"), which is the number of positions where the known motifs do not overlap with the occurrences of the reported motifs (Sinha and Tompa 2003). For each program, the *precision* and *sensitivity* will be reported for each motif and the overall precision for all motifs will be reported as well.

All of the approaches described above are expected to have high precision and sensitivity for the ETS motif regardless of occurrence levels are. XPRIME-EM with nucleosome positioning score incorporation is expected to has lower precision and sensitivity than the other two programs for the TAL motif. When the TAL motif has low occurrence level, XPRIME-EM with nucleosome positioning score incorporation is expected to fail to identify this motif because it is only planted in the nucleosome occupied regions. When the FTZ motif has low occurrence level, we expect that XPRIME-EM with nucleosome positioning score incorporation may have higher precision and sensitivity than the other programs. To statistically compare the precision rates and sensitivity of different programs, the *Hotelling's* T^2 test will be used.

The *Hotelling's* T^2 test is a multivariate test to compare two vectors of means. Assume there are n_1 observations for treatment 1: $X = (X_{i1}, X_{i2}, \ldots, X_{ik})$, $i = 1, \ldots, n_1$ and n_2 observations for treatment 2: $Y = (Y_{i1}, Y_{i2}, \ldots, Y_{ik})$, $i = 1, \ldots, n_2$, where each treatment has k response variables. The vectors of sample means are $\bar{X} = (\bar{X_1}, \bar{X_2}, \ldots, \bar{X_k})$ and $\bar{Y} = (\bar{Y_1}, \bar{Y_2}, \ldots, \bar{Y_k})$ (Higgins 2003).

Let C be the $k \times k$ matrix of pooled covariance. The uvth element in C matrix C_{uv} can be calculated as $C_{uv} = \frac{(n_1-1)C_{Xuv}+(n_2-1)C_{Yuv}}{n_1+n_2-2}$, where C_{Xuv} and C_{Yuv} are the covariance

between the uth and the vth response variables on treatment 1 and treatment 2 respectively. $C_{Xuv} = \frac{\sum_1^{n_1}(X_{iu}-\bar{X_u})(X_{iv}-\bar{X_v})}{n_1-1}$ and C_{Yuv} is calculated in the analogous way.

The *Hotelling's T^2 test* statistic is defined as $T^2 = \frac{n_1 n_2}{n_1+n_2}(\bar{X}-\bar{Y})'C^{-1}(\bar{X}-\bar{Y})$. Let μ_{Xj} be the population mean for the jth response variable in treatment 1 and μ_{Yj} be the population mean for the jth response variable in treatment 2. The null hypothesis in this test is $H_0 : \mu_{Xj} = \mu_{Yj}, j = 1, \ldots, k$ and the alternative hypothesis is $H_1 : \mu_{Xj} \neq \mu_{Yj}$, for at lease one j, $j = 1, \ldots, k$. Under the null hypothesis, $F = \frac{n_1+n_2-k-1}{(n_1+n_2-2)k}T^2$ follows an F distribution with degrees of freedom k and $n_1 + n_2 - k - 1$.

In this project, a pairwise *Hotelling's T^2* test has been chosen to test if any two of the three programs have the same means of precision rates. Each program could be looked as a treatment. The seven combinations of motif occurrences could be seven response variables. Each treatment has five observation vectors corresponding to five samples of simulation data sets. In this case, $k = 7$ and $n_1 = n_2 = 5$. The *Hotelling's T^2* test requires the assumption that all observations from each treatment are independent and randomly sampled from a multivariate normal population $N(\mu_t, \Sigma_t)$, where t indicates treatments. We can not assure that the assumption of multivariate normality is met due to the small sample size, so the permutation version of the *Hotelling's T^2* test has been used, in which the multivariate observation vectors are permuted between treatments (Higgins 2003). The steps below are followed:

1. Calculate the F statistics for the original observations F_{obs}.

2. For the two treatment with 5 observation vectors each, obtain 10000 random sample of permutations and calculate the F statistics.

3. Obtain the p value as $p = \frac{\text{number of } F \geq F_{obs}}{10000}$.

3.6 Real data set study

XPRIME-EM, XPRIME-GIBBS (Poulsen 2009) and MEME (Bailey and Elkan 1994) will be run using two real sequence sets. These sequences arise from ChIP-chip experiments in

active human CD4+ T cells performed by Barbara Graves Lab at University of Utah. One set contains 100 DNA sequences in which ETS1 TFs occupy the redundant promoters and the other one contains 100 DNA sequences in which ETS1 TFs occupy the specific enhancers. ETS1 is a member of the transcription factor family ETS. It is known that ETS1 TF plays a role in activating T cells and may function in skin cancer development (Torlakovic et al. 2004).

ETS1 TFs that bind to redundant promoters share common consensus binding sites with other members in ETS family, while ETS1 TFs that bind to specific enhancers are associated with distinct binding sequences (Hollenhorst et al. 2009). Besides the ETS1 TFBSs, we will allow 2 different *de novo* DNA motifs in the two sequence sets, whose presence may contribute to the different functions of ETS1TFs when they bind to enhancers and promoters.

Since the real motif occupancy positions are not known, it is impossible to calculate the precision and sensitivity in this case. The reported ETS1 motif and *de novo* motifs obtained by the three DNA motif searching programs will be compared with their occurrence times and their conservativity, which may be seen by the height of the nucleotide at each position of the motif seqlogos. The higher the nucleotide is , the more conservative it is.

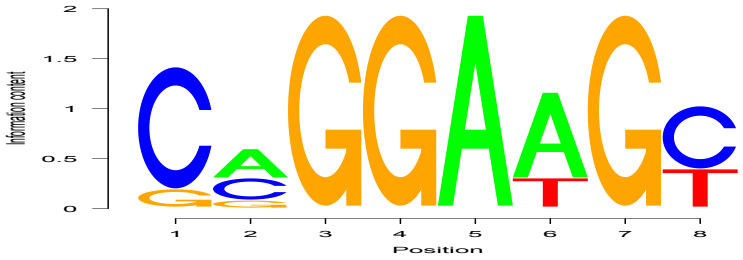

Figure 3.5: SeqLogo for ETS DNA motif from TRANSFAC (Wingender 2008).

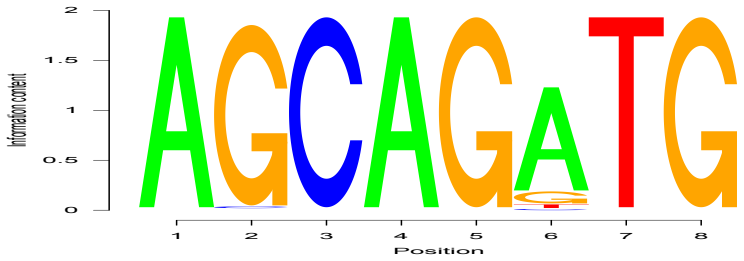

Figure 3.6: SeqLogo for TAL DNA motif from TRANSFAC (Wingender 2008).

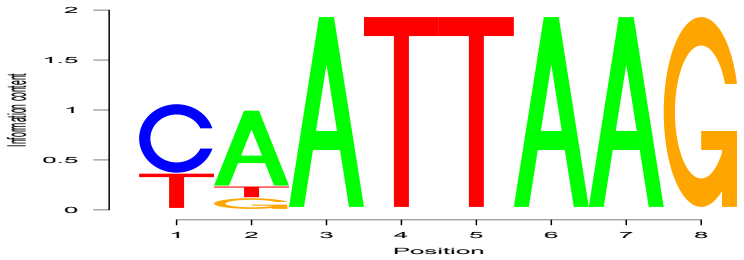

Figure 3.7: SeqLogo for FTZ DNA motif from TRANSFAC (Wingender 2008).

CHAPTER 4

RESULTS

To compare the performance of XPRIME-EM with XPRIME-GIBBS (Poulsen 2009) and MEME (Bailey and Elkan 1994), we analyzed the simulated data sets and the real biological data sets as described in the Methods section. In the simulated data sets, motifs are planted into the DNA sequences with seven different combinations of occurrences as Table 3.5 shows.

