A SURVEY OF DEOXYRIBONUCLEIC ACID MOTIF FINDING ALGORITHMS

By

MODAN KUMAR DAS

Bachelor of Science in Agriculture
Bangladesh Agricultural University
Mymensingh, Bangladesh
1979

Master of Science in Genetics and Plant Breeding
Bangladesh Agricultural University
Mymensingh, Bangladesh
1980

Doctor of Philosophy
Oregon State University
Corvallis, Oregon
1990

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2006

A SURVEY OF DEOXYRIBONUCLEIC ACID

MOTIF FINDING ALGORITHMS

Thesis Approved:

Dr. H.K. Dai

Thesis Adviser

Dr. J.P. Chandler

Dr. G.E. Hedrick

Dr. A. Gordon Emslie

Dean of the Graduate College

This thesis is dedicated:

In loving memory to my late parents Susheila and Madhab Das

and

To my beloved wife Shilpi and sons Neil and Raj

ACKNOWLEDGEMENTS

I would like to extend my sincere thanks and gratitude to my advisor, Dr. H.K. Dai, for his invaluable teachings, constant encouragements, patience, support and guidance in the successful completion of this study. Sincere thanks and gratitude are extended to members of my graduate committee Dr. J.P. Chandler and Dr. G.E. Hedrick for their invaluable advice and encouragements.

I am deeply indebted to my wife, Shilpi, who has sacrificed a lot by taking a lot of responsibilities of our family and allowing me to concentrate in my studies during my course work and thesis work. Without her special support, constant encouragements and love my M.S. degree in Computer Science could not be completed.

I would like to specially thank my sons Neil and Raj, their love has given me great inspiration during this study.

Financial support from the Department of Computer Science of Oklahoma State University is gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER I

INTRODUCTION

Unraveling the mechanisms that regulate the expression of genes is a major challenge in biology. An important task in this challenge is to identify regulatory elements, especially the binding sites in deoxyribonucleic acid (DNA) for transcription factors. These binding sites are short DNA segments which are called motifs. Transcription factors are proteins that bind to DNA, typically upstream from and close to the transcription start site of a gene, and regulate the expression of that gene by activating or inhibiting the transcription machinery. Pattern discovery in DNA sequences is one of the most challenging problems in both molecular biology and computer science. In its simplest form, the problem can be formulated as follows: given a set of sequences, find an unknown pattern that occurs in at least q sequences of the set. If a pattern m letters long appears exactly in every sequence, a simple enumeration of all m-letter patterns that appear in the sequences gives the solution. However, things are not that easy with DNA sequences because patterns include mutations, insertions or deletions of nucleotides and usually do not occur exactly. Thus, while approximate occurrences of a single pattern can be still found efficiently, searching for all the 4^m (4 for four nucleotide bases in DNA {adenine (A), cytosine (C), guanine (G), thymine (T)}) possible patterns becomes quickly time consuming and infeasible for large values of m (Staden, 1989; Tompa, 1999), even if we allow only mutations. In recent years, several algorithms have been developed for finding transcription factor binding sites in DNA sequences. In this survey thesis, at first we present some information from biology that will help understanding the motif finding problem, then we present a list of algorithms that have been developed for finding motifs in DNA sequences, we describe some of these algorithms in detail, next we present some results that have been obtained using the software developed from these algorithms, then we present performance comparisons of some of these algorithms and finally we present a discussion on the motif finding algorithms.

1.1. Gene Structure and Expression

Some information on gene structure and expression will be very useful for understanding the motif finding problem. A good source of information on this subject is the book "Molecular Biology of the Cell" by Alberts et al. (2002). Here we present a brief introduction to gene structure and expression.

1.1.1. Structure of gene

The gene is the fundamental unit of inherited information in DNA, and is defined as a section of base sequences that is used as a template for the copying process called transcription. Genes carry the necessary information to encode particular protein structures. Genes comprise only a fraction of all DNA carried on the chromosomes. Although some of the extragenic sequences function as regulatory elements for the control of gene expression, a role has yet to be assigned to the extensive stretches of non-coding DNA sequences in the genomes of higher organisms. It has been conjectured that these sections of the genome participate in the higher order structure of chromosomes, or they may interact with the cytoskeletal components to localize certain regions of DNA to specific nuclear locations. In higher organisms, most protein-coding gene sequences are unexpectedly interrupted by stretches of non-coding sequences, called introns. Intronic sequences often contribute more to the overall length of a gene than do the coding regions, called exons. Regulatory sequences that make up the promoter and include the TATA box occur close to the site where transcription starts. A TATA box is a DNA sequence found in the promoter region of most genes. It is the binding site of either transcription factors or histones and is involved in the process of transcription by RNA polymerase. It has the core DNA sequence 5'-TATAA-3', which is usually followed by three or more adenine bases and has been highly conserved through evolution. The promoter region contains the binding sites for the transcription factor proteins that start up transcription. Moreover, the region upstream of the transcription start contains many binding sites for transcription factors that act as activators and repressors of gene expression (although some transcription factors can bind outside this region). Enhancer sequences are located at variable distances from the gene.

1.1.2. Gene expression

The main idea in gene expression is that every gene contains the information to produce a protein, which is a linear arrangement of amino acids. There are 20 types of amino acids used in proteins. Thus, a sequence of {A, C, G, T} in DNA is converted to a sequence in the amino acid alphabet. Gene expression begins with the binding of multiple protein factors, known as transcription factors, to enhancer and promoter sequences. The mechanism is slightly different for prokaryotic and eukaryotic organisms. Prokaryotes lack a nucleus in their cells and their genome is floating around somewhere in the cell, while eukaryotes have a nucleus in which the genome is contained. The basic mechanism of gene expression can be divided into two stages, transcription and translation. In the first stage, DNA is transcribed to produce RNA (ribonucleic acid) and in the second stage the RNA is translated to produce a protein. The main difference between RNA and DNA is that RNA is usually a single stranded molecule and DNA is double stranded, and the nucleotides present in RNA are {A, C, G, U} where Thymine (T) used in DNA is replaced by Uracil (U). This intermediary RNA is known as messenger RNA (mRNA).

1.1.2.1. Transcription

In gene expression, the most important mechanism for determining whether or not a protein will be produced is the control of transcription initiation (Lodish et al., 2000). Transcription means the assembly of ribonucleotides into a single strand of mRNA. The sequence of this strand of mRNA is dictated by the order of the nucleotides in the part of the gene that is transcribed. The transcription process involves several components including a double stranded DNA, an enzyme RNA polymerase, transcription factors and the four ribonucleoside triphosphates ATP, CTP, GTP and UTP. Transcription process is initiated by the binding of several transcription factors to regulatory sites in the DNA, usually located in the promoter region of the gene. Transcription factor proteins bind each other to form a complex that associates with RNA polymerase. This association enables the binding of RNA polymerase to a specific site in the promoter. Together, the complex of transcription factors and the RNA polymerase unravel the DNA and separate both strands. Subsequently, the RNA polymerase proceeds down on one strand while it builds up a strand of mRNA complementary to the DNA. In this process, ribonucleoside triphosphates are incorporated, with a DNA strand A being matched to a U in the mRNA, C to G, G to C and T to A, producing an mRNA chain. This process continues, with the RNA polymerase shifting along the DNA, until it reaches a terminator site - a region of sequence that leads to the RNA polymerase being thrown off, ending transcription. Later the mRNA is processed, transported out of the nucleus (in eukaryotes) and translated into protein.

1.1.2.2. Translation

The mRNA that has been transcribed from the DNA is converted to a protein, requiring some kind of rules for translating from one alphabet to the other. The form this takes is that every three bases in the transcribed mRNA correspond to a particular amino acid, and it is called the genetic code. These triplets of bases are called codons, and the code is non-overlapping with no space or pause symbols, so a sequence of three bases would correspond to exactly one amino acid and a sequence of six bases would correspond to exactly two amino acids. There are 64 possible values that a codon can assume, 61 of which correspond to an amino acid. The remaining three (UAA, UAG, UGA) codes are for chain termination. Note that for a given sequence of bases, there are three possible ways to divide it into a sequence of trinucleotides, by shifting the boundaries between codons over by one base. These are called reading frames, the frame that corresponds to how the sequence is actually read is called an open reading frame.

The main molecules involved in translation are:

mRNA: contains the information from the DNA to be translated.

tRNA: transfer RNA, which holds the key to the code. Each tRNA associates a specific codon with a specific amino acid, so if the relevant tRNA binds to each codon there will be a linear arrangement of the required amino acid sequence.

Ribosome: a complex enzyme that facilitates the translation process. It moves along the mRNA, binding tRNAs to the appropriate codons and picking off each tRNA's amino acid to add on to the developing amino acid sequence.

1.1.3. Transcription factors

Transcription factors are proteins that bind to regulatory sequences on DNA thereby modifying the rate of transcription of a gene. Some transcription factors bind directly to specific sequences in the DNA (promoters, enhancers), others bind to each other. Most of them bind both to the DNA as well as to other transcription factors. The transcription rate can be positively or negatively affected by the action of transcription factors. When the transcription factor significantly decreases the transcription of a gene, it is called a repressor. On the other hand, if the transcription factor increases transcription it is called an activator. In some cases when only one of the transcription factors binds, there is no activation but the presence of two or more transcription factors activates the transcription of a certain gene.

1.1.4. Promoters

Promoters are the sites on DNA where transcription factor proteins bind. Eukaryotic promoters reside immediately upstream (5' ~30-200 base pairs (bp)) of the coding region of a gene. Key sequences of the promoter structure comprise a variety of binding sites, where RNA polymerase (RNAP), transcription factors, and other regulators of transcription bind to DNA. Promoters function not only to bind RNA polymerase and transcription factors, but also specify the places and times that transcription can occur from that gene. In additon, different promoters have different combinations of factor binding sites. In general, these sites can be moved around or even flipped into the opposite orientation and there will still be promoter activity. Promoters can be basal or core promoter and upstream promoter. The basal promoter is located within 40 bp upstream of the transcription start site, while the upstream promoter may extend over as many as 200 bp farther upstream. Basal promoter contains the TATA box. All protein coding genes contain basal promoter, while the upstream promoter whose structure and associated binding factors differ from gene to gene.

1.2. DNA Motifs

1.2.1. Definition and occurrence of DNA motifs

A DNA motif is a nucleic acid sequence pattern that has, or is conjectured to have, some biological significance. Normally, the pattern is fairly short (5 to 20 bp long) and is known to recur in different genes or several times within a gene (Rombauts et al., 2003). DNA motifs are often associated with structural motifs found in proteins. Motifs can occur on both strands of DNA. Transcription factors indeed bind directly on the double-stranded DNA; therefore, motif detection software should take this fact into account. Sequences could have zero, one, or multiple copies of a motif.

1.2.2. Special types of motifs:

Palindromic Motifs: Palindromic motifs are a special type of transcription factor binding site from a computational point of view. This kind of motif is a subsequence that is exactly the same as its own reverse complement. e.g., CACGTG.

Gapped or Spaced dyads: A second class of special motifs is gapped motifs or spaced dyads. Such a motif consists of two smaller conserved sites separated by a gap or spacer. The spacer occurs in the middle of the motif because the transcription factors bind as a dimmer. This means that the transcription factor is made out of two subunits that have two separate contact points with the DNA sequence. The parts where the transcription factor binds to the DNA are conserved but are typically rather small (3-5 bp). These two contact points are separated by a non-conserved gap or spacer. This gap is mostly of fixed length but might be slightly variable.

1.2.3. The motif finding problem

Given a set of DNA sequences (promoter region), the motif finding problem is the task of detecting overrepresented motifs that are good candidates for being transcription factor binding sites. It is assumed that co-expression of genes frequently arises from transcriptional co-regulation. As co-regulated genes are known to share some similarities in their regulatory mechanism, possibly at transcriptional level, their promoter regions might contain some common motifs that are binding sites for transcriptional regulators. A sensible approach to detect these regulatory elements is to search for statistically overrepresented motifs in the promoter region of such a set of co-expressed genes. A statistically overrepresented motif means a motif that occurs more often than one would expect by chance.