4.1 Effects of the motif correlation factor scaling in the modified EM algorithm

In the XPRIME-EM program, one of the proposed steps that we use to modify the EM algorithm is to scale the likelihood score for each W-mer by the motif correlation factor between the E-step and M step. After running XPRIME-EM on the simulated data sets, we have found that although XPRIME-EM with this step always generates very strong motifs as we expect, those motifs are always wrong and do not exist in the given DNA sequences. This results in the zero precision rate and zero sensitivity. Scaling the likelihood score for each W-mer by the motif correlation factor may lead to bias in the likelihood score for W-mers and lead the EM algorithm to converge to wrong stationary points. Therefore, we decided to exclude the " scaling the likelihood scores for each W-mer by the motif correlation factor" step in the XPRIME-EM program.

4.2 Performance evaluation on simulated data sets for *de novo* motifs discovery

We ran the XPRIME-EM, XPRIME-GIBBS and MEME on the simulated sequence sets. Random Dirichlet priors for motif PWMs are given in XPRIME-EM and XPRIME-GIBBS

for *de novo* or unknown motifs discovery. All of the three programs seek to identify three different motifs in each of the given sequence set.

Precision Rates

We are interested in the effects of incorporating the nucleosome positioning scores on DNA motif searching, so we ran the XPRIME-EM with and without nucleosome positioning scores incorporation to compare their precision rates. After running 10000 iterations with burn-in size 1000, XPRIME-GIBBS could not find any motif in all given DNA sequence sets. All elements of the posterior means PWMs are around 0.25. Therefore, the precision rates and sensitivity of XPRIME-GIBBS for *de novo* motifs discovery will not be reported.

Table 4.1 shows the precision rates of XPRIME-EM with and without nucleosome positioning scores incorporation and MEME for *de novo* motifs discovery. The first rows are for motif ETS, FTZ and TAL respectively. The last three rows are the overall precision rates for the three programs. Each entry in the table represents the average precision rate over five samples, each of which contains 100 DNA sequences with 550bp length.

Table 4.1: Precision rates of XPRIME-EM with and without nucleosome positioning scores incorporation and MEME for *de novo* motifs discovery.

Motif	Programs \ Data sets	121	111	151	115	112	211	511
ETS	XPRIME-EM w/o nucleo	0.77	0.77	0.80	0.76	0.86	0.90	0.52
	XPRIME-EM w/ nucleo	0.61	0.63	0.58	0.32	0.33	0.89	0.26
	MEME	0.93	0.94	0.91	0.94	0.74	0.95	0.71
FTZ	XPRIME-EM w/o nucleo	0.80	0.57	0.09	0.54	0.36	0.67	0.58
	XPRIME-EM w/ nucleo	0.66	0.44	0.05	0.39	0.14	0.16	0.24
	MEME	0.96	0.94	0.89	0.95	0.75	0.94	0.96
TAL	XPRIME-EM w/o nucleo	0.81	0.87	0.85	0.26	0.94	0.79	0.62
	XPRIME-EM w/ nucleo	0.79	0.78	0.82	0.57	0.87	0.68	0.77
	MEME	0.95	0.94	0.91	0.91	0.97	0.95	0.94
Overall	XPRIME-EM w/o nucleo	0.80	0.74	0.68	0.57	0.78	0.81	0.58
	XPRIME-EM w/ nucleo	0.68	0.62	0.57	0.40	0.55	0.65	0.46
	MEME	0.95	0.94	0.91	0.94	0.86	0.95	0.90

Table 4.2 presents the F_{obs} value and the associated p value for each permutation *Hotelling's* T^2 test. For the motif ETS, the test results indicate that the three programs have the same mean precision rates at $\alpha = 0.05$ level for all motif occurrence combinations. For the motif FTZ, The significant p values 0.02 and 0.03 suggests that MEME has significantly different mean precision rates than XPRIME-EM with and without nucleosome positioning scores incorporation for at least one motif occurrence combination respectively, while XPRIME-EM with and without nucleosome positioning scores incorporation have the same mean precision rates for all motif occurrence combinations. For the motif TAL, MEME has significantly different mean precision rates than XPRIME-EM with nucleosome positioning scores incorporation and marginally significantly different mean precision rates than XPRIME-EM without nucleosome positioning scores incorporation for at least one motif occurrence combination.

Table 4.2: The permutation *Hotelling's* T^2 test results on the mean precision rates of every two programs for each motif.

		F_{obs}	p value
ETS	MEME vs. XPRIME-EM w/o nucleo	5.78	0.06
	XPRIME-EM w/ nucleo vs. w/o nucleo	0.96	0.54
	MEME vs. XPRIME-EM w/ nucleo	4.04	0.16
FTZ	MEME vs. XPRIME-EM w/o nucleo	91.90	0.02
	XPRIME-EM w/ nucleo vs. w/o nucleo	0.59	0.73
	MEME vs. XPRIME-EM w/ nucleo	70.94	0.03
TAL	MEME vs. XPRIME-EM w/o nucleo	12.99	0.04
	XPRIME-EM w/ nucleo vs. w/o nucleo	5.56	0.11
	MEME vs. XPRIME-EM w/ nucleo	14.54	0.07

For each of the three permutation *Hotelling's* T^2 tests with significant p values, the permutation two-sample t-test has been performed for each individual response variable. Table 4.3 presents the t_{obs} values and the associated p values based on 10000 random sample of permutations.

All t_{obs} values are positive and all p values are significant in this table, indicating that for motif FTZ, the mean precision rates of MEME are significantly higher than XPRIME-EM

Table 4.3: The permutation two-sample t-test results on the mean precision rates for each individual motif occurrence combination.

		121	111	151	115	112	211	511
FTZ MEME vs. XPRIME-EM w/o nucleo	t_{obs}	7.58	4.31	9.06	5.24	1.61	7.22	4.36
	p value	0.00	0.00	0.00	0.00	0.00	0.00	0.01
FTZ MEME vs. XPRIME-EM w/ nucleo	t_{obs}	1.79	3.71	15.22	3.49	2.66	4.96	4.83
	p value	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TAL MEME vs. XPRIME-EM w/o nucleo	t_{obs}	2.90	1.99	1.15	3.88	2.52	3.75	1.93
	p value	0.00	0.00	0.00	0.00	0.00	0.00	0.01

with and without nucleosome positioning scores incorporation for all combinations of motif occurrences. For motif TAL, the mean precision rates of MEME are significantly higher than XPRIME-EM without nucleosome positioning scores incorporation for all combinations of motif occurrences. Therefore, in the level of precision rates, MEME performance on the simulated data set is nearly optimal and incorporating the nucleosome positioning scores does not significantly improve XPRIME-EM's performance.

Sensitivity

Table 4.4 shows the sensitivity of XPRIME-EM with and without nucleosome positioning scores incorporation and MEME for unknown motifs discovery. The first rows are for motif ETS, FTZ and TAL respectively. The last three rows are the overall sensitivity for the three programs. Each entry in the table represents the average sensitivity over five samples, each of which contains 100 DNA sequences with 550bp length. We expected that when the level of FTZ is low, XPRIME-EM with nucleosome positioning scores incorporation may fail to identify it because this motif is only planted in the nucleosome occupied region. As we can see in the table, the FTZ motif has been barely identified in the sequence set with motif occurrence combination 151 by XPRIME-EM with nucleosome positioning scores incorporation and the sensitivity is 0.02, which is lowest among all motif occurrence combinations.

Table 4.4: Sensitivity of XPRIME-EM with and without nucleosome positioning scores incorporation and MEME for unknown motifs discovery.