CHAPTER II

MOTIF DISCOVERY ALGORITHMS

Any algorithm whose goal is to discover novel regulatory elements takes as input a set of regulatory regions of genes, many of which are suspected to contain a common regulatory element. There are many possible sources for such co-regulated genes, including expression microarray experiments, gene knockout experiments, and functional classes from the literature.

Currently, there are four major classes of methods for motif discovery algorithms.

1. String-based methods – mostly rely on counting and comparing oligonucleotide frequencies.
2. Probabilistic sequence models – the model parameters are estimated using maximum-likelihood or Bayesian inference. Most of the algorithms based on these methods try to deduce the motifs by considering the regulatory regions of several co-regulated genes from a single genome. These algorithms search for overrepresented motifs in this collection of regulatory regions.
3. Phylogenetic footprinting - an approach of deducing motifs by considering orthologous regulatory regions of a single gene from several species (Tagle et al. 1988). The simple premise underlying phylogenetic footprinting is that selective pressure causes functional elements to evolve at a slower rate than non-functional

sequences. This means that usually well conserved sites among a set of orthologous regulatory regions are excellent candidates for functional regulatory elements or motifs.

4. Algorithms based on phylogenetic footprinting and probabilistic sequence models – this method uses phylogenetic footprinting and probabilistic sequence models in the same algorithm.

In Table 1, we present a list of motif discovery algorithms and then we describe some of the algorithms that are most frequently used for motif discovery.

Table 1. Some motif discovery algorithms.

Algorithm name	Operating principle based on	Reference
Galas et al., 1985	Enumeration	Galas et al., 1985
Mengeritsky and Smith , 1987	Enumeration	Mengeritsky and Smith, 1987
Staden, 1989	Enumeration	Staden, 1989
EM	Expectation Maximization	Lawrence and Reilly, 1990
WordUP	Enumeration	Pesole et al., 1992
Gibbs Sampler	Gibbs	Lawrence et al., 1993
MACAW	Gibbs	Liu, 1994
MEME	Expectation Maximization	Bailey and Elkan, 1995
AlignACE	Gibbs	Roth et al., 1998
Oligo-Analysis	Enumeration	van Helden et al., 1998
Consensus	Weight Matrix	Hertz and Stormo, 1999
Dyad-Analysis	Enumeration	van Helden et al., 2000
Winnower	Graph	Pevzner and Sze, 2000
ANN-Spec	Gibbs	Workman and Stormo, 2000
SMILE	Suffixtree	Marsan and Sagot, 2000

Verbumculus	Suffixtree	Apostolico et al., 2000
MobyDick	Dictionary	Bussemaker et al., 2000
YMF	Enumeration	Sinha and Tompa, 2000
Bioprospector	Gibbs	Liu et al., 2001
Co-Bind	Gibbs	GuhaThakurta and Stormo, 2001
ITB	Enumeration	Kielbasa et al., 2001
Weeder	Enumeration	Pavesi et al., 2001
MotifSampler	Gibbs	Thijs et al., 2001
MITRA	Prefixtree/Graph	Eskin and Pevzner, 2002
Projection	Hashing	Buhler and Tompa, 2002
Footprinter	Dynamic Programming	Blanchette and Tompa (2002)
MOPAC	Enumeration	Ganesh et al., 2003
DMotif	Enumeration	Sinha, 2003
LOGOS	Expectation Maximization	Xing et al., 2004
EC	Genetic	Fogel et al., 2004
GLAM	Gibbs	Frith et al., 2004
Improbizer	Expectation Maximization	Ao et al., 2004
QuickScore	Consensus	Regnier and Denise, 2004
SeSiMCMC	Gibbs	Favorov et al., 2004
PhyloGibbs	Gibbs	Siddharthan et al., 2005
GIMF	Expectation Maximization	Qi, et al., 2005
MaMF	Enumeration	Hon and Jain, 2006

2.1. Algorithms Developed from String-Based Approach

The most intuitive approach to extract a consensus pattern for a binding site or motif is a string-based approach, where typically overrepresentation is measured by exhaustive enumeration of all oligonucleotides. The observed number of occurrences of a given motif is compared with the expected number of occurrences. The expected number of occurrences and the statistical significance of a motif can be estimated in many ways.

2.1.1. Oligo-Analysis

Oligo-Analysis was developed by van Helden et al. (1998). This algorithm is a simple and fast method allowing the isolation of motifs for transcription factors from families of co-regulated genes, with results illustrated in yeast (*Saccharomyces cerevisiae*). Although conceptually simple, their algorithm proved efficient for extracting, from most of the yeast regulatory families analyzed, the upstream regulatory sequences which had been previously found by experimental analysis. Furthermore, putative new regulatory sites were predicted within upstream regions of several regulons. The method is based on the detection of over-represented oligonucleotides. A specificity of this approach is to define the statistical significance of a site based on tables of oligonucleotide frequencies observed in all non-coding sequences from the yeast genome. The authors claim that in contrast with heuristic methods, this oligonucleotide analysis is rigorous and exhaustive. However, its range of detection is limited to relatively simple patterns which include short motifs with highly conserved core.

Methodology used in developing the algorithm:

Constitution of regulatory families. The essential criterion for the constitution of a regulatory family is that all member genes have to show a common regulatory response. Families can be defined as a regulon, i.e., a set of genes controlled by common regulator, or as a stimulon, which is a set of genes whose transcription responds to a common environmental stimulus. When one builds regulatory families, one should avoid including pairs of structurally related upstream sequences which would strongly bias the probabilistic calculation. Another situation to avoid would be a pair of highly similar upstream regions due to a recent duplication event.

Definition of regulatory region limits. In yeast, regulatory elements are found almost exclusively upstream from the promoter. One would be tempted to consider as putative regulatory sequence the region located between the transcription start and the immediate upstream coding sequence. In eukaryotes, and particularly in yeast, the transcription start can hardly be predicted on the sole basis of the sequence. Its position is only reliable in the few experimentally determined promoters. Using the end of the immediate upstream Open Reading Frame (ORF) as the upstream limit of a regulatory region is not very satisfactory either. On the other hand, there is no reason a priori to discard the possibility that a coding sequence would simultaneously exert some regulatory action on a neighbor ORF. In order to determine the optimal size for the upstream region, the authors analyzed the position of the 308 yeast regulatory sites from TRANSFAC (a database for transcription factor) database. The vast majority (99%) of these sites are located within a 800 bp range. Consequently, the authors considered in their analysis the sequence comprised between coordinates –1 and –800 bp upstream from the ORF start.

After having retrieved the set of upstream sequences from the regulatory family, the numbers of occurrences of all oligonucleotides of the selected size are counted. This count takes into consideration multiple occurrences within the same upstream sequence. The number of occurrences of each oligonucleotide across the regulatory family is then compared to its expected value.

Calculation of expected oligonucleotide frequencies. The authors considered several factors for calculating expected oligonucleotide frequencies. First, the yeast genome is characterized by a sensitive bias in favor of A-T versus G-C bases (frequency of A = frequency of T = 0.31; frequency of G = frequency of C = 0.19). Moreover, nucleotide succession is not random, and some oligonucleotides are clearly over-represented, noticeably the poly (A), poly (T) and poly (AT) chains. An additional bias results from the fact that oligonucleotides are differentially represented in coding versus non-coding sequences. A specific expected frequency thus has to be used for each oligonucleotide sequence. One way to calculate the expected oligonucleotide frequencies is to use the frequency observed in the collection of all 800 bp upstream regions from the yeast genome. But there is no reason to systematically restrict the analysis to this precise length, and in fact, the user can select the length of the sequences to analyze. The authors took the biologically defined set of all non-coding sequences from the genome to evaluate the expected frequencies. They built tables showing for each possible oligonucleotide (b), the frequency observed throughout all non-coding segments of the whole yeast genome ($F_{nc}\{b\}$), and this for all sizes between one and nine. These frequencies were then used to estimate the oligonucleotide-specific expected frequencies

($F_e\{b\}$):

$$F_e\{b\} = F_{nc}\{b\}$$

These expected frequencies were used to calculate the number of expected occurrences for each oligonucleotide in the set of upstream sequences from the regulatory family:

$$E(occ\{b\}) = F_e\{b\} \times 2 \times \sum_{i=1}^{S} (L_i - w+1) = F_e\{b\} \times T$$

where E(occ{b}) is the expected number of occurrences for the oligonucleotide b; w is the oligonucleotide length; S is the number of sequences in the set; L_i is the length of the ith sequence of the set and T represents the total number of possible matching positions for a pattern of length w across both strands of the sequence set. The factor 2 stands for the fact that we sum the occurrences on both DNA strands, since in their model the action of regulatory sites is orientation insensitive. Since in their case all upstream sequences have the same length (L), T can be simplified as follows:

$$T = 2 \times S \times (L - w + 1).$$

For the statistical significance test they determine the probabilities of observing exactly n occurrences of the oligonucleotide b and also the probability of observing n or more occurrences of the oligonucleotide b.

Later, van Helden et al. (2000) extended their method to find spaced dyads, motifs consisting of two small conserved boxes separated by a fixed spacer. The spacer can be different for distinct motifs, therefore, the spacer length is systematically varied between 0 and 16. The significance of this type of motif can be computed based on the combined score of the two conserved parts in the input data or based on the estimated complete dyad frequency from a background data set.

The greatest shortcomings of the algorithm of van Helden et al. (1998) is that there are no variations allowed within an oligonucleotide. Tompa (1999) addressed this problem when he proposed an exact method to find short motifs in DNA sequences. The algorithm is described below.

2.1.2. Algorithm by Tompa (1999)

Tompa (1999) applied this algorithm particularly to the ribosome binding site problem. Tompa took into account both the absolute number of occurrences and the background distribution and created a table that, for each k-mer (sequence of length k) s, records the number N_s of sequences containing an occurrence of s, where an occurrence allows for a small, fixed number c of substitution residues in s. Then a reasonable measure of s as a motif would be based on how unlikely it is to have N_s occurrences if the sequences were drawn at random according to the background distribution. The statistical significance test for motif occurrences used and described by Tompa is as follows. Let X be a single random sequence of the specified length L, with residues drawn randomly and independently from the background distribution, or alternatively generated by a Markov chain according to the background dinucleotide distribution. Suppose that p_s is the probability that X contains at least one occurrence of the k-mer s, allowing for c substitutions. Under the reasonable assumption that N sequences are independent, the expected number containing at least one occurrence of s is Np_s, and its standard deviation is $\sqrt{(Np_s(1-p_s))}$. Therefore, the associated z-score is

$$M_s = (N_s - Np_s)/\sqrt{(Np_s(1-p_s))},$$

where M_s is the number of standard deviations by which the observed value N_s exceeds its expectation, and is sometimes called the "z-score", "normal deviate", or deviation in standard units. The random quantity of M_s is asymptotically normally distributed, and normalized to have mean 0 and standard deviation 1, making it suitable for comparing different motifs s. Tompa proposed an efficient algorithm to estimate p_s from a set of background sequences based on a Markov chain.

2.1.3. YMF

YMF (Yeast Motif Finder) algorithm was developed by Sinha and Tompa (2000). YMF is summarized as follows. The inputs to the algorithm are a set of upstream sequences, the number of non-spacer characters in the motifs to be enumerated, and the transition matrix for an order m Markov chain constructed from the full complement of upstream sequences of *Saccharomyces cerevisiae*.

The algorithm first makes a pass over the input sequences, tabulating the number N_s of occurrences of each motif s in either orientation. For each motif s for which $N_s >$ 0, it computes $E(X_s)$ and $\sigma(X_s)$, where X_s is a random variable representing the number of occurrences of the motif s in X, $E(X_s)$ and $\sigma(X_s)$ are its mean and standard deviations, respectively. It then computes the z-score z_s using the following formula: z_s = $(N_s - E(X_s)) / \sigma(X_s)$ and outputs the motifs sorted by z-score.

For a single motif s, the running time to compute z_s is O(c^2k^2), where k is the number of non-spacer characters in s, and c is the number of possible instantiations of R, Y, S, and W symbols in s. Where R represents adenine or guanine, Y represents thymine or cytosine, S represents guanine or cytosine, and W represents adenine or thymine based on the Nomenclature Committee of the International Union of Biochemistry for nomenclature for incompletely specified bases in nucleic acid sequences. Because the number of motifs is exponential in *k*, this enumerative method can be applied only for modest values of *k*. However, the dependence on genome size is linear, so that the method scales very well to large genomes.

Moreover, the O(c^2k^2) time z-score computation does not need to be computed for most of the motifs. A very significant reduction in running time is by the following optimization: the dominant part of the motif's z-score computation is the variance calculation. Also z_s can be bounded by the expression

$$z_s \leq (N_s - E(X_s)) / \sqrt{(E(X_s) - E(X_s)^2)} \,,$$

since $\sigma(X_s)^2 \geq E(X_s) - E(X_s)^2$. Hence, before computing $\sigma(X_s)$, $E(X_s)$, is computed and $z_s \leq (N_s - E(X_s)) / \sqrt{(E(X_s) - E(X_s)^2)}$ is used to examine if it may be worthwhile to go into the variance computation. This expression is compared to the lowest z-score among the top ranking motifs discovered so far. If not, the variance computation for s is aborted, and the next motif is examined. A similar bounding technique is used to optimize the variance computation itself. The authors claim that these two optimizations reduce the running time of the algorithm drastically.

2.2. Algorithms Based on Probabilistic Approach

2.2.1. AlignACE

AlignACE (Aligns Nucleic Acid Conserved Elements) was developed by Roth et al. (1998). It is a Gibbs sampling algorithm that returns a series of motifs as weight matrices that are over-represented in the input set of DNA sequences. In this algorithm, a motif is defined as the characteristic base-frequency patterns of the most information-rich columns of a set of aligned sites. Gibbs sampling algorithm has been used to find motifs in protein sequences by several other research groups (Lawrence et al., 1993; Liu et al., 1995; Neuwald et al., 1995;). AlignACE differs from this method in the following ways.

1. The motif model was changed so that the base frequencies for non-site sequence were fixed according to the source genome (62% A+T in the case of *S. cerevisiae*).
2. Both strands of the input sequence are simultaneously considered at each step of the algorithm and overlapping sites are not allowed even if the sites are on opposite strands.
3. Simultaneous multiple motif searching was replaced by an approach in which single motifs were found and iteratively masked. The masking is done by determining the most-information-rich column in each motif, mapping that column back to the input sequences, and placing a marker at each of those positions. The sampler is then reinitialized to find another motif with the stipulation that no sites contain a masking marker may be resampled. Such sites may, however, be added to any found motif at the end of sampling so that the AlignACE output includes all relevant sites for each output motif. In the case of a

very strong motif, it is possible for the motif to have one of its positions masked and yet still retain enough information in its other positions for a variant of the original motif to be found.

4. The near-optimum sampling method used by AlignACE is different from that used by Neuwald et al. (1995). Since a number of upstream regions in *S. cerevisiae* are nearly identical, AlignACE also includes the option to purge very similar input sequence before sampling. A Smith-Waterman algorithm (Smith and Waterman, 1981) is used to find such sets of repeated sequences, all but one of which are then removed from consideration. The cutoffs used for this are such that at least 60% sequence identity is required for a sequence to be purged.

Scoring. The MAP (maximum *a priori* log likelihood) score is used by AlignACE to judge different motifs sampled during the course of the algorithm. A crude, but useful approximation is given by the formula N log R, where N is the number of aligned sites and R is the degree of over-representation of the motif in the input sequence. In other words, if a site matching a given motif is expected to occur once every kilobase according to background genomic mononucleotide frequencies, and ten sites are observed in 2 kb of input sequence, then R = 5. The general properties of MAP score can be summarized by stating that all of the following lead to higher scores for otherwise similar motifs: (1) greater number of aligned sites; (2) more tightly conserved motifs; (3) less total input sequence; (4) more tightly packed information-rich positions; and (5) enrichment of the motif with nucleotides that are less prevalent in the genome.

2.2.2. Original Gibbs Sampler for motif finding

The Original Gibbs Sampler for Motif Finding was developed by Lawrence et al. (1993). The authors of this algorithm did not apply it to DNA sequence but applied to protein sequence in the original article. Since one of the original assumptions of this algorithm was that there exists at least one instance of a motif in every sequence, the method is sometimes called the "site sampler". Gibbs is a Markov Chain Monte Carlo (MCMC) approach: "Markov-Chain", since the results from every step depends only on the results of the preceding one like in Expectation Maximization (EM). "Monte-Carlo", since the way to select the next step is not deterministic but rather based on sampling, i.e., random-numbers.

The statistical background of MCMC methods is explained in the book by Jun S. Liu (Liu, 2001) and that of Gibbs Sampling in the article (Liu, 1995). In this algorithm it is assumed that we are given a set of N sequences $S_1, \ldots, S_N$ and that we seek within each sequence mutually similar segments of specified width W. The algorithm maintains two evolving data structures. The first is the pattern description, in the form of a probabilistic model of residue frequencies for each position i from 1 to W, and consisting of the variables $q_{i,1}, \ldots, q_{i,20}$. This pattern description is accompanied by an analogous probabilistic description of the "background frequencies" $p_1, \ldots, p_{20}$ with which residues occur in sites not described by the pattern. The second data structure, constituting the alignment, is a set of positions a_k, for k from 1 to N, for the common pattern within the sequences. The objective will be to identify the "best", defined as the most probable, common pattern. This pattern is obtained by locating the alignment that maximizes the ratio of the corresponding pattern probability to background probability.

The algorithm is initialized by choosing random starting positions within the various sequences. It then proceeds through many iterations to execute the following two steps of the Gibbs sampler: (1) *Predictive update step.* One of the N sequences, z, is chosen either at random or in specified order. The pattern description $q_{i,j}$ and background frequencies p_j are then calculated, as described below, from the current positions a_k in all sequences excluding z.

(2) *Sampling step.* Every possible segment of width W within sequence z is considered as a possible instance of the pattern. The probabilities Q_x of generating each segment x according to the current pattern probabilities $q_{i,j}$ are calculated, as are the probabilities P_{x} of generating these segments by the background probabilities p_{j}. The weight $A_x = Q_x/P_x$ is assigned to segment x, and with each segment so weighted, a random one is selected (segment x is chosen with probability $A_x/\sum_j A_j,$ where the sum is taken over all possible segments). Its position then becomes the new a_z. This simple iterative procedure constitutes the basic algorithm. The central idea is that the more accurate the pattern description constructed in step 1, the more accurate determination of its location in step 2, and vice versa. Given random position a_k, in step 2 the pattern description $q_{\mathrm{i,j}}$ will tend to favor no particular segment. Once some correct a_k have been selected by chance, however, the $q_{i,j}$ begin to reflect, even though imperfectly, a pattern extant within other sequences. This process tends to recruit further correct a_k, which in turn improve the discriminating power of the evolving pattern.

An aspect of the algorithm alluded to in step 1 above concerns the calculation of the $q_{i,j}$ from the current set of a_k. For the ith position of the pattern we have $N - 1$ observed amino acids, because sequence z has been excluded; let $c_{i,j}$ be the count of amino acid j in this position. Bayesian statistical analysis suggests that, for the purpose of pattern estimation, these $c_{i,j}$'s should be supplemented with residue-dependent "pseudocounts" b_j to yield pattern probabilities $q_{i,j} = (c_{i,j} + b_j)/(N - 1 + B)$, where B is the sum of the b_j. The p_j are calculated analogously, with the corresponding counts taken over all non-pattern positions.

After normalization, A_x gives the probability that the pattern in sequence z belongs at position x. The algorithm finds the most probable alignment by selecting a set of a_k's that maximizes the product of this ratio. Equivalently, one may maximize F, the sum of the logarithms of these ratios. In the notation developed above, F is given by the formula

$$F = \sum_{i=1}^{W} \sum_{i=1}^{20} c_{i,j} \log \tfrac{q_{i,j}}{p_j}$$

where the $c_{i,j}$ and $q_{i,j}$ are calculated from the complete alignment of the sequences.

2.2.3. MotifSampler

MotifSampler was developed by Thijs et al. (2001). This algorithm is a modification of the original Gibbs Sampling algorithm by Lawrence et al. (1993) described above. A probabilistic framework is used to estimate the expected number of copies of a motif in a sequence. In this algorithm the authors also introduce the use of a higher order background model based on a Markov chain. They describe the incorporation of these modifications in the Gibbs Sampling algorithm to find the parameters and have successfully tested their implementation on different data sets of intergenic sequences.

Finding multiple copies. The basic assumption in the gene expressions studies is that co-expressed genes are co-regulated, but it is expected that only a subset of the co-expressed genes are actually co-regulated. When searching for possible regulatory elements in such a set of sequences, this idea has to be taken into account. It is important to have an algorithm that can distinguish between sequences in which there is motif and the ones in which there is not. In higher organisms, regulatory elements can have several copies to increase the influence of the element in the process of transcriptional regulation. In this algorithm, the probabilistic sequence model is reformulated in such a way that makes it possible to estimate the number of copies of the motif in the sequence. The number of copies of a motif in each sequence is represented by creating a new missing value Q_k, the number of copies of the motif in S_k: Q_k varies between 0 and C_{max}, where C_{max} is a user defined parameter to set the maximal number of copies of motifs in a sequence.

The motif model is defined as follows. The motif is represented by a position probability matrix θ_w:

$$\text{Motif } \theta_w = \left[q_i^b\right]_{4\times w},$$

where w is the fixed length of the motif and q_i^b denotes the probability of position for base b $\in$ {A, C, G, T} and $i \in \{1,2, \ldots, w\}$. The background model is represented by B_m, with m the order of the model. Using Q_k and Bayes' theorem the following equation is written to calculate the probability $\gamma_{k,c}$ of finding c copies of the motif in sequence S_k given the motif and background model: $\gamma_{k,c} = P(Q_k = c|S_k, \theta_w, B_m)$, which describes a discrete probability distribution. The parameters are estimated in each iteration of the algorithm. Finally, the expected number of copies of the motif in sequence S_k is calculated as

$$E(Q_k) = \sum_{c=1}^{C_{\max}} c\gamma_{k,c}.$$

Background model. The second modification is the use of a higher order background model. This background model is developed based on a Markov process of order *m*. This means that the probability of the nucleotide b_l at position *l* in the sequence depends on the m previous bases in the sequence. Such a model is described with a transition matrix. Given a background model of order *m*, B_m, the probability of sequence *S* being generated by the background model can be written as:

$$` P(S|B_m) = P(b_1,...,b_l)\prod_{l=1}^{L} P(b_l|b_{l-1,...,}b_{l-m}).$$

It is important to note that the background model can be either constructed from the original sequence data or from an independent data set. The latter approach is the more sensible one if the independent data set is carefully created.

The algorithm. Both modifications described above have been included in the iterative procedure of the Gibbs sampling algorithm. First the number of copies is sampled according to the distribution Γ. In the next step the motif model is updated based on the current alignment vector and the probability distribution of the motif positions is re-estimated. An alignment vector is then selected by sampling according to this updated distribution Γ. Given the new alignment vector one can re-estimate the distribution Γ. The description of the algorithm is as follows:

1. Select or compute the background model B_m.
2. Compute the probability P_x^o for all segments *x* of length *W* in every sequence.

   Since the background model is fixed, it is not necessary to recalculate these values in each iteration.

3. Initialization of the alignment vector $A = \{A_k|k = 1\ldots N\}$ and the weighting factors

$\Gamma = \{\Gamma_k | k = 1 \ldots N\}$: $A_k = \{a_{k,1},\ldots,a_{k,C})$ and $\Gamma_k = \{\gamma_{k,1},\ldots,\gamma_{k,C}\}$

4. Sample each Q_k from the corresponding distribution Γ_k .

5. For each sequence S_z, $z = 1,\ldots,N$

   (a) Create subsets $\hat{S} = \{\hat{S}_i | i \neq z\}$ and $\hat{A} = \{\hat{A}_i | i \neq z\}$, with $\hat{A}_i = \{a_{i,1},\ldots,a_{i,Qi}\}$

   (b) Calculate θ_w and θ_o based on $\hat{S}$ and $\hat{A}$.