Motifs	Programs \ Data sets	121	111	151	115	112	211	511
ETS	XPRIME-EM w/o nucleo	0.63	0.61	0.66	0.63	0.72	0.85	0.50
	XPRIME-EM w/ nucleo	0.50	0.50	0.48	0.32	0.31	0.75	0.11
	MEME	0.70	0.69	0.67	0.70	0.58	0.71	0.53
FTZ	XPRIME-EM w/o nucleo	0.64	0.30	0.03	0.24	0.17	0.40	0.14
	XPRIME-EM w/ nucleo	0.43	0.22	0.02	0.19	0.05	0.20	0.05
	MEME	0.94	0.89	0.83	0.85	0.71	0.83	0.75
TAL	XPRIME-EM w/o nucleo	0.71	0.83	0.75	0.25	0.86	0.50	0.48
	XPRIME-EM w/ nucleo	0.98	0.99	0.97	0.91	1.00	0.68	0.94
	MEME	0.96	0.97	0.98	0.92	0.98	0.95	0.97
Average	XPRIME-EM w/o nucleo	0.65	0.58	0.57	0.40	0.66	0.65	0.35
	XPRIME-EM w/ nucleo	0.58	0.57	0.59	0.39	0.59	0.60	0.42
	MEME	0.89	0.85	0.83	0.80	0.81	0.80	0.79

Table 4.5 presents the F_{obs} statistics and the associated p values for the permutation *Hotelling's* T^2 tests on the mean sensitivity of every two programs for each motif.

For the motif ETS, the test results indicate that MEME has significantly different mean sensitivity than XPRIME-EM without nucleosome positioning scores incorporation

for at least one motif occurrence combination. For the motif FTZ, The significant p values 0 suggests that MEME has significantly different mean sensitivity than XPRIME-EM with and without nucleosome positioning scores incorporation, respectively, for at least one motif occurrence combination, while XPRIME-EM with and without nucleosome positioning scores incorporation have the same mean precision rates for all motif occurrence combinations. For the motif TAL, XPRIME-EM without nucleosome positioning scores incorporation has significantly different mean sensitivity than XPRIME-EM with nucleosome positioning scores incorporation and MEME, respectively, for at least one motif occurrence combination, while XPRIME-EM with nucleosome positioning scores incorporation and MEME have the same mean sensitivity for all motif occurrence combinations.

Table 4.5: The permutation *Hotelling's* T^2 test results on the mean sensitivity of every two programs for each motif.

		F_{obs}	p value
ETS	MEME vs. XPRIME-EM w/o nucleo	24.17	0.01
	XPRIME-EM w/ nucleo vs. w/o nucleo	0.96	0.54
	MEME vs. XPRIME-EM w/ nucleo	3.61	0.18
FTZ	MEME vs. XPRIME-EM w/o nucleo	127.43	0.00
	XPRIME-EM w/ nucleo vs. w/o nucleo	2.62	0.20
	MEME vs. XPRIME-EM w/ nucleo	1936.93	0.00
TAL	MEME vs. XPRIME-EM w/o nucleo	382.49	0.00
	XPRIME-EM w/ nucleo vs. w/o nucleo	2447.13	0.00
	MEME vs. XPRIME-EM w/ nucleo	1.26	0.52

For each of the permutation *Hotelling's* T^2 tests with significant p values in Table 4.5, the permutation two-sample t-test has been performed for each individual response variable. Table 4.6 presents the t_{obs} statistics and the associated p value based on 10000 random sample of permutations.

All t_{obs} values are positive and all p values are significant in this table except for the motif occurrence combinations 112, 211 and 511 for the motif ETS. The results indicate that for the motif ETS, MEME has significantly higher mean sensitivity than XPRIME-EM without nucleosome positioning scores incorporation for the motif occurrence combinations

Table 4.6: The permutation two-sample t-test results on the mean sensitivity for each individual motif occurrence combination.

		121	111	151	115	112	211	511
ETS MEME vs. XPRIME-EM w/o nucleo	t_{obs}	0.89	0.95	0.19	0.90	-0.92	-3.68	0.13
	p value	0.00	0.00	0.00	0.00	1.00	1.00	0.42
FTZ MEME vs. XPRIME-EM w/o nucleo	t_{obs}	6.82	5.11	16.81	7.04	2.72	3.88	11.80
	p value	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FTZ MEME vs. XPRIME-EM w/ nucleo	t_{obs}	3.33	5.46	18.61	5.76	3.49	3.08	16.08
	p value	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TAL MEME vs. XPRIME-EM w/o nucleo	t_{obs}	2.30	1.38	1.75	4.06	10.70	9.11	3.82
	p value	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TAL XPRIME-EM w/ vs. w/o nucleo	t_{obs}	2.43	1.56	1.70	3.84	14.43	1.41	3.54
	p value	0.00	0.00	0.00	0.00	0.00	0.00	0.00

121, 111, 151 and 115 and the two programs have the same mean sensitivity for the motif occurrence combinations 112, 211 and 511. For motif FTZ, the mean sensitivity of MEME are significantly higher than XPRIME-EM with and without nucleosome positioning scores incorporation for all combinations of motif occurrences. For motif TAL, the mean sensitivity of MEME and XPRIME-EM with nucleosome positioning scores incorporation are significantly higher than XPRIME-EM without nucleosome positioning scores incorporation for all combinations of motif occurrences.

Therefore, in the level of sensitivity, MEME performance on the simulated data set is nearly optimal. However, for identifying motifs that only locate in the nucleosome free region, incorporating the nucleosome positioning scores significantly improves XPRIME-EM's performance and the improved performance of XPRIME-EM is comparable to MEME's performance.

4.3 Performance comparison for known motifs discovery

In order to compare the EM algorithm and the Gibbs Sampling algorithm on the known motif searching, we have also run the XPRIME-EM without nucleosome positioning information incorporation and XPRIME-GIBBS on the simulated sequence sets with the PWMs for motifs given as the prior PWMs. Table 4.7 shows the precision rates of XPRIME-EM

without nucleosome positioning scores incorporation and XPRIME-GIBBS for known motifs discovery. The first rows are for motif ETS, FTZ and TAL respectively. The last three rows are the overall precision rates for the two programs. Each entry in the table represents the average precision rate over five samples, each of which contains 100 DNA sequences with 550bp length.

Table 4.7: Precision rates of XPRIME-EM without nucleosome positioning scores incorporation and XPRIME-GIBBS for known motifs discovery.

Motifs	Data sets / Programs	121	111	151	115	112	211	511
ETS	XPRIME-GIBBS	0.04	0.07	0.07	0.02	0.05	0.08	0.01
	XPRIME-EM w/o nucleo	0.75	0.76	0.75	0.75	0.76	0.85	0.60
FTZ	XPRIME-GIBBS	0.51	0.44	0.43	0.45	0.36	0.57	0.44
	XPRIME-EM w/o nucleo	0.96	0.91	0.85	0.92	0.92	0.92	0.92
TAL	XPRIME-GIBBS	0.59	0.83	0.56	0.57	0.73	0.52	0.66
	XPRIME-EM w/o nucleo	0.89	0.89	0.89	0.81	0.94	0.89	0.89
Overall	XPRIME-GIBBS	0.41	0.45	0.34	0.30	0.47	0.31	0.44
	XPRIME-EM w/o nucleo	0.89	0.85	0.83	0.83	0.89	0.88	0.85

Table 4.8 presents the F_{obs} statistic and the associated p value for each of the permutation *Hotelling's* T^2 tests. The significant p values at $\alpha = 0.05$ level indicate that XPRIME-EM without nucleosome positioning scores incorporation and XPRIME-GIBBS have significantly different precision rates for at least one motif occurrence, respectively, for all three motifs.

Table 4.8: The permutation *Hotelling's* T^2 *test* results on the mean precision rates of the two programs for each motif.

	F_{obs}	p value
ETS	1807.76	0.00
FTZ	1401.48	0.00
TAL	779.30	0.01

For each of the three permutation *Hotelling's* T^2 tests, the permutation two-sample t-test has been performed for each individual response variable. Table 4.9 presents the t_{obs} statistics and the associated p value based on 10000 random sample of permutations. All p

values are significant in this table and all observed t statistics are positive, indicating that for all three motifs and all motif occurrence combinations, XPRIME-EM without nucleosome positioning scores incorporation has significantly higher mean precision rates than XPRIME-GIBBS.

Table 4.9: The permutation two-sample t-test results on the mean precision rates for each individual motif occurance combination.