   (c) Assign to each segment x from A_z the weight $W_x = P_x / P_x^o$

   $P_x = P(x|\theta_w)$, $P_x^o = P(x|B_m)$

   (d) Sample new position az from probability distribution W_x.

   (e) Update the distribution Γ_z.

6. Repeat from 4 until convergence is reached.

2.2.4. WEEDER

WEEDER was developed by Pavesi et al. (2001). The authors claim that the WEEDER algorithm as "almost" exact. That is, they start from an exact algorithm based on suffix trees, where the time needed to find the occurrences of a pattern depends on the number of its valid occurrences in the sequences, rather than the size of the sequences themselves. Then instead of reducing the number of patterns to be searched, they speed up the algorithm by narrowing down the set of valid occurrences for each pattern. To achieve this they impose a restriction on the location of mismatches.

Suffix trees. A suffix tree is a data structure that exposes the internal structure of a string in a very deep and meaningful way. A suffix tree T for an n-character string $S = s_1 \ldots s_n$ is a rooted directed tree with exactly n leaves numbered 1 to n. Each internal node, other than the root, has at least two children. Each edge is labeled with a nonempty substring of

S. Two edges leaving the same node cannot have labels beginning with the same character. For any leaf I, the concatenation of the edge labels on the path from the root to leaf I exactly spells the suffix of S starting at position I, that is, it spells out $s_i \ldots s_n$.

The algorithm. Given a set of *k* sequences on the alphabet $\sum$= {A, C, G, T}, find all (M, e) patterns, that is patterns of length *M* that occur with at most *e* mutations in at least *q* sequences of the set. Let us suppose that it finds on the tree the endpoints of paths corresponding to the occurrences of a pattern $p = p_1 \ldots p_m$ in the sequences, that is, all the paths that spell words within distance *e* from *p*, with $m < M$. Also, they have associated with each path the distance (number of mismatches) from *p* of the corresponding substring. If *p* is valid, that is, occurs in at least *q* sequences (at least *q* bit are set in the resulting bit string), they try and expand it by one symbol. For each character $b \in$ {A, C, G, T}, they match *b* against the next symbol on each path. If a path ends just before a node *T* of the tree, they match *b* against the first symbol on each edge leaving *T*. Whenever they encounter a mismatch, they increase the previous error along the path by one. Otherwise, the error remains unchanged. If the new error is greater than *e*, they discard the path. Once all paths have been checked, the surviving ones represent the approximate occurrences of $p' = p1..p_m b$. If *p'* occurs in at least *q* sequences, and is shorter than M, they expand it as well; otherwise, they continue with *p* and the next character in $\sum$. The algorithm starts with the empty pattern from the root of the tree, and recursively expands it. That is, it matches the first symbol on each edge leaving the root against A. If A occurs in at least q sequences, it is expanded to AA. If also AA is valid, they move on to AAA, and so on. If it is not valid, they proceed with C, looking for occurrences of AC. Notice that in this method patterns do not have to occur exactly in the sequences. For every valid pattern, they have to follow at most $N)$ different paths in the tree, where N is the overall length of the k sequences. For every word of length M spelled by a path in the tree (since the tree has N leaves, they are at most N), there are at most $\sum_{i=1}^{e}\binom{M}{i}(|\Sigma|-1)^i \leq |\Sigma|^e M^e$ different patterns within distance e. The overall time complexity of the algorithm is thus $O(|\Sigma|^e M^e kN)$, where the additional k factor is needed to OR the bit strings. The algorithm is therefore exponential in the number of mutations allowed, and no longer in the length of the patterns. The main drawback, however, is that every pattern of length e satisfies the input constraints, since every other pattern of length e found in the tree is a valid occurrence for it. That is, the algorithm has at least 4^e valid patterns of length e to expand, always starting with 4^e paths (or N, if $N < 4^e$). Thus, the method works well only for small values of e.

To apply the algorithm also to longer patterns with higher values of e one could reduce the number of patterns that have to be searched, for example those that occur exactly in the sequences, by checking that at least one path has error zero. The approach these authors chose, however, is different. Instead of reducing the set of patterns that have to be searched, they restrict the number of paths that have to be followed for each pattern. That is, they narrow down the set of valid occurrences.

The authors gave the following example. If one wants to find patterns of length 16 that occur with at most 4 errors. The search for each pattern will start with 4^4 paths. Among these paths, 3^4 will spell words at distance four from the pattern, and are thus very unlikely to lead to a valid occurrence. The idea is to “weed out” all these paths.

They also weed out paths with error three and two, considering only paths with at most one mismatch.

They implemented this method in the algorithm by determining dynamically the error threshold according to the pattern length. They fix an initial error ratio ε. Given a pattern *p*, a path is valid if the distance from *p* of the word spelled by the path is not greater than $\lceil \varepsilon |p| \rceil$, where $|p|$ is the length of the pattern. If in the above example, ε = 0.25, and all paths of length four with error greater than one are eliminated. When one expands *p* by one symbol, the error threshold is set to$\lceil \varepsilon |p| + 1 \rceil$. The result is that, for every pattern *p* = p1…pm, valid occurrences are words $s_{i+1} \dots s_{i+m}$ occurring in the sequences for which:

$$\forall j = \in \{1,\dots,m\} d(p_1 .. p_{1+j}, s_{i+1} .. s_{i+j}) \le \lceil \varepsilon j \rceil$$

where $d(p_1 .. p_{1+j}, s_{i+1} .. s_{i+j})$ is the number of mismatches between $p_1 .. p_{1+j}$ and $s_{i+1} .. s_{i+j}$.

That is, $s_{i+1} .. s_{i+m}$ is a valid occurrence for *p* if it is a valid occurrence for all its prefixes $\{p_1, p_1 p_2, \dots, p_1 p_2 \dots p_{m-1}\}$. In other words, we can see *p* as composed of $\lceil \varepsilon m \rceil$ blocks. The *i*-th block starts at position $\lceil \frac{1}{\varepsilon} . (i-1) \rceil$of the pattern. Every occurrence of *p* must present at most one mismatch in the first block, at most two mismatches in the first two blocks, and so on. Now, the maximum number of paths that might correspond to a pattern of length m in the tree is $O(\lceil 1/\varepsilon \rceil^{\varepsilon m} |\Sigma|^{\varepsilon m})$. The time complexity is thus reduced to $O(\lceil 1/\varepsilon \rceil^{e} |\Sigma|^{e} kN)$, where *M* is the length of the longest pattern found, and *e* = $\lceil \varepsilon m \rceil$. Since the error threshold is now determined dynamically, one no longer have to provide the algorithm with exact values for the pattern length and the maximum number of mutations.

2.2.5. ANN-Spec

ANN-Spec was developed by Workman and Stormo (2000). ANN-Spec is a machine learning algorithm. This algorithm makes use of Artificial Neural Network and a Gibbs sampling method to define the Specificity of a DNA-binding protein. ANN-Spec searches for the parameters of a simple network (or weight matrix) that will maximize the specificity for binding sequences of a positive set compared to a background sequence set.

The neural network used in this method is a sparsely encoded perceptron with one processing unit. Perceptrons are linear discriminant functions which can be used to estimate posterior probability distribution. The weights of the perceptron are initialized to represent a randomly sampled site in the sequences. Then the following steps are repeated until a fixed number of iterations is reached:

Motif Sampling Step. The weighted sum of the perceptron's inputs for a substring j, h_j, could be read as a binding energy or probability for fixation or collision of two molecules, namely the transcription factor and the particular sequence j. Thus, the probability h_j is not used directly as in Gibbs sampling. Instead, the $\exp(h_j)$ is used. This results in a weight for every substring from which k sites are sampled.

Weight Update Step. The objective is the log-likelihood of the selected sites. A weight e is applied and specified as:

$$\Delta\Omega = \eta(\partial U^*/\partial\Omega) - \lambda\Omega,$$

where Ω is the weight vector, η is the learning rate or step size, λ is the decay rate and U^* is the objective. Therefore, the change is the log-likelihood from the sampled sites (the new weight) divided by the old weight which is corrected by the learning rate and decay rate. The training time scales linearly with the input data N.

2.2.6. MITRA

MITRA (Mismatch Tree Algorithm) was developed by Eskin and Pevzner (2002). This algorithm uses a mismatch tree data structure to split the space of all possible patterns into disjoint subspaces that start with a given prefix. By splitting the pattern space, MITRA keeps reducing the pattern discovery into smaller sub-problems similarly to the SPELLER algorithm (Sagot, 1998). MITRA also takes advantage of the pairwise similarity between instances. These similarities can be used to construct a graph where each vertex is an l-mer in the sample and there is an edge if the two l-mers are similar (e.g., differ in no more than $2d$ positions). An $(l, d) - k$ pattern will correspond to a clique of size k in this graph. This type of approach is the basis of the WINNOWER algorithm (Pevzner and Sze, 2000). In fact, the authors show that they can impose stronger conditions on the graph for the existence of a pattern than simply a clique of size k.

Splitting pattern space. MITRA splits the space of all possible patterns into disjoint subspaces corresponding to patterns that start with a given prefix. A pattern is called weak if it has less than k (l, d)-neighbors in the sample. A subspace is called weak if all patterns in this subspace are weak. For most of the subspaces, one can quickly conclude that they are weak and save time by not searching the subspaces exhaustively. For example, if one is looking for patterns of length l we would first split the space of all l-mers into 4 disjoint subspaces. The first subspace would be the space of all l-mers starting with A, the second subspace would be the space of all *l*-mers starting with C, etc. The authors further employ strategies for determining whether the subspace contains a $(l, d) - k$ pattern. If it can be ruled out that a subspace contains such a pattern, one stops searching in this subspace and release the memory slot. If one cannot rule out that a subspace contains such a pattern, this subspace is again splitted on the next symbol and the process is repeated. The key ingredient of MITRA is the method to rule out whether a subspace contains a $(l, d) - k$ pattern.

Mismatch tree data structure. Mismatch trees are similar to the suffix trees and tries that have a long history of applications to string matching problems (Gusfield, 1997). The paths from the root to the leaves in a mismatch tree represent not only the substrings in the data (like in suffix trees and tries), but also all neighbors of these substrings with upto *k* mismatches. The data structure is a variation of the sparse prediction trees from Eskin et al. (2000). A mismatch tree is a rooted tree where each internal node has 4 branches, each labeled with a symbol in {A, C, G, T}. The maximum depth of the tree is *l*. Each node in the mismatch tree corresponds to the subspace of patterns ρ with a fixed prefix (defined by the path from the root to the node) and contains pointers to all *l*-mers instances from the sample that are within d mismatches from a pattern $P \in \rho$ (valid *l*-mers). The tree is initialized to contain only a root node and is explored in a depth first fashion over the course of the algorithm.

MITRA starts with examining the root node of the mismatch that corresponds to the space of all patterns. When examining a node, MITRA tries to prove that it corresponds to a weak subspace. If one can not prove it, the node's children are expanded and each of them is examined. This corresponds to splitting the pattern subspace into 4 separate parts. Whenever, one reaches a node corresponding to a weak subspace, it is necessary to backtrack, thus effectively eliminating the subtree rooted at that node from the search. The intuition is that many of the nodes correspond to weak subspaces and can be ruled out. This allows avoiding searching much of the pattern space that would be searched in Sample Driven Approach (SDA). If one reaches depth l, the l-mer corresponding to the path from the root to the leaf corresponds to an $(l, d) - k$ pattern and the pointers from this node correspond to the instances of this pattern. In practice, it is not need to explicitly maintain the mismatch tree in memory since one 'virtually' traverses the mismatch tree in the depth first fashion.

MITRA keeps track of all valid l-mers at each node in the tree (i.e., instances of patterns from the subspace of patterns that correspond to the node). An l-mer is valid for a node if its prefix matches the prefix of the node with at most d mismatches. The set of valid l-mers for a node is a subset of the set of valid l-mers for the parent of the node. MITRA efficiently generates the set of valid l-mers for a node by keeping track of the number of mismatches between each valid l-mer and the prefix of the node. For a valid l-mer in the parent of a node, there are two cases. Either the position corresponding to the branch to the child matches the l-mer, or the position corresponding to the branch to the child does not match the l-mer. In the first case, the l-mer is still valid for the child. In the second case, the count of mismatches for that l-mer increases. If the mismatch count exceeds the threshold d, the l-mer is not passed on to the child. Thus a child node's set of valid l-mer is simply the set of valid l-mers of the parent that either match the label of the branch to the child or are still within an acceptable number of mismatches of the prefix.

The algorithm. At first examine the root node that corresponds to the set ρ of all 4^l l-mers of length l. This node points to all l-mers in the sample. Then examine the first child, A. This child points to all of the l-mers in the sample that have prefix A (with 0 mismatches) and to all of the l-mers in the sample that have a different prefix (with 1 mismatch). Continue with a depth first search and test every node to see if it corresponds to a weak subspace. If yes, backtrack since there is no $(l, d) - k$ pattern in this subspace. If depth l is reached, then the node corresponds to an $(l, d) - k$ pattern. Then compute the score of the pattern and output the pattern along with the score if it is above some threshold that is considered interesting. Since we are finished with this pattern, backtrack in the tree, collapse the current node, and expand the next node. Since the only expanded nodes are along the current search path, there is a maximum of l stored nodes in the tree (counting the root node) which bounds the memory usage of the algorithm. Unlike in the SDA algorithm, one does not need to keep all of the patterns in a large table.

2.2.7. BioProspector

BioProspector was developed by Liu et al. (2001). It is an algorithm for finding sequence motifs from a set of DNA sequences. It takes the following input parameters:

1. A file with N DNA sequences in which the motifs are to be found.
2. A file containing sequences or probabilities characterizing the background nucleotide distribution.
3. The widths of the two motif blocks w_1 and w_2, and their gap range, $[g_L, g_M]$. In the case when a one-block motif is of interest, one can set w_2, g_L and g_M to 0.
4. Whether each sequence has at least one copy of the motif.

5. Whether the motif could occur in both DNA strands.
6. Whether the motif has a palindromic pattern, in which case w_1 must be equal to w_2, and BioProspector checks both DNA strands automatically.

At the end, BioProspector outputs the following results:

1. The motif score, significance value, and the number of aligned segments.
2. A regular expression of the motif consensus and degenerate, as well as a probability matrix expression of the motif.
3. The number of segments each input sequence contributes to the motif, the starting position and sequence of each segment.

Bioprospector uses a Gibbs sampling strategy. It examines the upstream region of genes in the same gene expression pattern group and looks for regulatory sequence motifs. It differs from the original Gibbs sampler in the following points:

1. It uses zero to third-order Markov background models whose parameters are either given by the users or estimated from a specified sequence file.
2. The significance of each motif is judged based on a motif score distribution estimated by a Monte Carlo method.
3. It allows for the modeling of gapped motifs and motifs with palindromic patterns.

2.3. Algorithm Based on Phylogenetic Footprinting

In this category we describe an algorithm named Footprinter, which was developed by Blanchette and Tompa (2002). The basic method of this algorithm is dynamic programming. The inputs to the algorithm are n homologous sequences S_1, S_2, … ,S_n, the phylogenetic tree T relating them, the length k of the motifs sought, and the maximum parsimony score d allowed. The algorithm proceeds from the leaves of T to its root. At each node u of T, it computes a table W_u containing 4^k entries, one for each possible k-mer. For each such k-mer s, let $W_u[s]$ be the best parsimony score that can be achieved for the subtree of T rooted at u, if the ancestral sequence at u was forced to be s. Let $C(u)$ denote the set of children of u, and $h(s, t)$ denote the number of positions at which k-mers s and t differ, and let $\sum = \{A, C, G, T\}$. The table W_u is computed according to the following recurrence:

$$W_u[s] = 0, \text{ if } u \text{ is a leaf and s is a substring of } S_u.$$

$$W_u[s] = +\infty, \text{ if } u \text{ is a leaf and } s \text{ is not a substring of } S_u.$$

$$W_u[s] = \sum_{v \in C(u)} \min_{t \in \sum^k} \; W_v[t] + h(s, t), \text{ if } u \text{ is not a leaf.}$$

A straightforward implementation of this recurrence computes all W tables in time $O(nk(4^{2k} + l))$, where l is the average length of the input sequences $S_1, S_2, \ldots, S_n$. The main term $nk4^{2k}$ in this expression comes from the fact that for each of the $O(n)$ edges (u,v) of T, for each of 4^k possible values of s labeling u, and for each of the 4^k values of t labeling v, the recurrence calls for the computation of $h(s, t)$.

If r is the root of T, each entry of W_r that is at most d gives rise to one or more solutions to be reported. For each such entry, the corresponding k-mers of the n input sequences can be recovered by retracing the recurrence from the root back to the leaves. By maintaining appropriate pointers that reflect the computation of the W tables, the set of solutions can be recovered in time linear in its size. In non-repititive biological sequences the number of solutions is usually small (when d is small), and the time to enumerate them is negligible compared to the time to compute the W tables.

The 4^{2k} factor in the complexity of the algorithm as described makes it impractical to use for most interesting values of k. Blanchette et al. (2002) show how various algorithmic optimizations can reduce the running time of this algorithm to $O(nk \min(l(3k)^{d/2}, 4^k + l))$. It can be noticed that the running time is proportional to nl, which is the total length of all the input sequences. This means that the algorithm's performance scales well as the number of species or length of all motifs provided is increased. Although the running time is exponential in either $d/2$ or k (depending on which of $l(3k)^{d/2}$, $4^k + l$ is the lesser), in practice both of these parameters are quite small: typical values in the experiments reported by the authors were $k = 10$ and $d = 3$.

2.4. Algorithm Based on Phylogenetic Footprinting and Gibbs Sampling Strategies

In this category we describe an algorithm named PhyloGibbs, developed by Siddharthan et al. (2005). This algorithm combines the motif finding strategies of phylogenetic footprinting and Gibbs sampling into one integrated Bayesian framework. PhyloGibbs runs on arbitrary collections of multiple local sequence alignments of orthologous sequences. The algorithm searches over all arrangements in which an arbitrary number of binding sites for an arbitrary number of transcription factors can be assigned to the multiple sequence alignments. These binding site configurations are scored by a Bayesian probabilistic model that treats aligned sequences by a model for the evolution of binding sites and background intergenic DNA. This model takes the phylogenetic relationship between the species in the alignment explicitly into account. The algorithm uses simulated annealing and Monte Carlo Markov chain sampling to rigorously assign posterior probabilities to all binding sites that it reports.

CHAPTER III

RESULTS

Here we present some results that have been obtained in the literature using some of the algorithms described above.

3.1. Oligo-Analysis

van Helden et al. (1998) tested their Oligo-Analysis algorithm for genes controlling metabolism in yeast. They selected these genes because yeast metabolism has been widely studied and provides numerous examples of known regulons. In many cases, the transcriptional factor involved in the common response as well as its binding site is known. These families of co-regulated genes provide ideal datasets to calibrate the method, which, in a further step, could be extended to families whose regulatory elements are unknown. The authors built several families on the basis of the co-regulation of genes. They did not take in consideration the content of the upstream regions of these genes during developing these groups. The important findings of their analyses are as follows.

Clusters of overlapping hexanucleotides generally reveal wider regulatory sequences. For each gene family, they extracted the set of 800 bp upstream sequences, and performed an hexanucleotide analysis. All hexanucleotides with a significance coefficient higher than zero were retained. With the chosen threshold, very few sequences were retained, which was about ten per family, out of 2080 possible hexanucleotide pairs. Highly significant patterns generally appear clustered with a few additional overlapping hexanucleotides that have a weaker significance coefficient. For example, the most salient hexanucleotide of the MET family was CACGTG (significance coefficient = 7.0), was grouped with two strongly overlapping sequences: TCACGT (significance coefficient = 6.1) and GTCACG (significance coefficient = 0.7). In most families, the overlapping clusters reflect the fact that the recognition domain of the transcriptional factor is wider than six nucleotides. The maximum significance indicates the most conserved core that usually corresponds to the bases directly interacting with the transcriptional factor.

Oligonucleotide size. In most families, the simple hexanucleotide analysis of the 800 bp upstream regions allowed them to detect the regulatory sequences previously found by means of experimental analysis. Analysis performed with different oligonucleotide sizes generally revealed the same patterns with different significance indices. A higher statistical significance only indicates a stronger over-representation, which does not necessarily correspond to a functional requirement on size.

Multiple clusters revealed either multiple sites or single site variability. Several clusters generally appear from each family. In some cases, multiple clusters correspond to distinct regulatory sites. This is clearly the case in the MET family, where two independent clusters are detected, the former forming the pattern GTCACGTG, corresponding to the binding site for the transcription factor Gbf1p-Met4p-Met28p complex, and the second revealing the site AAAACTGTGG, recognized by Met31p and Met32p. Alternatively, clusters can be structurally related, and represent variants of the same binding site, as in

PHO family, where one cluster forms the pattern GCACGTGGG, shown to bind Pho4p with high affinity, and another cluster defines the low affinity consensus GCACGTTTT for this same transcription factor.

Unknown sites revealed by oligonucleotide analysis. Some additional hexanucleotides were identified within each family besides those belonging to known regulatory sequences. Based on the results for known sites, one can infer that the ideal unknown site should appear as a cluster of overlapping hexanucleotides with a core showing a high significance coefficient. They observed several unknown patterns extracted from the hexanucleotide analysis fit with these criteria, and are good candidates as new regulatory sequences.

3.2. YMF

Sinha and Tompa (2002) discussed the results of validation experiments in which YMF was used to identify candidate binding sites in 23 well studied regulons of *S. cerevisiae*.

For 18 of these regulons YMF succeeded in reporting the known binding site consensus for the regulon's principal transcription factor. The study was carried out using the SCPD (The Promoter Database of *Saccharomyces cerevisiae*) database (Zhu and Zhang, 1999). The SCPD database has a collection of transcription factors and the genes regulated by each factor. Each such set of genes comprises a regulon. For each gene in a regulon, the database lists the experimentally determined binding sites of the transcription factor, and in many cases the consensus sequence of the binding sites in the regulon is also given. The success of YMF was assessed by comparing the top motifs reported with the known consensus for the regulon. The program was run three times on each regulon, to find motifs of length 6, 7, and 8, respectively. For length 6 motifs, a maximum of 11 spacers in the middle was allowed. For lengths 7 and 8, the motif model did not include spacers. In all runs, a maximum of 2 degenerate symbols (R, Y, S, or W) was allowed in the candidate motifs.

For 15 of the 23 regulons, the top motif reported (for one or more value of the motif length parameter) was a match. For 14 of the 15 regulons, there was a match with P_{max} less than 0.1, the exception being MATa2. In another regulon, MCM1, the top ranking motifs (for length 6) were variants of the poly-A element (any motif that can be instantiated to a string of all A's, e.g., AAAAWAAA), and the first non-polyA motif, at rank 11 with $P_{max} = 0.01$, was CCSNNNNAGG, similar to the known consensus CCNNNWWRGG. For the regulon RAP1, the top motif reported (for length 7) is GCAYGTG, which matches part of the Inositol/Choline Response Element (ICRE) with consensus SCAYRTGAARW. The 23 regulons represent the typical input for a motif-finder – they are of varying sizes (3 to 38 genes) and have a variety of known binding sites (length 5 to 10, with few to many spacers or degenerate symbols). The results thus demonstrate the applicability of the method on a variety of data sets. In most cases, a match was found in the top three motifs for multiple values of l, indicating that the performance is not crucially dependent on prior knowledge of the motif length. In some cases, YMF found a match even though the known consensus of the binding site does not conform to the motif model YMF uses. For instance, the regulon SCB has the sequence CNCGAAA as its binding site consensus, with an 'N' that is not in the middle. Nevertheless, a very similar motif CACGAAA was reported. Similarly, for HAP1 (consensus CGGNNNTANCGG), the motif SGGNNNNNNSGG was discovered.