		121	111	151	115	112	211	511
ETS	t_{obs}	29.14	21.82	22.19	106.72	29.23	19.02	96.47
	p value	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FTZ	t_{obs}	30.26	4.06	2.59	4.06	21.02	2.60	4.29
	p value	0.00	0.00	0.00	0.00	0.00	0.00	0.01
TAL	t_{obs}	22.08	0.83	4.12	2.39	29.64	23.13	4.19
	p value	0.00	0.00	0.00	0.00	0.00	0.00	0.01

Table 4.10 shows the sensitivity of XPRIME-EM without nucleosome positioning scores incorporation and XPRIME-GIBBS for known motifs discovery. The first rows are for motif ETS, FTZ and TAL respectively. The last three rows are the overall sensitivity for the three programs. Each entry in the table represents the average sensitivity over five samples, each of which contains 100 DNA sequences with 550bp length. From the table, it can be seen that the sensitivity of both programs are fairly high and around 100 percent.

Table 4.10: Sensitivity of XPRIME-EM without nucleosome positioning scores incorporation and XPRIME-GIBBS for known motifs discovery.

Motifs	Data sets / Programs	121	111	151	115	112	211	511
ETS	XPRIME-GIBBS	0.99	0.94	0.86	0.93	0.99	0.98	0.97
	XPRIME-EM w/o nucleo	0.99	1.00	1.00	1.00	1.00	1.00	1.00
FTZ	XPRIME-GIBBS	1.00	0.98	0.98	0.99	0.99	1.00	0.99
	XPRIME-EM w/o nucleo	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TAL	XPRIME-GIBBS	0.97	0.99	0.98	0.99	0.99	0.99	0.99
	XPRIME-EM w/o nucleo	1.00	1.00	1.00	1.00	1.00	1.00	0.99
Overall	XPRIME-GIBBS	0.99	0.97	0.93	0.96	0.99	0.99	0.98
	XPRIME-EM w/o nucleo	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 4.11 presents the F_{obs} statistics and the associated p value for each permutation *Hotelling's* T^2 test. All p values from the three tests are not significant at $\alpha = 0.05$ level, suggesting that XPRIME-EM without nucleosome positioning scores incorporation and XPRIME-GIBBS have the same sensitivity for the known motif discovery for all motif occurrence combinations.

Table 4.11: The permutation *Hotelling's* T^2 *test* results on the mean sensitivity of the two programs for each motif.

	F_{obs}	p value
ETS	20.96	0.09
FTZ	–3.29e+14	0.99
TAL	4.24	0.09

Therefore, for the known motif discovery, XPRIME-EM without nucleosome positioning scores incorporation and XPRIME-GIBBS have the same mean sensitivity and XPRIME-EM without nucleosome positioning scores incorporation has significantly higher precision rates than XPRIME-GIBBS.

4.4 Motifs discovery on real biological data sets

In order to evaluate the performance of XPRIME-EM with and without nucleosome positioning scores incorporation, XPRIME-GIBBS and MEME on real biological data, these approaches were assessed using two sequence sets: one set contains 100 DNA sequences in which ETS1 TFs occupy the redundant promoters and the other one contains 100 DNA sequences in which ETS1 TFs occupy the specific enhancers. Each sequence in these two data sets are 550bps. Since *GGA* are known as the most conservative part in the ETS1 motifs, the prior PWM shown in Table 4.12 was utilized for the ETS1 motif in XPRIME-EM with and without nucleosome positioning scores incorporation and XPRIME-GIBBS. Two random Dirichlet priors are given for the two *de novo* motif searching. The *GGA* is located in position 3, 4 and 5 and in other positions, the nonsense priors are given.

Table 4.12: The prior PWM for the ETS1 motif.

Position	1	2	3	4	5	6	7	8
A	0.25	0.25	0.00	0.00	1.00	0.25	0.25	0.25
C	0.25	0.25	0.00	0.00	0.00	0.25	0.25	0.25
G	0.25	0.25	1.00	1.00	0.00	0.25	0.25	0.25
T	0.25	0.25	0.00	0.00	0.00	0.25	0.25	0.25

Figure 4.1 presents the motifs identified by the four programs on the sequence sets in which ETS1 TFs occupy the specific enhancers. The first column is for XPRIME-EM without nucleosome positioning scores incorporation, the second column is for XPRIME-EM with nucleosome positioning scores incorporation, the third column is for XPRIME-GIBBS, and the fourth column is for MEME. Table 4.13 shows the occurrences times of motifs that are corresponding the ones in Figure 4.1.

Table 4.13: Numbers of occurrences of motifs that are identified by the programs on the sequence sets in which ETS1 TFs occupy the specific enhancers.

	XPRIME-EM w/o nucleo	XPRIME-EM w/ nucleo	XPRIME-GIBBS	MEME
motif 1	228	151	3600	79
motif 2	90	87	10080	80
motif 3	308	296	39974	102

As we can see, XPRIME-GIBBS's performance in identifying unknown motifs is poor; in identifying known motifs the posterior PWM is dominated by the prior PWM in XPRIME-GIBBS. All the other three programs have identified the ETS1 TFs with $TGGGA$ in it (the first and the third motifs in the results of XPRIME-EM without nucleosome positioning scores incorporation, the first motif in the results of XPRIME-EM with nucleosome positioning scores incorporation, and the third motif in the results of MEME). In the motifs identified by XPRIME-EM without nucleosome positioning scores incorporation, the first and the third represent similar motifs, but the first is for stronger GGA and the third is for the weaker GGA. After incorporating the nucleosome positioning scores, the motif that was associated with the weaker GGA was removed and the motif that was associated with the stronger GGA was even more conservative. Besides the ETS1 motif, the first motif that