The regulon ABF1 is an example of a case where multiple occurrences of the binding site are found in the same promoter region. Of the 19 genes in this regulon, 8 have two or more occurrences of the motif TCRNNNNNNACG in their promoter region. There are a total 36 occurrences of the motif, giving it a very high z-score of 10.07. If each of the 19 genes had only one occurrence of the motif, for a total of 19 occurrences, the z-score would have been about 4.03, which is rather low, meaning that the motif would not have been reported as significant.

In addition to their validation experiments, they also used YMF for gene families in the functional and mutant phenotype catalogues of *S. cerevisiae* from the MIPS database, where YMF reported many promising novel transcription factor binding sites.

3.3. AlignACE

When used to search upstream of apparently coregulated genes, AlignACE finds motifs that often correspond to the DNA binding preferences of transcription factors. Roth et al. (1998) used AlignACE to analyze whole genome mRNA expression data. They applied it to three extensively studied regulatory systems in *Saccharomyces cerevisiae*: galactose response, heat shock, and mating type. Galactose response data yielded the known binding site Gal4, and six of nine genes known to be induced by galactose. Heat shock data yielded the cell-cycle activation motif, which is known to mediate cell-cycle dependent activation, and a set of genes coding for all four nucleosomal proteins. Mating type α and a data yielded all of the four relevant DNA motifs and most of the known a- and α-specific genes.

Hughes et al. (2000) presented a more detailed study of the effectiveness of AlignACE as applied to a variety of groups of genes in *Saccharomyces cerevisiae* genome. Details of the results are as follows.

The input sets of genes. A total of 248 groups of genes were examined, including 135 from the database at the Munich Information Center for Protein Sequences, 17 groups from the Yeast Protein Database, and 96 groups from *Saccharomyces* Genome Database. They considered only groups of six or more genes. The number of genes in each of these groups ranged from minimum of six to as many as 707, with an average of 42 genes per group. Runs of AlignACE on the upstream regions of these groups of genes produced 3311 motifs.

Motif measures. To reduce the set of 3311 motifs under consideration, they devised two motif measures: one related to group specificity, the other to positional bias. The group specificity score gauges how well a given motif targets the upstream regions of the genes used to find it relative to the upstream regions of all the genes in the genome. The positional bias score indicates the degree to which a motif tends to be preferentially positioned in a particular distance range upstream of the translation start.

Motif clustering. AlignACE generated many examples of identical or very similar motifs. This occurs when the same motif is found from AlignACE runs on overlapping or related groups of ORFs and also when multiple similar examples of a very strong motif are returned from a single AlignACE run. To automatically group very similar motifs together, they used a hierarchical clustering technique based on an algorithm which they named CompareACE.

Highly group-specific motifs. In order to present only the strongest of the great number of motifs found, they chose a MAP score cutoff of 10.0, which reduced the set of motifs under consideration to 1234. While largely arbitrary, this threshold did not lead to the rejection of any of the best examples of known *cis*-regulatory elements. To focus on the most selective motifs, they chose a cutoff of 10^{-10} for the group specificity score. A total of 54 highly specific motifs fulfilled both criteria and were grouped into 25 distinct motif clusters.

Known motifs. Assignment of AlignACE motifs to known *cis*-regulatory elements from the literature is an ideal application for the algorithm CompareACE. However, the authors did not use this algorithm, because databases of known transcription factor binding sites are still incomplete. The main criterion used to identify an AlignACE motif as a known *cis*-regulatory element was that the AlignACE motif matched the literature consensus and was found upstream of an appropriate set of genes. For motifs with numerous annotated, well-defined binding sites, this criterion allowed them to easily make the assignment. In cases involving very few known sites, the criterion used was whether the top genomic sites for the AlignACE motif included a significant fraction of the sites verified in the literature. They were able to identify the following 16 known motifs from among the 25 highly specific motif clusters: Rap1p, Gcn4p, the heat shock element (HSE), the Cbf1p-Met4p-Met28p complex, the Hap2p-Hap3p-Hap4p complex, Lys14p, the MluI cell-cycle box (MCB), the stress response element (STRE), the Met21p-Met32p complex, Leu3p, Oaf1p, the carbon source responsive element (CSRE), Pho4p, Ste12p, and Pdr3p.

Unknown motifs. In addition to the known motifs, the authors were able to find many unknown motifs using AlignACE. Three highly specific motifs were found to be associated with ribosomal proteins. One of these, the Rap1p motif, is well known. The other two motifs are primarily associated with small and large ribosomal subunits. These findings are especially interesting since the transcriptional regulation of a number of ribosomal proteins has been studied in detail, and the known Rap1p and Abf1p sites, along with a T-rich region, are generally found to be sufficient to explain their transcriptional control (Goncalves et al., 1995).

3.4. MotifSampler

G-box sequences. To validate the motif sampler the authors constructed two data sets: one with a known regulatory element involved in light regulation in plants, G-box and one of so-called random sequences in which no G-box is reported. The G-box data set consists of 33 sequences selected from PlantCARE (Rombauts et al., 1999) containing 500 bp upstream of the translation start. This data set is well suited to give a proof of concept and to test the performance of the motif sampler, since the consensus of the motif and also the positions of the motif in the sequences are known. The random set consisted of 87 sequences of 500 bp. This set was used to introduce noise to the test set. The authors show the results of two different tests. The first test shows the influence of the number of copies and the second test illustrates the improvement due to the use of a higher-order background model when noise is added to the data set.

First they experimented with 33 G-box sequences together with 10 sequences from the random set. This set was used to test the influence of the number of copies. When the number of copies is set to 1, a more conserved motif will be found, but a number of occurrences will be missed. Increasing the number of copies will allow to better locate the true number of copies of a motif but more noise is introduced to the initial model and the final model will be more degenerate. The results shown were based on parameter settings to search for a motif of length 8 bp that can have either 1 or 4 copies using third-order background model with *Arabidopsis*. In both cases a motif was found with a consensus resembling the G-box consensus CACGTG. These motifs are also the motifs with the highest scores.

Secondly, they tested the influence of noise on the performance of the motif sampler. Noise is due to the presence of upstream sequences that do not contain the motif. To introduce noise in the data set they added in several consecutive tests each time 10 extra sequences, in which no G-box is reported, to the G-box data set. They tested several configurations to see how the noise influences the performance of the motif sampler. They used three different background models. The number of times the G-box was detected decreased when more noise was added to the original set of 33 G-box sequences. This influence was more dramatic for the single nucleotide background model than the third-order background model.

Microarray experiments. They also used the motif sampler to find motifs in clusters of co-expressed genes identified from microarray experiment. As a test case, they used the data from Reymond et al. (2000), where the gene expression in response to mechanical wounding was measured. Messanger RNA (mRNA) was extracted from leaves at 8 time points up to 24 hours after the wounding and an expression profile was constructed. To find the groups of co-expressed genes they used a clustering algorithm. They found 8 small clusters of co-expressed genes. To analyze the clusters they selected the sequence 500 bp upstream of translation start for every gene present in one of the clusters. They looked for 10 different motifs of length 8 and 12 bp. To distinguish between stable motifs and motifs that are found just by chance, they repeated each experiment 10 times. The results using the third-order background model using *Arabidopsis* gave the most promising results. Four of the clusters contained only 3 genes and they did not find any interesting motif in these small clusters. The most interesting motifs were only detected in the clusters containing more than 3 genes. Table 2 gives an overview of the most important results. The consensus sequences are a compilation of the consensus sequences of length 8 and 12 bp. Only the relevant part of the consensus is displayed. Together with the consensus the number of times the consensus was found in 10 runs is indicated. The most frequent motifs are shown in Table 2.

To assign a functional interpretation to the motifs, the consensus of the motifs was compared with the entries described in PlantCARE. Several interesting motifs were found: methyl jasmonate (MeJa) responsive elements, elicitor-responsive elements and the abcissic acid response element (ABRE). It is not surprising to find these elements in gene promoters induced by wounding, because there is a clear cross-talk between the different signal pathways leading to inducible defense gene expression (Birkenmeier and Ryan, 1998). Depending on the nature of a particular aggressor (wounding/insects, fungi, bacteria, virus) the plant is able to fine-tune the induction of defense genes either by employing a single signal molecule or by a combination of the 3 regulators jasmonic acid (JA), ethylene and salicylic acid (SA). In the third and fourth cluster there are also some strong motifs found that do not have a corresponding motif in PlantCARE.

Table 2. Results of the motif search in 4 clusters for the third-order background model. In the second column the consensus of the motifs found is given and the third column contains the number of times this motif was found in the 10 runs. The corresponding motif in PlantCARE is given in column four and a short explanation of the described motif is given in column five.

Cluster	Consensus	Runs	PlantCARE	Function
1	TAArTAAGTCAC	7/10	TGAGTCA	tissue specific GCN4-motif
(11 sequences)			CGTCA	MeJa responsive element
	ATTCAAATTT	8/10	ATACAAAT	element associated to GCN4-motif
	CTTCTTCGATCT	5/10	TTCGACC	elicitor responsive element
2	TTGACyCGy	5/10	TGACG	MeJa responsive element
(sequences)			(T)TGAC(C)	Elicitor responsive element
	mACGTCACCT	7/10	CGTCA	MeJa-responsive element
			ACGT	Abcissic acid response element
3	WATATATmTT	5/10	TATATA	TATA-box like element
(5 sequences)	TCTwCnTC	9/10	TCTCCCT	TCCC-motif, light response element
	ATAAATAkGCnT	7/10	-	-
4	YTGACCGTCCsA	9/10	CCGTCC	Meristem specific activation of H4 gene
(5 sequences)			CCGTCC	Light or elicitor responsive element
			TGACG	MeJa responsive element
			CGTCA	MeJa responsive element
	CAGGTGG	5/10	CACGTG	Light responsive element
			ACGT	Abcissic acid response element
	GCCTymTT	8/10	-	-
	AGAATCAAT	6/10	-	-

3.5. BioProspector

The authors used BioProspector to test three sets of data. The first set consisted of 60 non-coding sequences that were shown to physically interact with the *S. cerevisiae* telomere-binding protein Rap1p. DNA associated with Rap1p were identified by chromatin immunoprecipitation (IP) and purification of DNA fragments enriched by the IP, followed by labeling and hybridization of purified fragments to DNA microarrays containing all of the yeast intergenic regions. The binding site for RAP1 is well characterized, and although published determinations differ slightly in length and consensus, they all agree on the core site RMAYCCR. The sequences analyzed ranged in length from 163 to 1339 bp, and some do not contain the RAP1-binding motif, while others contained multiple copies of it. Three runs of BioProspector were performed, using the input sequence, a zero-order, and a third-order Markov model estimated from the yeast intergenic region to represent the background, respectively. For each sequence, both the forward and the complementary strands were examined. The authors chose M = 200 for an accurate approximation of the motif score distribution, although M = 40 usually gives a reasonable estimate. To examine the performance of threshold sampler on the original data, they let it run 250 times and recorded the motif score and consensus of each.

The second data set contained 136 σ^A-dependent promoter sequence (mostly at positions [-100, 15]) from *Bacillus subtilis*. Each sequence has one RNA polymerase-binding motif on the forward strand, otherwise known as the TATA box. This is a two-block motif: the first block with consensus TTGACA mostly occurs at position -35, and the second block with consensus TATAAT mostly occurs at position -12. BioProspector performed motif finding on this data with a specified gap range of [15, 20].

The third data set consisted of 18 *Escherichia coli* sequences of length 105 which are known to contain CRP-binding sites. CRP is a prokaryotic dimeric DNA-binding protein that binds to adjacent DNA major grooves in a palindromic pattern. One-block motif models using both EM or Gibbs sampling have been applied to this data and yielded satisfactory results, although both mispredicted one or two sites. The authors ran BioProspector on this data to search for a palindromic two-block motif.

The results were as follows.

RAP1 site: background Markov dependency and motif score distribution. The 200 motif scores obtained from the 200 generated sequence sets were approximated by a normal distribution, no matter how the background model was estimated. When using the background model estimated from F_{in}, none of the reported motifs agree with the published RAP1 consensus. When using an independent background model estimated from yeast intergenic region, most of the high-scoring motifs are correct, although there are some high-scoring false positive motifs. When using a third-order Markov background model estimated from yeast intergenic region, the distributions of true positive and false positive motifs separate very well. At scores above 305, all the 9 motifs reported contain a consensus of ACACCCA which agrees with the published result.