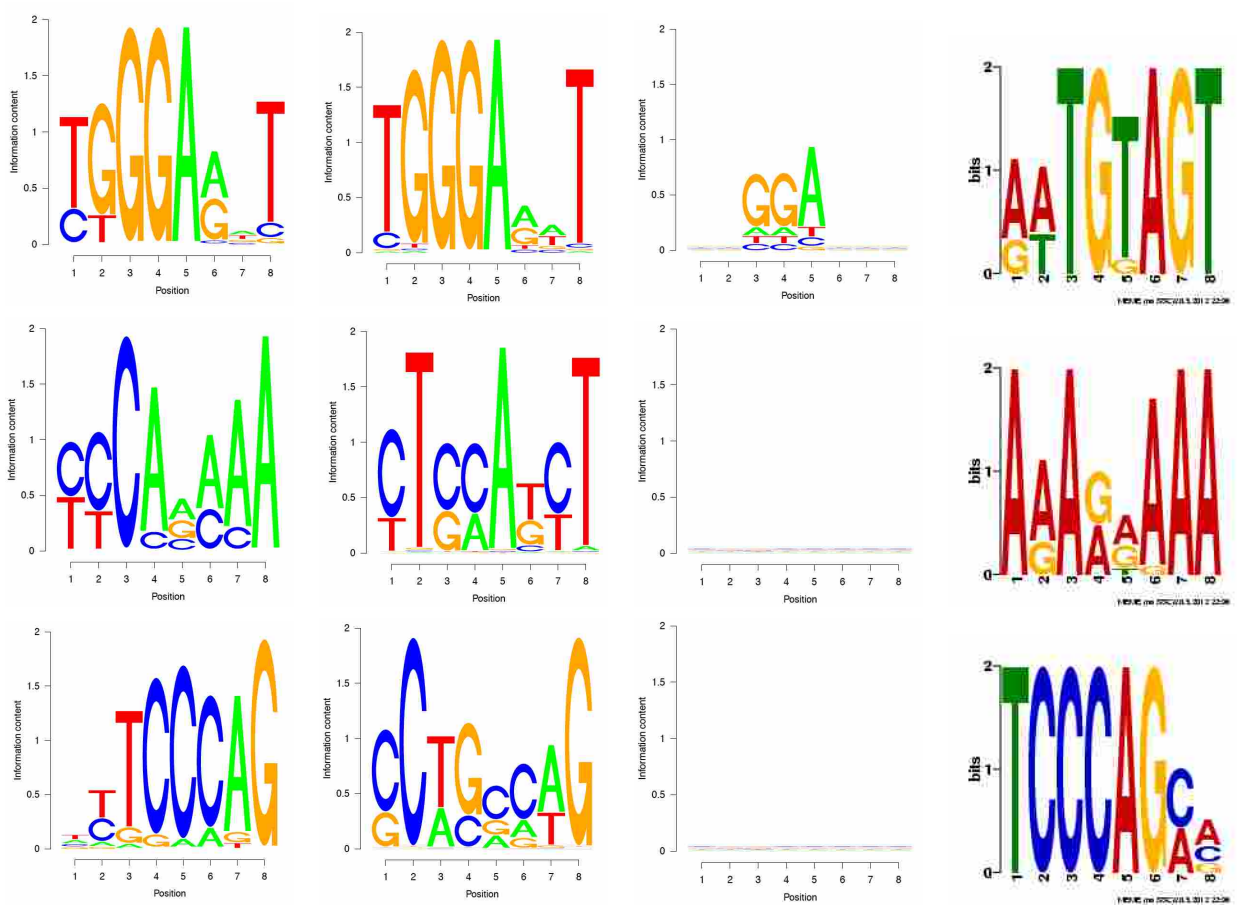

Figure 4.1: Identified motifs by the four programs on the sequence sets in which ETS1 TFs occupy the specific enhancers. The first column is for XPRIME-EM without nucleosome positioning scores incorporation, the second column is for XPRIME-EM with nucleosome positioning scores incorporation, the third column is for XPRIME-GIBBS, and the fourth column is for MEME.

MEME has found was relatively conservative and may be meaningful for ETS1 functioning when ETS occupy the specific enhancers.

Figure 4.2 presents the motifs identified by the four programs on the sequence sets in which ETS1 TFs occupy the redundant promoters. The first column is for XPRIME-EM without nucleosome positioning scores incorporation, the second column is for XPRIME-EM with nucleosome positioning scores incorporation, the third column is for XPRIME-GIBBS, and the fourth column is for MEME. Table 4.14 shows the numbers of occurrences of motifs that are corresponding the ones in Figure 4.2. XPRIME-GIBBS fails to identify both known

and unknown motifs. Both XPRIME-EM with and without nucleosome positioning scores incorporation has identified the ETS1 motif with conservative GGA (the first motifs), while the third motif in MEME results includes GGA in the middle but also contains GGC. The third motif identified by XPRIME-EM without nucleosome positioning scores incorporation is less conservative after incorporating nucleosome positioning scores, which may suggest that this motif tends to locate in the nucleosome occupied region and may relate to the function ETS1 motifs when ETS1 TFs occupy to the redundant promoters.

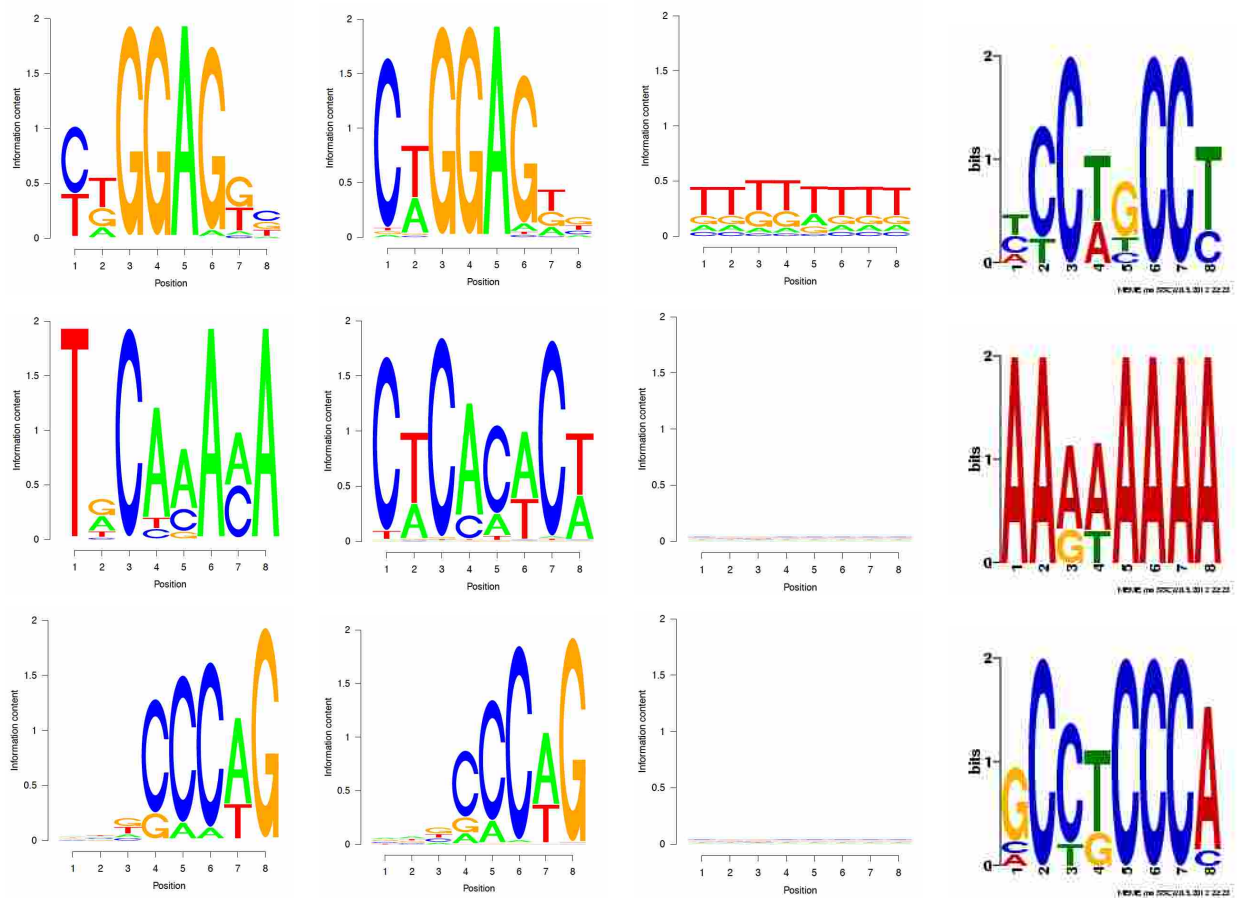

Figure 4.2: Identified motifs by the four programs on the sequence sets in which ETS1 TFs occupy the redundant promoters. The first column is for XPRIME-EM without nucleosome positioning scores incorporation, the second column is for XPRIME-EM with nucleosome positioning scores incorporation, the third column is for XPRIME-GIBBS, and the fourth column is for MEME.

Table 4.14: Numbers of occurrences of motifs that are identified by the programs on the sequence sets in which ETS1 TFs occupy the redundant promoters.

	XPRIME-EM w/o nucleo	XPRIME-EM w/ nucleo	XPRIME-GIBBS	MEME
motif 1	186	100	3643	232
motif 2	117	86	23380	87
motif 3	767	900	26633	70

CHAPTER 5

CONCLUSION

We have developed a DNA motif searching program using a modified expectation-maximization algorithm (XPRIME-EM). Like XPRIME-GIBBS (Poulsen 2009), our method allows to incorporate the expert prior information for motif PWMs, which makes it superior to other motif searching methods. In addition, XPRIME-EM also allows for the incorporation of nucleosome positioning scores in motif searching. The users are able to decide whether or not incorporating the nucleosome positioning scores into the motif searching by using a FASTQ or FASTA sequence file.

XPRIME-EM is superior to XPRIME-GIBBS by not allowing the overlapping of different motifs by the Phase Shift step. The performance of XPRIME-EM is better than the performance of the XPRIME-GIBBS for identifying *de novo* motifs in both simulated data sets and real biological data sets. For identifying known motifs, XPRIME-EM has higher mean precision rates than XPRIME-GIBBS and the two programs have the same mean sensitivity, which is around 100 percent.