TATA-box: two-block motifs. Among the 136 *B. subtilis* sequences containing the two-block TATA-box motif, BioProspector correctly found 70% of the sites and accurately identified the motif consensus as TTGACA, TATAAT. This motif is not very well conserved; many of the missed sites are significantly different from the consensus. For example, the following four missed sites are so variable that the sites predicted by BioProspector match with the consensus better:

	Correct site	Site found
ald	AAGAAT TACACT	TTTCCA TAAAAA
cspB	TTGTTT TGGAGT	ATTACT TATTTT
menE	AATACA GATGAT	TTGAGA TCTTTT
odhA	TTGTGA CAAATT	TTTACT TAGAAT

For the following three sequences, besides finding the correct sites, BioProspector also found a second site closely matching the consensus. In fact, the second site of the sequence veg matches exactly with the TATA-box consensus, which is even better than the correct site.

	Correct site	Site found
abrB	TTGACG TAGTCT	CTGACT TACAAT
veg	TTGACA TACAAT	TTGACA TATAAT
φ105	TTTACA TACAAT	TTGACG TACAAT

CRP site: palindrome motifs. Footprint experiments identified 24 CRP-binding sites in the 18 sequences. However, the aligned segments are not very conserved, especially at the ending positions. EM and Gibbs sampling with one-block motif model succeeded in finding most of the sites, although both mispredicted one or two sites. With a two-block palindromic motif model, all the sites found by BioProspector are correct. The base shifts of the starting position of the first block were caused by specification of a shorter block width and a flexible gap between the two blocks. The resulting probability matrix shows a much more conserved motif with a consensus of WTGTGAWM.

CHAPTER IV

COMPARISONS OF SOME MOTIF FINDING ALGORITHMS

Although users may appreciate guidance concerning which motif discovery algorithm to choose, there are no accepted criteria to predict when one such algorithm will be more successful than another. Tompa et al. (2005) compared 13 algorithms: AlignACE, ANN-Spec, Consensus, GLAM, Improbizer, MEME, MITRA, MotifSampler, oligo/dyad analysis, QuickScore, SeSiMCMC, Weeder and YMF.

For motifs, the authors used TRANSFAC database to choose real transcription factors, their known binding sites, and the positions and orientations of those binding sites. Each such transcription factor gave one data set of sequences. Each such data set consisted of one of three different types of background sequence, with the transcription factor's known binding sites planted at their known positions and orientations. The three types were (1) the binding sites' real promoter sequences (called 'real' hereafter), (2) randomly chosen promoter sequences from the same genome (called 'generic'), and (3) sequences generated by a Markov chain of order 3 (called 'markov'). The process for selecting transcription factors and binding sites from TRANSFAC was as follows: They selected only transcription factors for which TRANSFAC also lists a binding site consensus sequence. For each factor, they removed duplicate instances of the same binding site, removed binding sites missing sequence or position information, removed binding sites whose position was annotated with respect to anything other than transcription or translation start site, removed binding sites whose position was less than -3,000 bp or greater than 0, and removed sequences with two reported binding sites contradicting each other in sequence and position. Any factor with fewer than five remaining binding sites in a single species was then discarded. This resulted in 52 data sets. Six of the data sets are from fly, 26 from human, 12 from mouse and 8 from yeast. As negative controls, they added 4 additional data sets of type markov containing no planted binding sites, and added 2 of them to the fly collection and 2 of them to the yeast collection. For each species, about one-third of its data sets were of each of the types real, generic, and markov. To 31 of the 38 data sets of type generic or markov, they added 1 to 4 additional sequences with no planted binding sites, so that each input sequence contains 0 or more planted binding sites. The number of sequences per data set varied from 1 to 35 with mean 7, and the individual sequence length per data set varied from 500 bp to 3000 bp. The total size of each data set varied from 1 to 70 kb with mean 8 kb. The number of planted binding sites per data set varied from 0 to 76 with mean 9. The data sets are available as a benchmark at the assessment web site http://bio.cs.washington.edu/ assessment.

The authors used several statistics to compare the accuracy of the 13 tools they used in this study. Data revealed that the absolute measures of correctness of these programs were low. For example, site sensitivity was at most 0.22 and correlation coefficient was at most 0.20. The authors warn that this should not be taken as an indictment of computational methods for prediction of regulatory element, for a very great number of reasons. Most importantly, the underlying biology of regulatory mechanisms is very incompletely understood. We lack an absolute standard against which to measure the correctness of tools. The assessment allowed no comparative sequence analysis among species, a powerful method for the prediction of regulatory elements. The assessment allowed no exploitation, except possibly in the data sets of type real, of the fact that the binding sites of multiple transcription factors often occur in close proximity to each other. The assessment depended on TRANSFAC as its standard for the true binding sites; any such database is fallible and biased. Many of the binding sites cataloged in TRANSFAC are usually long: 35 of the sites used in this assessment were each 31 to 71 bp in length. This may reflect lack of precision in the experimental method used, with the true binding site actually a shorter subsequence of the cataloged site. Such long cataloged sites have a detrimental effect on measured sensitivity, both at the nucleotide and site levels. The assessment allowed only one known motif for each data set, despite the fact that 18 data sets of type real are likely to have binding sites for multiple transcription factors. In addition, in comparing the performance of tools, one must keep in mind the fact that predicted set of motif instances was subject to human choices of parameters and pre- and post-processing.

The results of the comparison experiments showed that the tool Weeder outperformed the other tools in most domains and by most measures in this assessment. The authors believe that some part of Weeder's success is due to judicious choices regarding when to predict no motif in a data set: Weeder was run in a "cautious mode", where only the strongest motifs were reported. A few small exceptions to Weeder's domination were that the SeSiMCMC did somewhat better on the fly data set and the MEME3 and YMF did somewhat better on the mouse data sets.

The authors mention that biologists would be well advised to use a few complementary tools in combination rather than relying on a single one and to pursue the top few predicted motifs of each rather than the single most significant motif. They also mention that one of the surprises resulting from this assessment was the realization that the design of a good assessment was itself far from straightforward. Constructing representative data sets, when we do not understand the full truth about transcription factor binding sites, was problematic from the outset. Choosing the most appropriate statistics for evaluating the correctness of predictions was also challenging. This was particularly the case in light of the reality that different tools may predict zero, one or more significant motifs on a given data set, and that real promoter sequences may indeed contain binding sites for zero, one or more distinct transcription factors.

CHAPTER V

DISCUSSION

Despite considerable efforts to date, prediction of regulatory elements remains a complex challenge for biologists and computer scientists. A commonly accepted assumption in biology is that co-regulated genes share similarities in their regulatory mechanism. These similarities at transcriptional level imply that the promoter regions of the genes might contain consensus motifs recognized by the same regulatory proteins (transcription factors). In the upstream regions of such sets of co-regulated genes, the common consensus motifs are statistically overrepresented as compared to their frequency in a background set (of non-coregulated genes). Several methods to search for overrepresented motifs in the upstream region of a set of co-regulated genes have been developed and tested. The weak point of these algorithms is that they tend to be sensitive to the noise. Noise is due to the presence of upstream sequences in the data set that do not contain the motif. Another source of noise comes from the large size of the upstream sequences of the selected genes as compared to the small size of the motifs. Parts of the sequences not containing a motif can indeed be considered as noise. Therefore, it is important to have a motif detection algorithm that can cope with this noise and discriminate between motifs that are overrepresented by chance and motifs that are biologically functional. Some of the motif finding algorithms such as (MEME and

AlignACE) use simple background model based on the frequency of the nucleotides A, C, G, and T in the data set to represent an intergenic sequence. However, a background model solely based on single nucleotide frequencies poorly reflects the complex structure of genomes sequences. To overcome this problem some algorithms such as BioProspector (Liu et al., 2001) uses zero to third-order Markov background model.

Considering the fact that little is known about most transcription factors and their target binding sites, even in well studied organisms, computational tools designed for the discovery of novel regulatory elements, where nothing is assumed a priori of the transcription factor or its preferred binding sites will have advantage over other tools. For these tools, usually a user provides a collection of regulatory regions of genes that are believed to be co-regulated, the computational tool identifies the overrepresented motifs.

As we have seen in the result section of this thesis, all the methods were able to correctly detect the motifs that have been previously detected by laboratory experimental approaches. In addition some algorithms were able to find novel motifs. However, most of these motif finding algorithms have been shown to work successfully in yeast and other lower organisms, but perform significantly worse in higher organisms as Tompa et al. (2005) reported in their recent paper on the comparison of motif finding algorithms. Hon and Jain (2006) take this fact into consideration and develop a deterministic motif finding algorithm with application to the human genome.

Although the biology of the regulatory mechanism is still poorly understood, motif discovery algorithm should include most available biological information. Over the past several years, many tools have been developed for motif discovery. We agree with Tompa et al. (2005) as they have suggested from the outcome of their assessment of the several motif finding tools that biologists should use a few complementary tools in combination rather than relying on a single one and pursue the top few predicted motifs of each rather than the single most significant motif.

REFERENCES

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P. 2002. Molecular Biology of the Cell. 4th Edition, New York: Garland Science.

Ao, W., Gaudet, J., Kent, W.J., Muttumu, S. and Mango, S.E. 2004. Environmentally induced foregut remodeling by PHA-4/FoxA and DAF-12/NHR. Science 305:1743-1746.

Apostolico, A., Bock, M., Lonardi, S. and Xu, X. 2000. Efficient detection of unusual words. J. Comput. Biol. 7:71-94.

Bailey, T.L. and Elkan, C. 1995. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. Machine Learning 21:51-80.

Birkenmeier, G.F. and Ryan, C.A. 1998. Wound signaling in tomato plants evidence that aba is not a primary signal for defense gene activation. Plant Physiol. 117(2):687-693.

Blanchette, M. and Tompa, M. 2002. Discovery of regulatory elements by a computational method for phylogenetic footprinting. Genome Res. 12:739-748.

Blanchette, M., Schwikkowski, B. and Tompa, M. 2002. Algorithms for phylogenetic footprinting. J. Comput. Biol. 9:2110223.

Buhler, J., and Tompa, M. 2002. Finding motifs using random projections. J. Comput. Biol. 9:225-242.

Bussemaker, H., Li, H. and Siggia, E. 2000. Regulatory element detection using a probabilistic segmentation model. Proc. Int. Conf. Intell. Syst. Mol. Biol. 8:67-74.

Eskin, E. and Pevzner, P. 2002. Finding composite regulatory patterns in DNA sequences. Bioinformatics (Supplement 1) 18:S354-S363.

Eskin, E., Grundy, W.N. and Singer, Y. 2000. Protein family classification using sparse Markov transducers. In Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology. AAAI Press, Menlo Park, CA, pp. 134-145.

Favorov, A.V., Gelfand, M.S., Gerasimove, A.V., Mironov, A.A. and Makeev, V.J. 2004. Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length and its validation on the ArcA binding sites. In Proceedings of BGRS 2004 (BGRS, Novosibirsk, 2004).

Fogel, G.B., Weekes, D.G., Varga, G., Dow E.R., Harlow H.B., Onyia, J.E. and Su C. 2004. Discovery of sequence motifs related to coexpression of genes using evolutionary computation. Nucleic Acids Res. 32:3826-3835.

Frith, M.C., Hansen, U., Spouge, J.L. and Weng, Z. 2004. Finding functional sequence elements by multiple local alignment. Nucleic Acids Res. 32:189-200.

Galas, D.J., Eggert, M. and Waterman, M.S. 1985. Rigorous pattern-recognition methods for DNA sequences: analysis of promoter sequences from *Escherichia coli* . J. Mol. Biol. 186:117-128.

Ganesh, R., Siegele, D.A. and Ioerger, T.R. 2003. MOPAC: motif finding by preprocessing and agglomerative clustering from microarrays. Pac. Symp. Biocomput. 8: 41-52.

Goncalves, P.M., Griffioen, G., Minnee, R., Bosma, M., Kraakman, L.S., Mager, W.H. and Planta, R.J. 1995. Transcriptional activation of yeast ribosomal protein genes requires additional elements apart from binding sites for Abf1p and Rap1p. Nucleic Acids Res. 23:1475-1480.