For identifying *de novo* motifs, compared to MEME, before incorporating the nucleosome positioning score, XPRIME has lower mean precision rates and mean sensitivity than MEME. However, after incorporating the nucleosome positioning score, XPRIME-EM's mean sensitivity for identifying motifs that locate only in the nucleosome free region is as high as MEME's. If we could improve the performance of XPRIME-EM without the nucleosome positioning score incorporation, after the nucleosome positioning scores incorporation, the performance of XPRIME-EM for identifying motifs that locate only in the nucleosome free region may will be even better than MEME.

Computationally, XPRIME-GIBBS takes close to 24 hours to run in parallel over a node with 8 cores with 10,000 iterations on a set of 100 550bp long simulated sequences for identifying 3 motifs. XPRIME-EM takes close to 15 minutes to run on the same sequences in parallel over a node with a 8 quad-core with 25 iterations. Through the web server $http://meme.sdsc.edu/meme/cgi-bin/meme.cgi$, MEME takes close to 18 minutes to run.

In search for motifs in the real biological data sets (i.e. DNA sequences containing ETS1 motifs), analysis by XPRIME-EM and MEME resulted in some interesting new motifs. Users are encouraged to search for *denovo* motifs in given sequence sets both by MEME and XPRIME-EM with nucleosome positioning score incorporation; results from both programs can be valuable.

Future research will focus on improve the precision and the sensitivity of XPRIME-EM. More specifically, we will work on a way in which scaling the likelihood scores by motif correlation factors does not affect the convergence of the EM algorithm. We expect that after the performance of XPRIME-EM without nucleosome positioning scores incorporation is improved, the performance of XPRIME-EM with nucleosome positioning scores incorporation on identifying motifs locating in nucleosome free regions will offer improvements over MEME.

BIBLIOGRAPHY

Bailey, T. L., and Elkan, C. (1993), "Unsupervised learning of multiple motifs in biopolymers using expectation maximization." *Technical Report CS93-302, Department of Computer Science, University of California, San Diego, August 2003.*

—— (1994), "Fitting a mixture model by expectation maximization to discover motifs in biopolymers." *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology*, 2, 28–36.

—— (1995), "Unsupervised Learning of Multiple Motifs in Biopolymers Using Expectation Maximization," *Machine Learning*, 21, 51–80.

Baldi, P., Chauvin, Y., Hunkapiller, T., and McClure, M. A. (1994), "Hidden Markov models of biological primary sequence information," *Proc. Natl. Acad. Sci. U.S.A.*, 91, 1059–1063.

Bembom, O. (2007), "Sequence logos for DNA sequence alignments," .

Bembom, O., Keles, S., and van der Laan, M. J. (2007), "Supervised detection of conserved motifs in DNA sequences with cosmo," *Stat Appl Genet Mol Biol*, 6, Article8.

Bussemaker, H. J., Li, H., and Siggia, E. D. (2000), "Building a dictionary for genomes: identification of presumptive regulatory sites by statistical analysis," *Proc. Natl. Acad. Sci. U.S.A.*, 97, 10096–10100.

Chen, X., Hughes, T. R., and Morris, Q. (2007), "RankMotif++: a motif-search algorithm that accounts for relative ranks of K-mers in binding transcription factors," *Bioinformatics*, 23, i72–79.

Clement, N. L., Snell, Q., Clement, M. J., Hollenhorst, P. C., Purwar, J., Graves, B. J., Cairns, B. R., and Johnson, W. E. (2010), "The GNUMAP algorithm: unbiased probabilistic mapping of oligonucleotides from next-generation sequencing," *Bioinformatics*, 26, 38–45.

Cock, P. J., Fields, C. J., Goto, N., Heuer, M. L., and Rice, P. M. (2010), "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants," *Nucleic Acids Res.*, 38, 1767–1771.

Conlon, E. M., Liu, X. S., Lieb, J. D., and Liu, J. S. (2003), "Integrating regulatory motif discovery and genome-wide expression analysis," *Proc. Natl. Acad. Sci. U.S.A.*, 100, 3339–3344.

Das, M. K., and Dai, H. K. (2007), "A survey of DNA motif finding algorithms," *BMC Bioinformatics*, 8 Suppl 7, S21.

Galas, D. J., and Schmitz, A. (1978), "DNAse footprinting: a simple method for the detection of protein-DNA binding specificity," *Nucleic Acids Res.*, 5, 3157–3170.

Gupta, M., and Liu, J. S. (2005), "De novo cis-regulatory module elicitation for eukaryotic genomes," *Proc. Natl. Acad. Sci. U.S.A.*, 102, 7079–7084.

Hellman, L. M., and Fried, M. G. (2007), "Electrophoretic mobility shift assay (EMSA) for detecting protein-nucleic acid interactions," *Nat Protoc*, 2, 1849–1861.

Higgins, J. J. (2003), *Introduction to Modern Nonparametric Statistics* (1st ed.), Duxbury Press.

Hollenhorst, P. C., Chandler, K. J., Poulsen, R. L., Johnson, W. E., Speck, N. A., and Graves, B. J. (2009), "DNA specificity determinants associate with distinct transcription factor functions," *PLoS Genet.*, 5, e1000778.

Knezetic, J. A., and Luse, D. S. (1986), "The presence of nucleosomes on a DNA template prevents initiation by RNA polymerase II in vitro," *Cell*, 45, 95–104.

Kornberg, R. D., and Lorch, Y. (1999), "Twenty-five years of the nucleosome, fundamental particle of the eukaryote chromosome," *Cell*, 98, 285–294.

Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F., and Wootton, J. C. (1993), "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment," *Science*, 262, 208–214.

Lawrence, C. E., and Reilly, A. A. (1990), "An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences," *Proteins*, 7, 41–51.

Lee, C. K., Shibata, Y., Rao, B., Strahl, B. D., and Lieb, J. D. (2004), "Evidence for nucleosome depletion at active regulatory regions genome-wide," *Nat. Genet.*, 36, 900–905.

Liu, X., Brutlag, D. L., and Liu, J. S. (2001), "BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes," *Pac Symp Biocomput*, 127–138.

Liu, X. S., Brutlag, D. L., and Liu, J. S. (2002), "An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments," *Nat. Biotechnol.*, 20, 835–839.

Narlikar, L., Gordan, R., and Hartemink, A. J. (2007), "A nucleosome-guided map of transcription factor binding sites in yeast," *PLoS Comput. Biol.*, 3, e215.

Narlikar, L., Gordan, R., Ohler, U., and Hartemink, A. J. (2006), "Informative priors based on transcription factor structural class improve de novo motif discovery," *Bioinformatics*, 22, e384–392.

Poulsen, R. (2009), *XPRIME: A Method Incorporating Expert Prior Information Into Motif Exploration*, Brigham Young University. Department of Statistics.

Roth, F. P., Hughes, J. D., Estep, P. W., and Church, G. M. (1998), "Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation," *Nat. Biotechnol.*, 16, 939–945.

Sandelin, A., Alkema, W., Engstrom, P., Wasserman, W. W., and Lenhard, B. (2004), "JASPAR: an open-access database for eukaryotic transcription factor binding profiles," *Nucleic Acids Res.*, 32, D91–94.

Schones, D. E., Cui, K., Cuddapah, S., Roh, T. Y., Barski, A., Wang, Z., Wei, G., and Zhao, K. (2008), "Dynamic regulation of nucleosome positioning in the human genome," *Cell*, 132, 887–898.

Shim, H., and Keles, S. (2008), "Integrating quantitative information from ChIP-chip experiments into motif finding," *Biostatistics*, 9, 51–65.

Sinha, S., and Tompa, M. (2000), "A statistical method for finding transcription factor binding sites," *Proc Int Conf Intell Syst Mol Biol*, 8, 344–354.

—— (2003), "Performance Comparison of Algorithms for Finding Transcription Factor Binding Sites," *Bioinformatic and Bioengineering, IEEE International Symposium on*, 0, 214.

Tompa, M. (1999), "An exact method for finding short motifs in sequences, with application to the ribosome binding site problem," *Proc Int Conf Intell Syst Mol Biol*, 262–271.

Torlakovic, E. E., Bilalovic, N., Nesland, J. M., Torlakovic, G., and Fl?renes, V. A. (2004), "Ets-1 transcription factor is widely expressed in benign and malignant melanocytes and its expression has no significant association with prognosis," *Mod. Pathol.*, 17, 1400–1406.

Valouev, A., Johnson, S. M., Boyd, S. D., Smith, C. L., Fire, A. Z., and Sidow, A. (2011), "Determinants of nucleosome organization in primary human cells," *Nature*, 474, 516–520.

van Helden, J., Andr., and Collado-Vides, J. (1998), "Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies," *Journal of Molecular Biology*, 281, 827 – 842.

van Helden, J., Rios, A. F., and Collado-Vides, J. (2000), "Discovering regulatory elements in non-coding sequences by analysis of spaced dyads," *Nucleic Acids Res.*, 28, 1808–1818.

Vlieghe, D., Sandelin, A., De Bleser, P. J., Vleminckx, K., Wasserman, W. W., van Roy, F., and Lenhard, B. (2006), "A new generation of JASPAR, the open-access repository for transcription factor binding site profiles," *Nucleic Acids Res.*, 34, D95–97.

Wingender, E. (2008), "The TRANSFAC project as an example of framework technology that supports the analysis of genomic regulation," *Brief. Bioinformatics*, 9, 326–332.

Wingender, E., Dietze, P., Karas, H., and Knuppel, R. (1996), "TRANSFAC: a database on transcription factors and their DNA binding sites," *Nucleic Acids Res.*, 24, 238–241.