GuhaThakuta, D. and Stormo, G.D. 2001. Identifying target sites for cooperatively binding factors. Bioinformatics 17:608-621.

Gusfield, D. 1997. Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. Cambridge University Press, New York.

Hon, L.S. and Jain, A.N. 2006. A deterministic motif finding algorithm with application to the human genome. Bioinformatics 22:1047-1054.

Hughes, J.D., Estep, P.W., Tavazoie, S. and Church, G.M. 2000. Computational identification of cis-regulatory elements associated with functionally coherent groups of genes in Saccharomyces cerevisiae. J. Mol. Biol. 296:1205-1214.

Kielbasa, S., Korbel, J., Beule, D., Schuchhardt, J. and Herzel, H. 2001. Combining frequency and positional information to predict transcription factor binding sites. Bioinformatics 17:1019-1026.

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.

Lawrence, C.E., Altschule, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F. and Wootton, J.C. 1993. Dteecting subtle sequence signals : A Gibbs sampling strategy for multiple alignment. Science 262:208-214.

Lodish, H., Berk, A., Zipursky, S.,Matsudaira, P., Baltimore, D. and Darnell, J. 2000. Molecular Cell Biology. 4th edition.

Liu, J.S. 2001. Monte Carlo Strategies in Scientific Computing. Springer Series in Statistics, 1st edition. Pp 46.

Liu, J.S. 1995. Bayesian models for multiple local sequence alignment and gibbs sampling strategies. J. Amer. Statist. Assoc. 90:1156-1170.

Liu, J.S. 1994. The collapsed gibbs sampler in Bayesian computations with applications to a gene regulation problem. J. American Stat. Assoc. 29:958-967.

Liu, J.S., Neuwald, A.F. and Lawrence, C.E. 1995. Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. J. American Stat. Assoc. 90:1156-1170.

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. 6:127-138.

Marsan, L. and Sagot, M. 2000. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. J. Comput. Biol. 7:345-362.

Mengeritsky, G. and Smith, T.F. 1987. Recognition of characteristic patterns in sets of functionally equivalent DNA sequences. Comput. Appl. Biosci. 3:223-227.

Pavesi, G., Mauri, G. and Pesole, G. 2001. An algorithm for finding signals of unknown length in DNA sequences. Bioinformatics 17 Suppl. 1: S207-214.

Pavesi, G., Mereghetti, P., Mauri, G. and Pesole, G. 2004. Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. Nucleic Acids Res. 32:W199-W203.

Pesole, G., Prunella, N., Liuni, S., Attimonelli, M. and Saccon, C. 1992. WORDUP: an efficient algorithm for discovering statistically significant patterns in DNA sequences. Nucleic Acids Res. 20: 2871-2875.

Pevzner, P. and Sze, S. 2000. Combinatorial approaches to finding subtle signals in DNA sequences. Proc. Int. Conf. Intell. Syst. Mol. Biol. 8:269-278

Qi, Y., Ye, P. and Bader, J.S. 2005. Genetic interaction motif finding by expectation maximization – a novel statistical model for inferring gene modules from synthetic lethality. BMC Bioinformatics 6:288.

Regnier, M. and Denise, A. 2004. Rare events and conditional events on random strings. Discrete Math. Theor. Comput. Sci. 6:191-214

Reymond, P., Weber, H., Damond, M. and Farmer, E.E. 2000. Differential gene expression in response to mechanical wounding and insect feeding in Arabidopsis. Plant Cell 12:707-719.

Rombauts, S., Dehais, P., Van Montagu, M. and Rouze, P. 1999. PlantCARE, a plant cis-acting regulatory element database. Nucleic Acids Res. 27:295-296.

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. Nature Biotechnology 16:939-945.

Siddharthan, R., Siggia, E.D. and van Nimwegen, E. 2005. PhyloGibbs: A Gibbs sampling motif finder that incorporates phylogeny. PLoS Computational Biology 1:534-556.

Sagot, M. 1998. Spelling approximate or repeated motifs using a suffix tree. LNCS 1380:111-127.

Sinha, S. 2003. Discriminative motifs. J. Comput. Biol. 10:599-615.

Sinha, S. and Tompa, M. 2003. Performance comparison of algorithms for finding transcription factor binding sites. In Third IEEE Symposium on Bioinformatics and Bioengineering, IEEE Press, Los Alamitos, pp. 214-220.

Sinha, S. and Tompa, M. 2003. YMF: a program for discovery of novel transcription factor binding sites by statistical overrepresentation. Nucleic Acids Research 31:3586-3588.

Sinha, S. and Tompa, M. 2002. Discovery of novel transcription factor binding sites by statistical overrepresentation. Nucleic Acids Research 30:5549-5560.

Sinha, S. and Tompa, M. 2000. A statistical method for finding transcription factor binding site. Proc. 8th Intl. Conf. on Intelligent Systems for Molecular Biology, San Diego, CA, August 2000. pp. 344-354.

Smith, T.F. and Waterman, M.S. 1981. Identification of common molecular subsequences. J. Mol. Biol. 147:195-197.

Staden, R. 1989. Methods for discovering novel motif in nucleic acid sequences. Comput. Appl. Biosci. 5:293-298.

Tagle, D., Koop, B., Goodman, M., Slightom, J., Hess, D. and Jones, R. 1988. Embryonic ε and γ globin genes of a prosimian primate (Galago crassicaudatus): nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. J. Mol. Biol. 203:439-455.

Thijs, G., Marchal, K. and Moreau, Y. 2001. A Gibbs sampling method to detect over-represented motifs in upstream regions of co-expressed genes. RECOMB 5:305-312.

Tompa, M. 1999. An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. Proc. 7th Intl. Conf. on Intelligent Systems for Molecular Biology, Heidelberg, Germany, August 1999. pp. 262-271.

Tompa, M., Li, N., Bailey, T., Church, G.M., De Moor, B., Eskin, E., Favorov, A., Frith, M.C., Fu, Y., Kent, W.J., Makeev, V.J., Mironov, A.A., Noble, W.S., Pavesi, G., Pesole, G., Regnier, M., Simonis, N., Sinha, S., Thijs, G., van Helden, J., Vandenbogaert., Weng, Z., van Helden, J., Andre, B. and Collado-Vides, J. 1998. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. J. Mol. Biol. 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 Research 28:1808-1818.

Workman, C.T. and Stormo, G.D. 2000. ANN-Spec: a method for discovering transcription factor binding sites with improved specificity. In Pacific Symposium on Biocomputing (ed. Xing, E.P. Wu, W., Jordan, M.I. and Karp, R.M. 2004. Logos: a modular Bayesian model for de novo motif detection. J. Bioinform. Comput. Biol. 2:127-154.

Zhu, J. and Zhang, M.Q. 1999. SCPD: a promoter database of the yeast *Saccharomyces cerevisiae*. Bioinformatics 15:563-577.

APPENDIX

GLOSSARY OF TERMS USED

Codon – Any group of three consecutive nucleotide bases in a given mRNA molecule that, by its composition and sequence, specifies either a particular amino-acid residue in the polypeptide chain synthesized by translation of that mRNA, or signals the beginning or the end of the message.

Enhancer – A eukaryotic control element that can increase expression of a gene. Enhancers may be located some distance from the gene and may be either upstream or downstream.

Exon - Those portions of genomic DNA sequence which will be represented in the final, mature mRNA.

Genome - The total genetic content contained in a haploid set of chromosomes in eukaryotes, in a single chromosome in bacteria, or in the DNA or RNA of viruses.

Intron – Those portions of genomic DNA which are transcribed (and thus present in the primary transcript) but which are later spliced out. They thus are not present in the mature mRNA.

mRNA (messenger RNA) - an RNA which contains sequences coding for a protein.

Oligonucleotide – Any molecule that contains a small number (two to about ten) of nucleotide units connected by phosphodiester linkages between the 3′ position on the glycose moiety of one nucleotide unit and the 5′ position on the glycose moiety of the adjacent one.

Promoter - The first few hundred nucleotides of DNA "upstream" (on the 5′ side) of a gene, which control the transcription of that gene. The promoter is part of the 5′ flanking DNA, i.e., it is not transcribed into RNA, but without the promoter, the gene is not functional.

Regulon - A set of genes controlled by a common regulator.

Ribosome - A cellular particle which is involved in the translation of mRNAs to make proteins. Ribosomes are a complex consisting of ribosomal RNAs (rRNA) and several proteins.

Stimulon - A set of genes whose transcription responds to a common environmental stimulus.

ORF (Open Reading Frame) – A series of codon triplets, deduced from a DNA sequence, that include a 5′ initiation codon running through a termination codon, and representing a putative or known gene.

TATA box - A TATA box (also called Goldberg-Hogness box) is a DNA sequence found in the promoter region of most genes. It is the binding site of either transcription factors or histones and is involved in the process of transcription by RNA polymerase. It has the core DNA sequence 5′-TATAA-3′, which is usually followed by three or more adenine bases highly conserved through evolution.

VITA

MODAN KUMAR DAS

Candidate for the Degree of

Master of Science

Thesis: A SURVEY OF DEOXYRIBONUCLEIC ACID MOTIF FINDING ALGORITHMS

Major Field: Computer Science

Biographical:

Education: Received B.S. degree in Agriculture from Bangladesh Agricultural University, Mymensingh, Bangladesh in 1979. Received M.S. degree in Genetics and Plant Breeding from the same Institution in 1980. Graduated with a Ph.D. in Crop Science from Oregon State University in 1990. Completed the requirements of an M.S. degree in Computer Science from Oklahoma State University in December, 2006.

Experience: Worked as a Lecturer in Genetics and Plant Breeding at Bangladesh Agricultural College. Employed as Post-Doctoral Research Associate at Virginia Tech and Oklahoma State University in the area of Plant Breeding, Genetics and Molecular Biology. Currently working as a Molecular Biologist for the US Department of Agriculture at the University of Arizona.

Professional Memberships: Member American Society of Agronomy and Crop Science Society of America since 1989.

Name: Modan Kumar Das Date of Degree: December, 2006

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: A SURVEY OF DEOXYRIBONUCLEIC ACID MOTIF FINDING ALGORITHMS

Pages in Study: 69 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study: Unraveling the mechanisms that regulate the expression of genes is a major challenge in biology. An important task in this challenge is to identify regulatory elements, especially the binding sites in deoxyribonucleic acid (DNA) for transcription factors. These binding sites are short DNA segments which are called motifs. Given a set of DNA sequences (promoter region), the motif finding problem is the task of detecting overrepresented motifs that are good candidates for being transcription factor binding sites. Co-regulated genes are known to share some similarities in their regulatory mechanism, possibly at transcriptional level, their promoter regions might contain some common motifs that are binding sites for transcriptional regulators. In recent years, due to combined efforts of computer scientists and molecular biologists several algorithms have been developed for finding DNA motifs. The current study is a survey of these motif finding algorithms. We present some relevant information from biology, then we present a list of motif finding algorithms and describe some of these algorithms in detail, next we present some results that have been obtained in the literature using these algorithms, then we present performance comparisons of some of these algorithms and finally we present a discussion on the motif finding algorithms.

Findings and Conclusions: A survey of the motif finding algorithms in the current study shows that a sensible approach to detect regulatory elements is to search for statistically overrepresented motifs in the promoter region of a set of co-regulated genes. The weak point of the currently available motif finding algorithms is that they tend to be sensitive to the noise. Noise is due to the presence of upstream sequences in the data set that do not contain the motif. All the algorithms studied were able to correctly detect the motifs that have been previously detected by laboratory experimental approaches. In addition, some algorithms were able to find novel motifs. However, most of these motif finding algorithms have been shown to work successfully in yeast and other lower organisms, but perform significantly worse in higher organisms. We conclude that although the biology of the regulatory mechanism is still poorly understood, motif discovery algorithm should include most available biological information. Instead of relying on a single motif finding tool, biologists should use a few complementary tools in combination and pursue the top few predicted motifs of each rather than the single most significant motif.

ADVISER'S APPROVAL: Dr. H.K. Dai