APPENDICES

APPENDIX A

CODE OF THE XPRIME-EM ALGORITHM

```
rm(list=ls())
t1=proc.time()
Xprime2=function(seqs,nmotif=5,cnum=9,len=NULL,pmotif=NULL,background=NULL,
it=2,par=NULL,r.prior=NULL,fastq=FALSE){

   if (!is.null(pmotif)){len=ncol(pmotif[[1]])}

  # choose random motifs if not enough in pmotif
  if (length(pmotif)<nmotif){
  set.seed(0)
  for (i in 1:(nmotif-length(pmotif))){
    tmp=NULL
    for (j in 1:len){
      tmp=cbind(tmp,sample(1:4,4))
      }
    pmotif[[length(pmotif)+1]]=tmp/9
    }
  }
### choose random background
    if (is.null(background)){background=sample(c(TRUE,FALSE),
    length(pmotif),replace=TRUE)}
  if (length(pmotif)>nmotif){nmotif=length(pmotif)}
```

```
cat("Searching for", nmotif, "motifs\n")
if (is.null(r.prior)){r.prior=rep(1,nmotif)}

if (!is.null(par)){
  library(snow)
  c1= makeCluster(par)
  #clusterEvalQ(c1, dyn.load("scoreSeq.so"))
  cat("Using", par, "processors in parallel\n")
  lapply1=function(l,f,...){parLapply(c1,l,f,...)}
  apply1=function(x,d,f,...){parApply(c1,x,d,f,...)}
}else{
  lapply1=function(l,f,...){lapply(l,f,...)}
  apply1=function(x,d,f,...){apply(x,d,f,...)}}

## Read in sequences
if(!fastq){tmp=read.fasta(seqs)}else{tmp=read.fastq(seqs)}
title=tmp$title
sequence=tmp$sequence
qual=tmp$qual

seq=unlist(lapply1(sequence,strsplit,NULL),recursive=F)
if (fastq){qual=lapply1(qual,qseqToqual)}

amb=NULL
for (i in 1:length(seq)){
  if (any(seq[[i]]=="N")){amb=c(amb,i)}
}
```

```
  if (length(amb)>0){
    cat("Deleted", length(amb), "sequences because they are ambiguous.\n")
    seq=seq[-amb]
    qual=qual[-amb]
  }
  seqMats=lapply1(seq,seqMat)
   print(length(seqMats))
print(length(qual))
  if(fastq){for (i in 1:length(qual)){

seqMats[[i]]=rbind(seqMats[[i]],qual[[i]])}}

  short=which(unlist(lapply1(seq, length))<=len)
  if (length(short)>0){
    cat("Deleted", length(short), "sequences because they are too short.\n")
    seqMats=seqMats[-short]
  }

  seqLens=unlist(lapply1(seqMats,ncol))

  cat("Found",length(seqMats),"sequences for motif searching.\n")

##### EM Algorithm ####
  cat("Starting EM algorithm\n")

  r=rep(1/nmotif,nmotif)
  scoresf=scoresrc=matrix(0,nrow=sum(seqLens)-length(seqMats)*
```

```
  (len-1),ncol=nmotif)

    tmpMotif=pmotif
##############################
   seqMatsc=seqMats
    seqMatsb=seqMats
   for(y in 1:length(seqMatsb)){
seqMatsb[[y]][5,]=0.5
}
#print(seqMatsb[[1]])
  for (i in 1:it){
    cat("iteration:",i,"\n")

    if(i<=10){seqMats=seqMatsc}else{seqMats=seqMatsb}

    #E-step
    for (j in 1:nmotif){
      scoresf[,j]=unlist(lapply1(seqMats,scoreFunction,tmpMotif[[j]],
      back=background[j]))
      scoresrc[,j]=unlist(lapply1(seqMats,scoreFunctionrc,tmpMotif[[j]],
      back=background[j]))
}
    zi=t(apply(cbind(scoresf,scoresrc),1,ziFunction,rep(r,2)))

zi[,1:7]=phase.shift(zi[,1:7])
zi[,8:14]=phase.shift(zi[,8:14])
```

```
if(i==it){
zi=apply1(as.matrix(zi), 2, remv)
}
    #M step
    zishort=zi[,1:nmotif]+zi[,nmotif+(1:nmotif)]
    r=(apply(zishort,2,sum)+r.prior)/(sum(zishort)+sum(r.prior))

    tmpMotif=pmotif
    tmpMotifF=apply1(as.matrix(zi[,1:nmotif]),2,motifMaxF,seqMats,
    seqLens,len)

    tmpMotifRc=apply1(as.matrix(zi[,nmotif+(1:nmotif)]),2,motifMaxRc,
    seqMats,seqLens,len)

    for (j in 1:nmotif){
      tmpMotif[[j]]=tmpMotif[[j]]+matrix(tmpMotifF[,j],4)+matrix(tmpMotifRc[,j],4)
    tmpMotif[[j]]=tmpMotif[[j]]/sum(tmpMotif[[j]][,1])

      }
     #print(tmpMotif)
}

  if (!is.null(par)){stopCluster(c1)}

   zilist=list();index=0
#zi[,5]=phase.shift(zi[,5])
```

```
  for (i in 1:length(seqMats)){zilist[[i]]=zi[index+(1:(seqLens[i]-len+1)),
  1:nmotif]+zi[index
  +(1:(seqLens[i]-len+1)),nmotif+(1:nmotif)];index=index
  +seqLens[i]-len+1}

  Z=NULL
  for(i in 1:length(seqMats)){
   Z=rbind(Z, zilist[[i]], matrix(0, nrow=7, ncol=ncol(zilist[[i]])))
  }
  Loc=NULL
  for(i in 1:ncol(Z)){
  Loc[[i]]=which(Z[,i]!=0)
  }

  S=NULL
  for(i in 1:length(Loc)){
   S[[i]]=Z[Loc[[i]], i]
  }


  return(list(file=seqs,seqNames=title,seqLengths=seqLens,
  sequences=sequence,
  seqs=seq,seqMats=seqMats,motifProbs=zilist,
  background=background,motProps=r,
  prior.motif=pmotif,posterior.motif=tmpMotif, Loc=Loc, S=S))
}
seqMat=function(DNAseq){
```

```
      if (length(DNAseq)<1){return(NULL)}
      tmp=list("A"=c(1,0,0,0),"C"=c(0,1,0,0),"G"=c(0,0,1,0),
      "T"=c(0,0,0,1),"N"=c(0,0,0,0))
      matrix(unlist(tmp[DNAseq]),nrow=4)
}



scoreFunction=function(seq, PWM, back=FALSE){
  if(nrow(seq)==5){q=seq[5,];seq=seq[1:4,]}else{q=rep(.5,ncol(seq))}
  if(back){q=1-q}
  scores=NULL
  for (i in 1:(ncol(seq)-ncol(PWM)+1)){
     scores=c(scores, prod(q[i:(i+ncol(PWM)-1)]/(1-q[i:(i+ncol(PWM)-1)]))
     *prod(diag(t(PWM)
     %*%seq[,i:(i+ncol(PWM)-1)])))
  }
  scores}

scoreFunctionrc=function(seq, PWM, back=FALSE){
  if(nrow(seq)==5){q=seq[5,];seq=seq[1:4,]}else{q=rep(.5,ncol(seq))}
  if(back){q=1-q}
  scores=NULL
  for (i in 1:(ncol(seq)-ncol(PWM)+1)){
     scores=c(scores, prod(q[ncol(seq):1][i:(i+ncol(PWM)-1)]/(1-q[ncol(seq):1]
     [i:(i+ncol(PWM)-1)]))
     *prod(diag(t(PWM)%*%seq[4:1,ncol(seq):1][,i:(i+ncol(PWM)-1)])))
  }
```

```
  scores[length(scores):1]
  }

phase.shift=function(g){
b=7
m=g[,(-1:-4)]
i=b+1
while(i<=(nrow(m)-b)){
if(m[i,1]==max(m[(i-b):(i+b),])){
# print((m[(i-b):(i+b),]))
# print("no1")
a=m[i,1]
m[(i-b):(i+b),]=0
m[i,1]=a
i=i+b+1
}else if (m[i,2]==max(m[(i-b):(i+b),])){
a=m[i,2]
m[(i-b):(i+b),]=0
m[i,2]=a
i=i+b+1
# print("no2")
}else if (m[i,3]==max(m[(i-b):(i+b),])){
a=m[i,3]
m[(i-b):(i+b),]=0
m[i,3]=a
i=i+b+1
# print("no3")
```

```
}
else{
i=i+1
# print("ok")
}
# cat("i is", i,"\n")
}
#}
for(j in 1:nrow(m)){
if(sum(m[j,]!=0)!=1){
m[j,]=0
}
}
return(cbind(g[,1:4],m))
}



read.fasta=function(seqs){
  cat('Reading FASTA file:',seqs,"\n")
  dna<-readLines(seqs)
  print(dna)
  n<-length(dna)
  sequence<-NULL
  title<-NULL
  for(i in 1:n){
    if(strsplit(dna[i],NULL)[[1]][1]=='>'){
      sequence<-c(sequence,'')
```

```
      title<-c(title,dna[i])
    }else{sequence[length(sequence)]=paste(sequence[length(sequence)],
    dna[i],sep='')}
  }
  return(list("sequence"=sequence,"title"=title,qual=NULL))
}

read.fastq=function(seqs){
  cat('Reading FASTQ file:',seqs,"\n")
  dna<-readLines(seqs)
  n<-length(dna)
  sequence<-NULL
  title<-NULL
  qual <- NULL
  for(i in 1:n){
    if(strsplit(dna[i],NULL)[[1]][1]=='@' | strsplit(dna[i],NULL)[[1]][1]=='+'){
      if(strsplit(dna[i],NULL)[[1]][1]=='@'){
        title<-c(title,dna[i])
        sequence<-c(sequence,'')
        seqqual='s'
      }else{
        qual=c(qual,'')
        seqqual='q'
      }
    }else{
      if(seqqual=="s"){
        sequence[length(sequence)]=paste(sequence[length(sequence)],
```

```
        dna[i],sep='')
      }else{
        qual[length(qual)]=paste(qual[length(qual)],dna[i],sep='')
      }
    }
  }
  return(list(sequence=sequence,title=title,qual=qual))
}

ziFunction=function(scores,r){
(r*scores)/sum(r*scores)
}



motifMaxF=function(zi,seqMats,seqLens,len){
      tmpMotif=matrix(0,4,len)
      for (k in 1:len){
        index=0
        for (m in 1:length(seqMats)){
          tmpMotif[,k]=tmpMotif[,k]+apply(matrix(zi[index+
          (1:(seqLens[m]-len+1))],4,seqLens[m]-len+1,
          byrow=T)*seqMats[[m]][1:4,k:(k+seqLens[m]-len)],1,sum)
          #tmpMotif[,k]=tmpMotif[,k]+apply(matrix(zi[index+
          (1:(seqLens[m]-len+1))],4,seqLens[m]-len+1,
          byrow=T)*seqMats[[m]][4:1,seqLens[m]:1]
          [,k:(k+seqLens[m]-len)],1,sum)
```

```
          index=index+seqLens[m]-len+1
        }
      }
      tmpMotif
    }

motifMaxRc=function(zi,seqMats,seqLens,len){
     zi=zi[length(zi):1]
      tmpMotif=matrix(0,4,len)
      for (k in 1:len){
        index=0
        for (m in length(seqMats):1){
          tmpMotif[,k]=tmpMotif[,k]+apply(matrix(zi[index+
          (1:(seqLens[m]-len+1))],4,seqLens[m]-len+1,byrow=T)
          *seqMats[[m]][4:1,seqLens[m]:1][,k:(k+seqLens[m]-len)],1,sum)
          index=index+seqLens[m]-len+1
        }
      }
      tmpMotif
    }



remv=function(x){
x[which(x<0.5)]=0
return(x)
}
```

```
require(R.oo)
library(R.oo)
qseqToqual=function(q){
  require(R.oo)
  sangerQchr<-q
  sangerQnum<-charToInt(unlist(strsplit(sangerQchr,split=NULL)))
  PHREDQ<-sangerQnum-31
  p<-10^((-PHREDQ)/10)
  p
}

library(MCMCpack)

set.seed(1)
backPWM1=t(rdirichlet(8,c(1,1,1,1)))
backPWM2=t(rdirichlet(8,c(1,1,1,1)))
backPWM3=t(rdirichlet(8,c(1,1,1,1)))
backPWM4=t(rdirichlet(8,c(1,1,1,1)))
backPWM5=t(rdirichlet(8,c(1,1,1,1)))
backPWM6=t(rdirichlet(8,c(1,1,1,1)))
backPWM7=t(rdirichlet(8,c(1,1,1,1)))

cnum=8
setwd("/fslhome/wzhou/compute/Thesis/XPRIME2")

library(seqLogo)
```

```
file='seqfastq115b.txt'
pmotifs=list(backPWM1,backPWM2,backPWM3,backPWM4,
backPWM5,backPWM6,backPWM7)

n=length(pmotifs)
l=length(pmotifs)-4
b1=rep(TRUE,times=4)
b2=rep(FALSE,times=l)
backgrounds=c(b1,b2)
result=Xprime2(file,nmotif=n,pmotif=pmotifs,
background=backgrounds,par=8,it=25,fastq=TRUE)
LOC=result$Loc
save(LOC, file="Loc_EMq_115b.txt")
S=result$S
save(S, file="S_EMq_115b.txt")
Post=result$posterior.motif
save(Post, file="Post_EMq_115b.txt")
print(result)
PWMconvert=function(m){
for(i in 1:ncol(m)){
m[,i]=m[,i]/sum(m[,i])
}
return(m)
}
pdf(file="fastqposter_EM_115b.pdf", onefile=TRUE)
seqLogo(PWMconvert(result$posterior.motif[[5]]))
seqLogo(PWMconvert(result$posterior.motif[[6]]))
```

```
seqLogo(PWMconvert(result$posterior.motif[[7]]))

dev.off()

pdf(file="fastqprior_EM_115b.pdf", onefile=TRUE)

seqLogo(PWMconvert(result$prior.motif[[5]]))
seqLogo(PWMconvert(result$prior.motif[[6]]))
seqLogo(PWMconvert(result$prior.motif[[7]]))
dev.off()
```